Lookahead Strategies for Sequential Monte Carlo

Ming Lin, Rong Chen, and Jun S. Liu
Xiamen University, Rutgers University and Harvard University

May 26, 2012

Abstract

Based on the principles of importance sampling and resampling, sequential Monte Carlo (SMC) encompasses a large set of powerful techniques dealing with complex stochastic dynamic systems. Many of these systems possess strong memory, with which future information can help sharpen the inference about the current state. By providing theoretical justification of several existing algorithms and introducing several new ones, we study systematically how to construct efficient SMC algorithms to take advantage of the "future" information without creating a substantially high computational burden. The main idea is to allow for lookahead in the Monte Carlo process so that future information can be utilized in weighting and generating Monte Carlo samples, or resampling from samples of the current state.

Keywords: Sequential Monte Carlo; Lookahead weighting; Lookahead sampling; Pilot lookahead; Multilevel; Adaptive lookahead.

*Ming Lin is Associate Professor at the Wang Yanan Institute for Studies in Economics, Xiamen University. Rong Chen is Professor at Department of Statistics, Rutgers University, Peking University. Jun S. Liu is Professor of Statistics, Harvard University. The corresponding author is Rong Chen, Department of Statistics, Rutgers University, Piscataway, NJ 08854, email: rongchen@stat.rutgers.edu, Tel: 732-445-2690 x1123. Ming Lin's research is supported by the National Nature Science Foundation of China grant 11101341 and by the Fundamental Research Funds for the Central Universities 2010221093. Rong Chen's research is sponsored in part by NSF grants DMS 0800183, DMS 0905076 and DMS 0915139. Jun Liu's research was supported in part by NSF grant DMS-0706989 and DMS-1007762.

1 Introduction

Sequential Monte Carlo (SMC) methods have been widely used to deal with stochastic dynamic systems often encountered in engineering, bioinformatics, finance, and many other fields (Gordon et al., 1993; Kong et al., 1994; Avitzour, 1995; Hürzeler and Künsch, 1995; Liu and Chen, 1995; Kitagawa, 1996; Kim et al., 1998; Liu and Chen, 1998; Pitt and Shephard, 1999; Chen et al., 2000; Doucet et al., 2001; Liu, 2001; Fong et al., 2002; Godsill et al., 2004). They utilize the sequential nature of stochastic dynamic systems to generate sequentially weighted Monte Carlo samples of the unobservable state variables or other latent variables, and use these weighted samples for statistical inference of the system or finding stochastic optimization solution. A general framework for SMC is provided in Liu and Chen (1998) and Del Moral (2004). Many successful applications of SMC in diverse areas of science and engineering can be found in Doucet et al. (2001) and Liu (2001).

Dynamic systems often possess strong memory so that future information is often critical for sharpening the inference about the current state. For example, in target tracking systems (Godsill and Vermaak, 2004; Ikoma et al., 2001), at each time point along the trajectory of a moving object, one observes a function of the object's location with noise. Such observations obtained in the future contain substantial information about the current true location, velocity and acceleration of the object. In protein structure prediction problems, often the objective is to find an optimal polymer conformation that minimizes certain energy function. By "growing" the polymer sequentially (Rosenbluth and Rosenbluth, 1955), the construction of polymer conformations can be turned into a stochastic dynamic system with long memory. In such cases, lookahead techniques have been proven very useful (Zhang and Liu, 2002).

To utilize the strong memory effect, Clapp and Godsill (1999) studied fixed-lag smoothing using the information from future. Independently, Chen et al. (2000) proposed the *delayed-sample* method that generates samples of the current state by integrating (marginalizing) out the future states, and showed that this method is effective in solving the problem of adaptive detection and decoding in a wireless communication problem. The computational complexity of this method, however, can be substantial when the number of future states being marginalized out is large. Wang et al. (2002) developed the *delayed-pilot* sampling method which generates random pilot streams to partially explore the space of future states, as well as the *hybrid-pilot* method that combines delayed-sample method and delayed-pilot sampling method. Guo et al. (2004) proposed a *multilevel* method to reduce complexity for large state space system. These low-complexity techniques have been shown to be effective in the flat-fading channel problem treated in Chen et al. (2000). Doucet et al. (2006) proposed a block sampling strategy to utilize future information in generating better samples of the current states. Zhang and Liu (2002) developed the *pilot-exploration resampling* method, which utilizes multiple random pilot paths for each particle of the current state to gather future information, and showed that it is effective in finding the minimum-energy polymer conformation.

In this paper, we formalize the general principle of lookahead in SMC. Several existing methods are then systematically summarized and studied under this principle, with more detailed theoretical justifications. In addition, we propose an adaptive lookahead scheme. The rest of this paper is organized as follows. In Section 2, we briefly overview the general framework of SMC. Section 3 introduces the general principle of lookahead. Section 4 discusses several lookahead methods in detail. In Section 5, we discuss adaptive lookahead. Section 6 presents several applications. The proof of all theorems are presented in the Appendix.

2 Sequential Monte Carlo (SMC)

Following Liu and Chen (1998), we define a stochastic dynamic system as a sequence of evolving probability distributions $\pi_0(\boldsymbol{x}_0), \pi_1(\boldsymbol{x}_1), \cdots, \pi_t(\boldsymbol{x}_t), \cdots$, where $\boldsymbol{x}_t$ is called the state variable. We focus on the case when the state variable evolves with increasing dimension, *i.e.*, $\boldsymbol{x}_t = (x_0, x_1, \cdots, x_t) = (\boldsymbol{x}_{t-1}, x_t)$, where x_t can be multi-dimensional. For example, in the state space model, the latent state x_t evolves through state dynamic $x_t \sim g_t(\cdot \mid \boldsymbol{x}_{t-1})$, and "information" $y_t \sim f_t(\cdot \mid \boldsymbol{x}_t)$ is observed at each time t. In this case,

$$\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_t) \propto g_0(x_0) \prod_{s=1}^{t} g_s(x_s \mid \boldsymbol{x}_{s-1}) f_s(y_s \mid \boldsymbol{x}_s).$$

In this paper, we use the notation $\pi_t(x_t \mid \boldsymbol{x}_{t-1}) \equiv p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_t)$ and $\pi_{t-1}(x_t \mid \boldsymbol{x}_{t-1}) \equiv p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})$. Usually, the goal is to make inference of certain function $h(\boldsymbol{x}_t)$ given all past information $\boldsymbol{y}_t = (y_1, \cdots, y_t)$.

With all the information up to time t, we see that the minimum mean squared error (MMSE) estimator of $h(\boldsymbol{x}_t)$, which minimizes $E_{\pi_t}\left[\widehat{h} - h(\boldsymbol{x}_t)\right]^2$, is $\widehat{h} = E_{\pi_t}\big(h(\boldsymbol{x}_t)\big)$. When an analytic solution of $E_{\pi_t}\big(h(\boldsymbol{x}_t)\big)$ is not available, importance sampling Monte Carlo scheme can be employed (Marshall, 1956; Liu, 2001). Specifically, we can draw samples $\boldsymbol{x}_t^{(j)}$, $j = 1, \cdots, m$, from a trial distribution $r_t(\boldsymbol{x}_t)$, given that $r_t(\boldsymbol{x}_t)$'s support covers $\pi_t(\boldsymbol{x}_t)$'s support, then $E_{\pi_t}\big(h(\boldsymbol{x}_t)\big)$ can be estimated by

$$\frac{1}{m}\sum_{j=1}^{m} w_t^{(j)} h(\boldsymbol{x}_t^{(j)}) \qquad \text{or} \qquad \frac{1}{\sum_{j=1}^{m} w_t^{(j)}} \sum_{j=1}^{m} w_t^{(j)} h(\boldsymbol{x}_t^{(j)}),$$

where $w_t^{(j)} = w_t(\boldsymbol{x}_t^{(j)}) = \pi_t(\boldsymbol{x}_t^{(j)})/r_t(\boldsymbol{x}_t^{(j)})$ is referred to as a *proper importance weight* for $\boldsymbol{x}_t^{(j)}$ with respect to $\pi_t(\boldsymbol{x}_t)$. Although the second estimator is biased, it is often less variable and easier to use since in this case w_t only needs to be evaluated up to a multiplicative constant. Throughout this paper, we will use $\boldsymbol{x}_t$ and $\boldsymbol{x}_t^{(j)}$ to denote the true state and the Monte Carlo sample, respectively.

The basis of all SMC methods is the so-called "sequential importance sampling (SIS)" (Kong et al., 1994; Liu, 2001), which sequentially builds up a high dimensional sample according to the chain rule. More precisely, the sample $\boldsymbol{x}_t^{(j)}$ is built up sequentially according to a series of low dimensional conditional distributions:

$$r_t(\boldsymbol{x}_t) = q_0(x_0)q_1(x_1|\boldsymbol{x}_0)q_2(x_2|\boldsymbol{x}_1)\cdots q_t(x_t|\boldsymbol{x}_{t-1}).$$

The importance weight for the sample can be updated sequentially as

$$w_t(\boldsymbol{x}_t^{(j)}) = w_{t-1}(\boldsymbol{x}_{t-1}^{(j)})u_t(\boldsymbol{x}_t^{(j)}), \qquad \text{where} \qquad u_t(\boldsymbol{x}_t^{(j)}) = \frac{\pi_t(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})q_t(x_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}$$

is called the *incremental weight.* The choice of the trial distribution r_t (or q_t) has a significant impact on the accuracy and efficiency of the algorithm. As a general principle, a good trial distribution should be close to the target distribution. An obvious choice of q_t in the dynamic system setting is $q_t(x_t|\boldsymbol{x}_{t-1}) = \pi_{t-1}(x_t|\boldsymbol{x}_{t-1})$ (Avitzour, 1995; Gordon et al., 1993; Kitagawa, 1996). Kong et al. (1994) and Liu and Chen (1998) argued that $q_t(x_t|\boldsymbol{x}_{t-1}) = \pi_t(x_t|\boldsymbol{x}_{t-1})$ is a better trial distribution because of its usage of the most "up-to-date" information to generate x_t. More choices of $q_t(x_t|\boldsymbol{x}_{t-1})$ can be found in Chen and Liu (2000); Kotecha and Djuric (2003); Lin et al. (2005); Liu and Chen (1998); van der Merwe et al. (2002) and Pitt and Shephard (1999).

As t increases, the distribution of the importance weight w_t often becomes increasingly skewed (Kong et al., 1994), resulting in many unrepresentative samples of $\boldsymbol{x}_t$. A resampling scheme is often used to alleviate this problem (Gordon et al., 1993; Liu and Chen, 1995; Kitagawa, 1996; Liu and Chen, 1998; Pitt and Shephard, 1999; Chopin, 2004; Del Moral, 2004). The basic idea is to imagine implementing multiple SIS procedures in parallel, i.e., to generate $\{\boldsymbol{x}_t^{(1)}, \cdots, \boldsymbol{x}_t^{(m)}\}$ at each step t, with corresponding weights $\{w_t^{(1)}, \cdots, w_t^{(m)}\}$, and resample from the set according to a certain "priority score." More precisely, suppose we have obtained $\{(\boldsymbol{x}_t^{(j)}, w_t^{(j)}), j = 1, \cdots, m\}$ that is properly weighted with respect to $\pi_t(\boldsymbol{x}_t)$, then we create a new set of weighted samples as follows:

Resampling Scheme

- For each sample $\boldsymbol{x}_t^{(j)}$, $j = 1, \cdots, m$, assign a *priority score* $\alpha_t^{(j)} > 0$.
- For $j = 1, \cdots, m$,
  - Randomly draw $\boldsymbol{x}_t^{*(j)}$ from the set $\{\boldsymbol{x}_t^{(j)}, j = 1, \cdots, m\}$ with probabilities proportional to $\{\alpha_t^{(j)}, j = 1, \cdots, m\}$;
  - If $\boldsymbol{x}_t^{*(j)} = \boldsymbol{x}_t^{(j_0)}$, then set the new weight associated with $\boldsymbol{x}_t^{*(j)}$ to be $w_t^{*(j)} = w_t^{(j_0)}/\alpha_t^{(j_0)}$.

- Return the new set of weighted samples $\{(\boldsymbol{x}_t^{*(j)}, w_t^{*(j)}), j = 1, \cdots, m\}$.

This new set of weighted samples is also approximately properly weighted with respect to $\pi_t(\boldsymbol{x}_t)$. Often, $\alpha_t^{(j)}$ are chosen to be proportional to $w_t^{(j)}$, so that the new samples are equally weighted. Some improved resampling schemes can be found in Liu and Chen (1998); Carpenter et al. (1999); Crisan and Lyons (2002); Liang et al. (2002) and Pitt (2002).

Resampling plays an important role in SMC. Chopin (2004) and Del Moral (2004) provides asymptotic results on its effect, but its finite sample effects have not been fully understood. Performing resampling at every step t is usually neither necessary nor efficient since it induces excessive variations (Liu and Chen, 1995). Liu and Chen (1998) suggests to use either a deterministic schedule, in which resampling only takes place at time $T, 2T, 3T, \cdots$, or a dynamic schedule, in which resampling is performed when the effective sample size (Kong et al., 1994) $\mathrm{ESS} = m/(1 + v_t(w))$ is less than certain threshold, where $v_t(w)$ is the estimated coefficient of variation, i.e.,

$$v_t(w) = \frac{\frac{1}{m}\sum_{j=1}^m \left(w_t^{(j)} - \sum_{j=1}^m w_t^{(j)}/m\right)^2}{\left(\sum_{j=1}^m w_t^{(j)}/m\right)^2}. \tag{1}$$

In problems that the state variable x_t takes values in a finite set $\mathcal{A} = \{a_1, \cdots, a_{|\mathcal{A}|}\}$, duplicated samples produced in sampling or resampling steps result in repeated calculation and a waste of resources. Using an idea related to the rejection control (Liu et al., 1998), Fearnhead and Clifford (2003) developed a more efficient scheme that combines sampling and resampling in one step and guarantees to generate distinctive samples.

Most of the SMC algorithms are designed for filtering and smoothing problems. It is a challenging problem when the system has unknown fixed parameters to be estimated and learned. Some new development can be found in Gilks and Berzuini (2001); Chopin (2002); Fearnhead (2002); Andrieu et al. (2010) and Carvalho et al. (2010). In this paper we assume all the parameters are known.

3 The Principle of Lookahead

To formalize our argument that the "future" information is helpful for the inference about the current state, we assume that the dynamic system π_t offers more and more "information" of the state variables as t increases. A simple way to quantify this concept is to assume that the information available at time t takes the form $\boldsymbol{y}_t = (y_1, y_2, \cdots, y_t)$, and increments to $(\boldsymbol{y}_t, y_{t+1})$ at time $t + 1$. The dynamic system $\pi_t(\boldsymbol{x}_t)$ simply takes the form of $\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_t)$. Although this framework is not all-inclusive, it is sufficiently broad and our theoretical results are all under this setting. The basic lookahead principle is to use "future" information for the inference of the current state. That is, we believe that $E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta})$ results in a better inference of the current state $h(\boldsymbol{x}_t)$ than $E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_t)$ for any $\Delta > 0$. Thus, if the added computational burden is not considered, we would like to use a Monte Carlo estimate of $E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta})$ to make inference on $h(\boldsymbol{x}_t)$.

Here we study the benefit of the lookahead strategy rigorously. Let $\widehat{h}_{t+\Delta}$ be a consistent Monte Carlo estimator of $E_{\pi_{t+\Delta}}\big(h(\boldsymbol{x}_t)\big) = E\big(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}\big)$ and $\widehat{h}_{t+\Delta}$ is independent of the true state $\boldsymbol{x}_t$ conditional on $\boldsymbol{y}_{t+\Delta}$. The mean squared difference between $h(\boldsymbol{x}_t)$ and its estimator $\widehat{h}_{t+\Delta}$, averaged over the Monte Carlo samples, the true state, and the future observations can be decomposed as

$$\begin{aligned} E_{\pi_t}\left[\widehat{h}_{t+\Delta} - h(\boldsymbol{x}_t)\right]^2 &= E_{\pi_t}\left[\widehat{h}_{t+\Delta} - E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}\right)\right]^2 + E_{\pi_t}\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}\right) - h(\boldsymbol{x}_t)\right]^2 \\ &\triangleq I(\Delta) + II(\Delta). \end{aligned} \tag{2}$$

As the Monte Carlo sample size tends to infinity, $I(\Delta)$, which is the variance of the consistent estimator, tends to zero. For $II(\Delta)$, we can show that,

Proposition 1. *For any square integrable function $h(\cdot)$, $II(\Delta)$ decreases as Δ increases.*

The proof is given in the appendix.

When the Monte Carlo sample size is sufficiently large, $I(\Delta)$ becomes negligible relative to $II(\Delta)$. Hence, the above proposition implies that a consistent Monte Carlo estimator of $E\big(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}\big)$ is always more accurate with larger Δ when the Monte Carlo sample size is sufficiently large.

However, this gain of accuracy is not always desirable in practice because of the additional computational costs. Most of the time additional computational resources are needed to obtain consistent estimators of $E\big(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}\big)$ with larger Δ. Furthermore, $I(\Delta)$ sometimes increases sharply as Δ increases when the Monte Carlo sample size is fixed. More detailed analysis is shown in Section 5.

In order to achieve the goal of estimating the *lookahead* expectation $E_{\pi_{t+\Delta}}\big(h(\boldsymbol{x}_t)\big)$ effectively using SMC, we may consider defining a new stochastic dynamic system with the probability distribution at step t being the Δ-step lookahead (or delayed) distribution, *i.e.,*

$$\pi_t^*(\boldsymbol{x}_t) = \pi_{t+\Delta}(\boldsymbol{x}_t) = \int \pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}) dx_{t+1} \cdots dx_{t+\Delta}. \tag{3}$$

With the system defined by $\{\pi_0^*(\boldsymbol{x}_0), \pi_1^*(\boldsymbol{x}_1), \cdots\}$, the same SMC recursion can be carried out.

In practice, however, it is often difficult to use this modified system directly since the analytic evaluation of the integration/summation in (3) is impossible for most systems. Even when the state variables take values from a finite set so that $\pi_{t+\Delta}(\boldsymbol{x}_t)$ can be calculated exactly through summation, the number of terms in the summation grows exponentially with Δ. Nonetheless, the lookahead system $\{\pi_0^*(\boldsymbol{x}_0), \pi_1^*(\boldsymbol{x}_1), \cdots\}$ suggests a potential direction that we can work towards.

There are three possible ways to make use of the future information: (i) for choosing a good trial distribution $r_t(\boldsymbol{x}_t)$ close to $\pi_t^*(\boldsymbol{x}_t)$; (ii) for calculating and keeping track of the importance weight for $\boldsymbol{x}_t$ using $\pi_t^*(\boldsymbol{x}_t)$ as the target distribution; and (iii) for setting up an effective resampling priority score function $\alpha_t(\boldsymbol{x}_t)$ using information provided by $\pi_t^*(\boldsymbol{x}_t)$. Detailed algorithms are given in the next section.

We note here that *lookahead* (into the "future") strategies are mathematically equivalent to *delay* strategies (i.e., making inference after seeing more data) in Chen et al. (2000). In our setup, we assume that the current time is $t+\Delta$ and we observe $y_1, \cdots, y_{t+\Delta}$. In fact, some of the algorithms we covered were initially named 'delay algorithms', under the notion that the system allows certain delay in estimation. The reason that we choose to use the term 'lookahead' instead of 'delay' is that, we focus on sampling of x_t, using information after time t (i.e its own future). It is easier to discuss and compare the same x_t when looking further into the future (increasing Δ), rather than a longer delay (with a fixed current time and discuss the estimation of $x_{t-\Delta}$ with changing Δ).

The lookahead algorithms we discuss here are closely related to the smoothing problem in state space models where one is interested in making inference with respect to $p(x_t \mid y_1, \cdots, y_T)$ for $t = 1, \cdots, T$. Many algorithms, some are closely related to our approach, can be found in (Godsill et al., 2004; Douc et al., 2009; Briers et al., 2010; Carvalho et al., 2010; Fearnhead et al., 2010) and others. However, in this paper we emphasize on dynamically processing of $p(x_t \mid y_1, \cdots, y_{t+\Delta})$ for $t = 1, \cdots, n$. It has the characteristic of both filtering (updating as new information comes in) and smoothing (inference with future information).

Another possible benefit of the proposed lookahead strategy is that it tends to be more robust to outliers, since the future information will correct the mis-information from the outliers. This is particularly helpful during resampling stages. With an outlier, the 'good samples' that are close to the true state will be mistakenly given smaller weights. Resampling according to weights will then more likely to remove these 'good sample'. Lookahead that takes into account of more information will be very useful in such a situation.

A 'true' lookahead would utilize the expected (but unobserved) future information in generating samples of current x_t. The popular and powerful *auxiliary particle filter* (Pitt and Shephard, 1999) is based on such an insight, though it only looks ahead one step. Our experience shows that the improvement is limited with more steps of such a 'true' lookahead scheme, as the information is limited to $y_1, \cdots, y_t$. Here we focus on the utilization of the extra information provided by future observations.

4 Basic Lookahead Strategies

4.1 Lookahead weighting algorithm

Suppose at step $t+\Delta$, we obtain a set of weighted samples $\{(\boldsymbol{x}_{t+\Delta}^{(j)}, \overline{w}_{t+\Delta}^{(j)}), j=1,\cdots,m\}$ properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta})$, using the standard concurrent SMC. With the same weight $w_t^{(j)} \triangleq \overline{w}_{t+\Delta}^{(j)}$, the partial chain $\boldsymbol{x}_t^{(j)}$ is also properly weighted with respect to the marginal distribution $\pi_{t+\Delta}(\boldsymbol{x}_t)$. Specifically, we have the following algorithmic steps.

Lookahead weighting algorithm:

- *At time $t=0$, for $j=1,\cdots,m$*
  - *Draw $(x_0^{(j)},\cdots,x_\Delta^{(j)})$ from distribution $q_0(x_0)\prod_{s=1}^{\Delta} q_s(x_s \mid \boldsymbol{x}_{s-1})$.*
  - *Set*

$$w_0^{(j)} \propto \frac{\pi_\Delta(\boldsymbol{x}_\Delta^{(j)})}{q_0(x_0)\prod_{s=1}^{\Delta} q_s(x_s^{(j)} \mid \boldsymbol{x}_{s-1}^{(j)})}.$$

- *At times $t=1,2,\cdots$, suppose we obtained $\{(\boldsymbol{x}_{t+\Delta-1}^{(j)}, w_{t-1}^{(j)}), j=1,\cdots,m\}$ properly weighted with respect to $\pi_{t+\Delta-1}(\boldsymbol{x}_{t+\Delta-1})$.*
  - *(Optional) Resample with probability proportional to the priority scores $\alpha_{t-1}^{(j)} = w_{t-1}^{(j)}$ to obtain a new set of weighted samples.*
  - *Propagation: For $j=1,\cdots,m$:*
    - *(Sampling.) Draw $x_{t+\Delta}^{(j)}$ from distribution $q_{t+\Delta}(x_{t+\Delta} \mid \boldsymbol{x}_{t+\Delta-1}^{(j)})$. Set $\boldsymbol{x}_{t+\Delta}^{(j)} = (\boldsymbol{x}_{t+\Delta-1}^{(j)}, x_{t+\Delta}^{(j)})$.*
    - *(Updating Weights.) Set*

$$w_t^{(j)} \propto w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}^{(j)})}{\pi_{t+\Delta-1}(\boldsymbol{x}_{t+\Delta-1}^{(j)}) q_{t+\Delta}(x_{t+\Delta}^{(j)} \mid \boldsymbol{x}_{t+\Delta-1}^{(j)})}.$$

  - *Inference: $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ is estimated by $\sum_{j=1}^m w_t^{(j)} h(\boldsymbol{x}_t^{(j)}) / \sum_{j=1}^m w_t^{(j)}$.*

Because the $x_t^{(j)}$ are still generated based on the information up to step t, e.g., $q_t(x_t \mid \boldsymbol{x}_{t-1}) = \pi_t(x_t \mid \boldsymbol{x}_{t-1})$, and the future information is utilized only through weight adjustments, Chen et al. (2000) call this method the *delayed-weight* method. Clapp and Godsill (1999) called the procedure *sequential imputation with decision step*, as inference and decisions are made separately at different time steps.

The lookahead weighting algorithm is a simple scheme to provide a consistent estimator for $E_{\pi_{t+\Delta}}\big(h(\boldsymbol{x}_t)\big)$ with almost no additional computational cost, except for some additional memory buffer. Hence, it is often useful in real-time filtering problems (Chen et al., 2000; Kantas et al., 2009). However, when Δ is large, it is well known that such a forward algorithm is highly inaccurate and inefficient in approximating the smoothing distribution $\pi_{t+\Delta}(x_t)$ (e.g. Godsill et al. (2004); Douc et al. (2009); Briers et al. (2010); Fearnhead et al. (2010); Carvalho et al. (2010)).

4.2 Exact lookahead sampling

This method was proposed by Chen et al. (2000), termed as *delayed-sample* method. Its key is to use the modified stochastic dynamic system defined by $\pi_t^*(\boldsymbol{x}_t) = \pi_{t+\Delta}(\boldsymbol{x}_t)$ in (3) to construct the importance sampling distribution. At step t, the conditional sampling distribution for $x_t^{(j)}$ is chosen to be

$$q_t(x_t|\boldsymbol{x}_{t-1}^{(j)}) = \pi_t^*(x_t|\boldsymbol{x}_{t-1}^{(j)}) = \pi_{t+\Delta}(x_t|\boldsymbol{x}_{t-1}^{(j)}), \tag{4}$$

and the weight is updated accordingly as

$$w_t(\boldsymbol{x}_t^{(j)}) = w_{t-1}(\boldsymbol{x}_t^{(j)}) \frac{\pi_t^*(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}^*(\boldsymbol{x}_{t-1}^{(j)}) \pi_t^*(x_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})} = w_{t-1}(\boldsymbol{x}_t^{(j)}) \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t+\Delta-1}(\boldsymbol{x}_{t-1}^{(j)})}.$$

Figure 1 illustrates the method with $x_t \in \mathcal{A} = \{0, 1\}$ and $\Delta = 2$, in which the trial distribution is

$$\begin{aligned} q_t(x_t = i\,|\boldsymbol{x}_{t-1}^{(j)}) &= \pi_{t+2}(x_t = i\,|\boldsymbol{x}_{t-1}^{(j)}) = \sum_{x_{t+1}} \sum_{x_{t+2}} \pi_{t+2}(x_t = i, x_{t+1}, x_{t+2}|\boldsymbol{x}_{t-1}^{(j)}) \\ &\propto \sum_{x_{t+1}} \sum_{x_{t+2}} \pi_{t+2}(\boldsymbol{x}_{t-1}^{(j)}, x_t = i, x_{t+1}, x_{t+2}) \end{aligned}$$

for $i = 0, 1$.

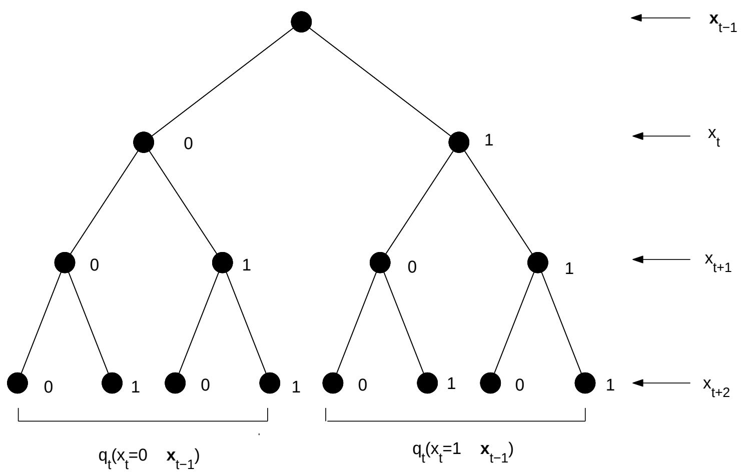

Figure 1: Illustration of the exact lookahead sampling method, in which the trail distribution $q_t(x_t = i|\boldsymbol{x}_{t-1}^{(j)})$, $i = 0, 1$, is proportional to the summation of $\pi_{t+2}(x_t = i, x_{t+1}, x_{t+2}|\boldsymbol{x}_{t-1}^{(j)})$ for $x_{t+1}, x_{t+2} = 0, 1$.

The exact lookahead sampling algorithm is shown as follows.

Exact lookahead sampling algorithm:

- *At time* $t = 0$, *for* $j = 1, \cdots, m$
  - *Draw* $x_0^{(j)}$ *from distribution* $q_0(x_0)$.
  - *Set* $w_0^{(j)} = \pi_\Delta(x_0^{(j)})/q_0(x_0^{(j)})$.
- *At times* $t = 1, 2, \cdots,$
  - *(Optional) Resample* $\{\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)}, j = 1, \cdots, m\}$ *with priority scores* $\alpha_{t-1}^{(j)} = w_{t-1}^{(j)}$.
  - *Propagation: For* $j = 1, \cdots, m$
    - *(Sampling.) Draw* $x_t^{(j)}$ *from distribution*

$$q_t(x_t \mid \boldsymbol{x}_{t-1}^{(j)}) = \pi_{t+\Delta}(x_t \mid \boldsymbol{x}_{t-1}^{(j)}) = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t)}{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}.$$

    - *(Updating Weights.) Set*

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t+\Delta-1}(\boldsymbol{x}_{t-1}^{(j)})}.$$

  - *Inference:* $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ *is estimated by* $\sum_{j=1}^m w_t^{(j)} h(\boldsymbol{x}_t^{(j)}) / \sum_{j=1}^m w_t^{(j)}$.

Specifically, for models with finite state space, the sampling and weight update steps in the exact lookahead sampling method involve evaluation of summations of the form

$$\begin{aligned} \pi_{t+\Delta}(\boldsymbol{x}_t) &= \sum_{x_{t+1},\cdots,x_{t+\Delta}} \pi_{t+\Delta}(\boldsymbol{x}_t, x_{t+1}, \cdots, x_{t+\Delta}) \\ &\propto \sum_{x_{t+1},\cdots,x_{t+\Delta}} g_0(x_0) \prod_{s=1}^{t+\Delta} g_s(x_s \mid \boldsymbol{x}_{s-1}) f_s(y_s \mid \boldsymbol{x}_s). \end{aligned} \tag{5}$$

For continuous state space, it is more difficult to adopt this approach, as one needs to generate samples from

$$q_t(x_t \mid \boldsymbol{x}_{t-1}^{(j)}) = \pi_{t+\Delta}(x_t \mid \boldsymbol{x}_{t-1}^{(j)}) \propto \int \pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t, x_{t+1}, \cdots, x_{t+\Delta}) dx_{t+1} \cdots dx_{t+\Delta} \tag{6}$$

and evaluate it in order to update the weight. A slightly different version of the algorithm was proposed in Clapp and Godsill (1999), termed as *lagged time filtering density*. Instead of calculating the exact sampling density (5) or (6), and sample from it, they proposed to use forward filtering backward sampling techniques of Carter and Kohn (1994) and Clapp and Godsill (1997).

As demonstrated in Chen et al. (2000) and Clapp and Godsill (1999), the exact lookahead sampling method can achieve a significant improvement in performance compared to the concurrent

SMC method. Chen et al. (2000) provided some heuristic justification of this method. Here we provide a theoretical justification by showing that the exact lookahead sampling method generates more effective samples (or "particles") than any trial distribution that does not utilize the future information.

To set up the analysis, we assume that $\{(\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)}), j = 1, \cdots, m\}$ is properly weighted with respect to $\pi_{t-1}(\boldsymbol{x}_{t-1})$ (not the lookahead distribution). We compare two sampling schemes. In exact lookahead sampling, $x_t^{(1,j)}$ is generated from $\pi_{t+\Delta}(x_t|\boldsymbol{x}_{t-1}^{(j)})$, and $\boldsymbol{x}_t^{(1,j)} = (\boldsymbol{x}_{t-1}^{(j)}, x_t^{(1,j)})$ is properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$ by weight

$$w_t^{(1,j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}. \tag{7}$$

Let sample $x_t^{(2,j)}$ be generated from a trial distribution $q_t(x_t|\boldsymbol{x}_{t-1}^{(j)})$ that uses no future information, *i.e.*, $q_t(x_t|\boldsymbol{x}_{t-1}^{(j)})$ does not depend on $y_{t+1}, \cdots, y_{t+\Delta}$, then $\boldsymbol{x}_t^{(2,j)} = (\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)})$ is properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$ using the weight

$$w_t^{(2,j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_t^{(2,j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(2,j)} \mid \boldsymbol{x}_{t-1}^{(j)})}. \tag{8}$$

Let the subscript $\pi_{t+\Delta}$ indicate that the corresponding operations are to be taken conditional on $\boldsymbol{y}_{t+\Delta}$, and let

$$E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right) = \int h(\boldsymbol{x}_{t-1}^{(j)}, x_t) \pi_{t+\Delta}(x_t \mid \boldsymbol{x}_{t-1}^{(j)})\, dx_t.$$

We have the following proposition:

Proposition 2.

$$var_{\pi_{t+\Delta}}\left(w_t^{(2,j)}\right) \geq var_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right), \tag{9}$$

and

$$\begin{aligned} var_{\pi_{t+\Delta}}\left[w_t^{(2,j)} h(\boldsymbol{x}_t^{(2,j)})\right] &\geq var_{\pi_{t+\Delta}}\left[w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right)\right], \quad (10) \\ var_{\pi_{t+\Delta}}\left[w_t^{(2,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right)\right] &\geq var_{\pi_{t+\Delta}}\left[w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right)\right]. \quad (11) \end{aligned}$$

The proof is presented in the Appendix.

Note that the right-hand sides of (10) and (11) use Rao-Blackwellization estimator

$$w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right).$$

For finite state space, it is often achievable since

$$E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right) = \sum_{i=1}^{|\mathcal{A}|} h(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i) \pi_{t+\Delta}(x_t = a_i \mid \boldsymbol{x}_{t-1}^{(j)})$$

where $\pi_{t+\Delta}(x_t = a_i \mid \boldsymbol{x}_{t-1}^{(j)})$ have been computed during the propagation step. Also note that (10) does not provide a direct comparison between $\sum_{j=1}^m w_t^{(1,j)} h(\boldsymbol{x}_t^{(j)})$ and $\sum_{j=1}^m w_t^{(2,j)} h(\boldsymbol{x}_t^{(j)})$. This is because the sampling efficiency is also related to function $h(\cdot)$. If $h(\boldsymbol{x}_t)$ does not depend on x_t, then (10) indeed shows that the full lookahead sampler is always better. Otherwise, this proposition suggests to use

$$\frac{1}{\sum_{j=1}^m w_t^{(j)}} \sum_{j=1}^m w_t^{(j)} E_{\pi_{t+\Delta}} \left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)} \right)$$

for estimation in the exact lookahead sampler.

As a direct consequence of Proposition 2, the following proposition shows that exact lookahead sampling is more efficient than lookahead weighting. Suppose in lookahead weighting, sample $\boldsymbol{x}_t^{(3,j)} = \boldsymbol{x}_t^{(2,j)} = (\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)})$ is available at time t and $\boldsymbol{x}_{t+1:t+\Delta}^{(3,j)}$ is generated from

$$\prod_{s=t+1}^{t+\Delta} q_s(x_s \mid \boldsymbol{x}_t^{(3,j)}, \boldsymbol{x}_{t+1:s-1})$$

in the next Δ steps. Let $\boldsymbol{x}_{t+\Delta}^{(3,j)} = (\boldsymbol{x}_t^{(3,j)}, \boldsymbol{x}_{t+1:t+\Delta}^{(3,j)})$, then the weight corresponding to the lookahead weighting algorithm is

$$w_t^{(3,j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}^{(3,j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{s=t}^{t+\Delta} q_s(x_s^{(3,j)} \mid \boldsymbol{x}_{s-1}^{(3,j)})}.$$

We have the following proposition:

Proposition 3.

$$var_{\pi_{t+\Delta}} \left(w_{t+\Delta}^{(3,j)} \right) \geq var_{\pi_{t+\Delta}} \left(w_t^{(2,j)} \right),$$

and for any square integrable function $h(\boldsymbol{x}_t)$*,*

$$var_{\pi_{t+\Delta}} \left[w_{t+\Delta}^{(3,j)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}) \right] \geq var_{\pi_{t+\Delta}} \left[w_t^{(2,j)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}) \right].$$

The proof is presented in the Appendix.

In the exact lookahead sampling, the incremental weight $U_t = \pi_{t+\Delta}(\boldsymbol{x}_{t-1})/\pi_{t+\Delta-1}(\boldsymbol{x}_{t-1})$ usually will be close to 1 when Δ is large, so the variance of weights typically decreases as Δ increases (Doucet et al., 2006). The benefit of exact lookahead sampling, however, comes at the cost of increased analytical and computational complexities due to the need of marginalizing out the future states $x_{t+1}, \cdots, x_{t+\Delta}$ in (3). Often, the computational cost grows exponentially as the lookahead step Δ increases.

4.3 Block sampling

Doucet et al. (2006) proposes a block sampling strategy, which can be viewed as a variation of lookahead. A slightly modified version (under our notation) is given as follows.

Block sampling algorithm:

- *At time* $t = 0$, *for* $j = 1, \cdots, m$
  - *Draw* $(x_0^{(j)}, \cdots, x_\Delta^{(j)})$ *from distribution* $q_0(x_0) \prod_{s=1}^{\Delta} q_s(x_s \mid \boldsymbol{x}_{s-1})$
  - *Set*

$$w_0^{(j)} \propto \frac{\pi_\Delta(\boldsymbol{x}_\Delta^{(j)})}{q_0(x_0) \prod_{s=1}^{\Delta} q_s(x_s^{(j)} \mid \boldsymbol{x}_{s-1}^{(j)})}.$$

- *At times* $t = 1, 2, \cdots,$
  - *(Optional) Resample* $\{\boldsymbol{x}_{t+\Delta-1}^{(j)}, w_{t-1}^{(j)}, j = 1, \cdots, m\}$ *with priority scores* $\alpha_{t-1}^{(j)} = w_{t-1}^{(j)}$.
  - *Propagation: For* $j = 1, \cdots, m$
    - *(Sampling.) Draw* $\boldsymbol{x}_{t:t+\Delta}^{*(j)}$ *from* $q_t(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t+\Delta-1}^{(j)})$.
    - *(Updating Weights.) Set*

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{*(j)}) \lambda_t(\boldsymbol{x}_{t:t+\Delta-1}^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{*(j)})}{\pi_{t+\Delta-1}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta-1}^{(j)}) q_t(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta-1}^{(j)})}.$$

    - *Let* $\boldsymbol{x}_{t+\Delta}^{(j)} = (\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{*(j)})$.
  - *Inference:* $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ *is estimated by* $\sum_{j=1}^m w_t^{(j)} h(\boldsymbol{x}_t^{(j)}) / \sum_{j=1}^m w_t^{(j)}$.

Here $\lambda_t(\boldsymbol{x}_{t:t+\Delta-1}^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{*(j)})$ is called the *artificial* conditional distribution.

Doucet et al. (2006) suggested that one should choose $q_t(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t+\Delta-1}^{(j)}) = q_t(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t-1}^{(j)})$, that is, the trial distribution doesn't depend on $\boldsymbol{x}_{t:t+\Delta-1}^{(j)}$. Then the optimal choice of q_t and λ_t are

$$q_t(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta-1}^{(j)}) = \pi_{t+\Delta}(\boldsymbol{x}_{t:t+\Delta}^{*(j)} \mid \boldsymbol{x}_{t-1}^{(j)}),$$

and

$$\lambda_t(\boldsymbol{x}_{t:t+\Delta-1}^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{*(j)}) = \pi_{t+\Delta-1}(\boldsymbol{x}_{t:t+\Delta-1}^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)}).$$

Note that in this case, the marginal trial distribution of $x_t^{*(j)}$ is $\pi_{t+\Delta}(x_t^{*(j)} \mid \boldsymbol{x}_{t-1}^{(j)})$, and the weight is updated by

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t+\Delta-1}(\boldsymbol{x}_{t-1}^{(j)})}.$$

In this case, the blocking sampling method becomes the exact lookahead sampling.

In practice, we can use

$$q_t(\boldsymbol{x}^{*(j)}_{t:t+\Delta} \mid \boldsymbol{x}^{(j)}_{t-1}, \boldsymbol{x}^{(j)}_{t:t+\Delta-1}) = \widehat{\pi}_{t+\Delta}(\boldsymbol{x}^{*(j)}_{t:t+\Delta} \mid \boldsymbol{x}^{(j)}_{t-1}),$$

and

$$\lambda_t(\boldsymbol{x}^{(j)}_{t:t+\Delta-1} \mid \boldsymbol{x}^{(j)}_{t-1}, \boldsymbol{x}^{*(j)}_{t:t+\Delta}) = \widehat{\pi}_{t+\Delta-1}(\boldsymbol{x}^{(j)}_{t:t+\Delta,t-1} \mid \boldsymbol{x}^{(j)}_{t-1}),$$

which are low complexity approximations of the optimal q_t and λ_t.

4.4 Pilot lookahead sampling

Because of the desire to explore the space of future states with controllable computational cost, Wang et al. (2002) and Zhang and Liu (2002) considered the *pilot* exploration method, in which the space of future states $\{x_{t+1}, \cdots, x_{t+\Delta}\}$ is partially explored by pilot "paths". The method could be viewed as a low-accuracy Monte Carlo approximation to the exact lookahead sampling method.

The method was introduced for the case of finite state space of $x_t \in \mathcal{A} = \{a_1, \cdots, a_{|\mathcal{A}|}\}$ in both Wang et al. (2002) and Zhang and Liu (2002). Specifically, suppose at time $t-1$ we have a set of samples $\{(\boldsymbol{x}^{(j)}_{t-1}, w^{(j)}_{t-1}), j = 1, \cdots, m\}$ properly weighted with respect to $\pi_{t-1}(\boldsymbol{x}_{t-1})$. For each $\boldsymbol{x}^{(j)}_{t-1}$ and each possible value a_i of x_t, a pilot path $\boldsymbol{x}^{(j,i)}_{t:t+\Delta} = (x^{(j,i)}_t = a_i, x^{(j,i)}_{t+1}, \cdots, x^{(j,i)}_{t+\Delta})$ is constructed sequentially from distribution

$$\prod_{s=t+1}^{t+\Delta} q^{pilot}_s(x_s \mid \boldsymbol{x}^{(j)}_{t-1}, x_t = a_i, \boldsymbol{x}_{t+1:s-1}). \tag{12}$$

Then, $x^{(j)}_t$ can be drawn from a trial distribution that utilizes the "future information" gathered by the pilot samples $\boldsymbol{x}^{(j,i)}_{t+1:t+\Delta}$, $i = 1, \cdots, |\mathcal{A}|$.

Figure 2 illustrates the pilot lookahead sampling operation, with $\mathcal{A} = \{0, 1\}$ and $\Delta = 2$, in which the pilot path for $(\boldsymbol{x}^{(j)}_{t-1}, x_t = 0)$ is $(x_{t+1} = 1, x_{t+2} = 1)$ and the pilot path for $(\boldsymbol{x}^{(j)}_{t-1}, x_t = 1)$ is $(x_{t+1} = 0, x_{t+2} = 1)$, both are shown as a dark path.

The single pilot lookahead algorithm is as follows.

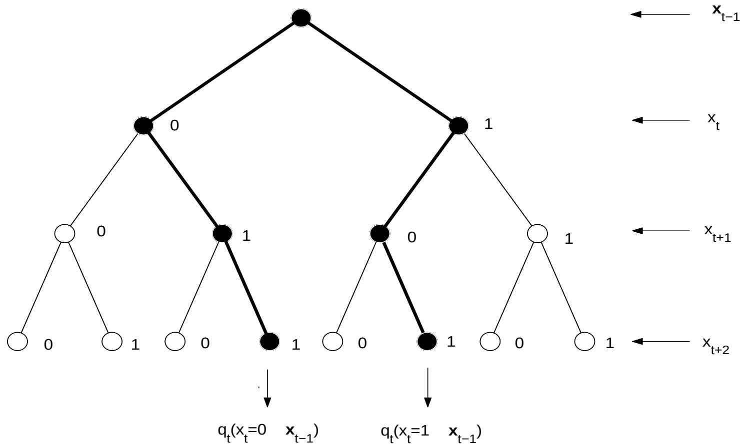

Figure 2: Illustration of the single-pilot lookahead sampling method, in which the pilot path for $(\boldsymbol{x}_{t-1}^{(j)}, x_t = 0)$ is $(x_{t+1} = 1, x_{t+2} = 1)$ and the pilot path for $(\boldsymbol{x}_{t-1}^{(j)}, x_t = 1)$ is $(x_{t+1} = 0, x_{t+2} = 1)$.

Single pilot lookahead algorithm:

- *At time* $t = 0$, *for* $j = 1, \cdots, m$
  - *Draw* $x_0^{(j)}$ *from distribution* $q_0(x_0)$.
  - *Set* $w_0^{(j)} = \pi_0(x_0^{(j)})/q_0(x_0^{(j)})$.
  - *Generate pilot path* $\boldsymbol{x}_{1:\Delta}^{(j,*)}$ *from* $\prod_{s=1}^{\Delta} q_s^{pilot}(x_s \mid x_0^{(j)}, \boldsymbol{x}_{1:s-1})$ *and calculate*

$$w_0^{aux(j)} = w_0^{(j)} \frac{\pi_\Delta(x_0^{(j)}, x_{1:\Delta}^{(j,*)})}{\pi_0(x_0^{(j)}) \prod_{s=1}^{\Delta} q_s^{pilot}(x_s \mid x_0^{(j)}, \boldsymbol{x}_{1:s-1})}.$$

- *At times* $t = 1, 2, \cdots,,$
  - *(Optional) Resample* $\{\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)}, j = 1, \cdots, m\}$ *with priority scores* $\alpha_{t-1}^{(j)} = w_{t-1}^{aux(j)}$.
  - *Propagation: For* $j = 1, \cdots, m$
    - *(Generating Pilots.) For* $x_t = a_i$, $i = 1, \cdots, |\mathcal{A}|$, *draw* $\boldsymbol{x}_{t+1:t+\Delta}^{(j,i)}$ *from (12) and calculate*

$$U_t^{(j,i)} = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid \boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:s-1}^{(j,i)})}. \tag{13}$$

    - *(Sampling.) Draw* $x_t^{(j)}$ *from distribution* $q_t(x_t = a_i \mid \boldsymbol{x}_{t-1}^{(j)}) = \frac{U_t^{(j,i)}}{\sum_{k=1}^{|\mathcal{A}|} U_t^{(j,k)}}$.
    - *(Updating Weights.) We will keep two sets of weights. Let*

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{\pi_t(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)})} \quad \text{and} \quad w_t^{aux(j)} = w_{t-1}^{(j)} \sum_{k=1}^{|\mathcal{A}|} U_t^{(j,k)}.$$

  - *Inference:* $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ *is estimated by*

$$\frac{\sum_{j=1}^{m} w_{t-1}^{(j)} \sum_{i=1}^{|\mathcal{A}|} U_t^{(j,i)} h\left(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i\right)}{\sum_{j=1}^{m} w_t^{aux(j)}}. \tag{14}$$

In the algorithm we maintain two sets of weights. The weight $w_t^{(j)}$ is being updated at each step, and the sample $(\boldsymbol{x}_t, w_t^{(j)})$ is properly weighted with respect to $\pi_t(\boldsymbol{x}_t)$, but not $\pi_{t+\Delta}(\boldsymbol{x}_t)$. A second set of weights, the *auxiliary weight* $w_t^{aux(j)}$, is obtained for resampling and making inference of $E_{\pi_{t+\Delta}}\big(h(\boldsymbol{x}_t)\big)$. We have the following proposition:

Proposition 4. *The weighted sample $(\boldsymbol{x}_t^{(j)}, w_t^{aux(j)})$ obtained by the single pilot lookahead algorithm is properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$, and estimator (14) is a consistent estimator of $E_{\pi_{t+\Delta}}\big(h(\boldsymbol{x}_t)\big)$.*

The proof is given in the Appendix.

The pilot scheme can be quite flexible. For example, multiple pilots can be used for each $(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)$. This would be particularly useful when the size of the state space $\mathcal{A}$ is large. Specifically, for each $(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)$, multiple pilots $\boldsymbol{x}_{t+1:t+\Delta}^{(j,i,k)}$, $k = 1, \cdots, K$, are generated from distribution (12) independently and the corresponding cumulative incremental weights $U_t^{(j,i,k)}$ are calculated by

$$U_t^{(j,i,k)} = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i,k)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i,k)} \mid \boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:s-1}^{(j,i,k)})}.$$

Sample $x_t^{(j)}$ is then generated from distribution

$$q_t(x_t = a_i \mid \boldsymbol{x}_{t-1}^{(j)}) = \frac{\sum_{k=1}^{K} U_t^{(j,i,k)}}{\sum_{i=1}^{|\mathcal{A}|} \sum_{k=1}^{K} U_t^{(j,i,k)}}. \tag{15}$$

The corresponding weight and auxiliary weight are updated by

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{\pi_t(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)})} \quad \text{and} \quad w_t^{aux(j)} = w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} U_t^{(j,i,k)},$$

respectively.

Similar to the conclusion of Proposition 4, samples $(\boldsymbol{x}_t^{(j)}, w_t^{aux(j)})$ is properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$. In addition, we have the following proposition:

Proposition 5. *Suppose sample $\boldsymbol{x}_t^{(1,j)}$ is generated by the exact lookahead sampling algorithm with weight $w_t^{(1,j)}$ as in (7). Denote $(\boldsymbol{x}_t^{(4,j)}, w_t^{(4,j)}, w_t^{aux(j)})$ as the weighted samples from k-pilot lookahead algorithm, $U_t^{(j,i,k)}$ are the cumulative incremental weights, then*

$$0 \le var_{\pi_{t+\Delta}}\left(w_t^{aux(j)}\right) - var_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right) \sim O(1/K)$$

and

$$0 \le var_{\pi_{t+\Delta}}\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} U_t^{(j,i,k)} h\left(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i\right)\right] - var_{\pi_{t+\Delta}}\left[w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right)\right] \sim O(1/K).$$

The proof is in Appendix.

This proposition shows that the variance of the weights under the multiple-pilot lookahead sampling method is larger than that under the exact lookahead sampling method, but converges to the latter at the rate of $1/K$ as the number of pilots K increases. As a consequence, the samples generated by the multiple-pilot lookahead sampling method are more effective than the samples generated by the lookahead weighting method when pilot number K is reasonably large.

When the state space for x_t is continuous, it is infeasible to explore all the possible values of x_t. Finding a more efficient method to carry out lookahead in continuous state space cases is a challenging problem currently under investigation.

One possible approach is the following simple algorithm. For each j, draw multiple samples of $x_t^{(j,i)}$, $i = 1, \cdots, A$, from $q_t(x_t|\boldsymbol{x}_{t-1}^{(j)})$ and treat this set as the space of $x_t^{(j)}$ (the possible values $x_t^{(j)}$ can take). Then we run single or multiple pilots from each of these values, and sample $x_t^{(j)}$ according to the lookahead cumulative incremental weights, just as in the discrete state-space case. In the special case of $A = 1$, the sampling distribution of this lookahead method will be the same as that in the concurrent SMC, but one would use the lookahead weight as the resampling priority score at time t.

An improvement of this approach for the continuous state-space case can be achieved if the dimension of x_t is relatively low and when the state space model is Markovian. That is,

$$g_t(x_t \mid \boldsymbol{x}_{t-1}) = g_t(x_t \mid x_{t-1}) \quad \text{and} \quad f_t(y_t \mid \boldsymbol{x}_t) = f_t(y_t \mid x_t).$$

In this case, the cumulative incremental weight $U_t^{(j,i)}$ of the pilot $(x_t^{(j,i)}, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})$ can be written as

$$\begin{aligned}
U_t^{(j,i)} &= \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(j,i)}, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(j,i)}|x_{t-1}^{(j)}) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid x_{s-1}^{(j,i)})} \\
&\propto \frac{g_t(x_t^{(j,i)}|x_{t-1}^{(j)}) f_t(y_t \mid x_t^{(j,i)}) \prod_{s=t+1}^{t+\Delta} g_s(x_s^{(j,i)} \mid x_{s-1}^{(j,i)}) f_s(y_s \mid x_s^{(j,i)})}{q_t(x_t^{(j,i)}|x_{t-1}^{(j)}) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid x_{s-1}^{(j,i)})} \\
&\triangleq V_t^{(j,i)} V_{t+1:t+\Delta}^{(j,i)},
\end{aligned}$$

where

$$V_t^{(j,i)} = \frac{g_t(x_t^{(j,i)}|x_{t-1}^{(j)}) f_t(y_t \mid x_t^{(j,i)})}{q_t(x_t^{(j,i)}|x_{t-1}^{(j)})} \quad \text{and} \quad V_{t+1:t+\Delta}^{(j,i)} = \frac{\prod_{s=t+1}^{t+\Delta} g_s(x_s^{(j,i)} \mid x_{s-1}^{(j,i)}) f_s(y_s \mid x_s^{(j,i)})}{\prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid x_{s-1}^{(j,i)})}.$$

Standard procedure would choose $x_t^{(j)}$ from the generated $x_t^{(j,i)}$, $i = 1, \cdots, A$, with probability

$U_t^{(j,i)}/\sum_l U_t^{(j,l)}$. However, note that

$$\begin{aligned}\overline{V}_{t+1:t+\Delta}^{(j,i)} &\triangleq E(V_{t+1:t+\Delta}^{(j,i)} \mid \boldsymbol{x}_{t-1}^{(j)}, x_t^{(j,i)}, \boldsymbol{y}_{t+\Delta}) \\ &= \int g(x_{t+1} \mid x_t^{(j,i)}) f(y_{t+1} \mid x_{t+1}) \prod_{s=t+2}^{t+\Delta} g_s(x_s \mid x_{s-1}) f(y_s \mid x_s)\, dx_{t+1} \cdots dx_{t+\Delta},\end{aligned} \tag{16}$$

only depends on $x_t^{(j,i)}$, and

$$V_t^{(j,i)} \overline{V}_{t+1:t+\Delta}^{(j,i)} \propto \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(j,i)} \mid \boldsymbol{x}_{t-1}^{(j)})},$$

is the lookahead cumulative incremental weight in (8), which is shown to be more efficient than $U_t^{(j,i)} = V_t^{(j,i)} V_{t+1:t+\Delta}^{(j,i)}$ as in Proposition 3.

With a Markovian model, $\overline{V}_{t+1:t+\Delta}^{(j,i)}$ is function of $x_t^{(j,i)}$, and $V_{t+1:t+\Delta}^{(j,i)}$ can be considered as a noisy version of $\overline{V}_{t+1:t+\Delta}(x_t^{(j,i)})$. That is, one can write

$$V_{t+1:t+\Delta}^{(j,i)} = \overline{V}_{t+1:t+\Delta}(x_t^{(j,i)}) + e_t^{(j,i)}, \quad \text{where} \quad E(e_t^{(j,i)} \mid x_t^{(j,i)}) = 0.$$

Hence if the dimension of x_t is small, one can smooth $V_{t+1:t+\Delta}^{(j,i)}$ in the space of x_t to obtain an estimate of $\overline{V}_{t+1:t+\Delta}^{(j,i)}$, using all the pilot samples. The estimate is then used for sampling and resampling. For example, let $\hat{V}_{t+1:t+\Delta}^{(j,i)}$ be a nonparametric estimate of $V_{t+1:t+\Delta}^{(j,i)}$ and let $\hat{U}_t^{(j,i)} = V_t^{(j,i)} \hat{V}_{t+1:t+\Delta}^{(j,i)}$. One can choose $x_t^{(j)}$ from $x_t^{(j,i)}$, $i = 1, \cdots, A$, with probability $\hat{U}_t^{(j,i)}/\sum_l \hat{U}_t^{(j,l)}$ and weight it accordingly. Experience shows that a very accurate smoothing method (*e.g.,* kernel smoothing) is not necessary, as to control computational cost. Often a piecewise constant smoother is sufficient.

4.5 Deterministic piloting

It is also possible to use deterministic pilots in the pilot lookahead sampling method. For example, at time t, the pilot starting with $(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)$ for each $a_i \in \mathcal{A}$, can be a future path $\boldsymbol{x}_{t+1:t+\Delta}^{(j,i)}$ that maximizes $\pi_{t+\Delta}(\boldsymbol{x}_{t+1:t+\Delta} \mid \boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)$. Since such a global maximum is usually difficult to obtain, an easily obtainable local maximum is to sequentially, for $s = t+1, \cdots, t+\Delta$, obtain

$$x_s^{(j,i)} = \arg\max_{x_s} \pi_s(x_s \mid \boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:s-1}^{(j,i)}). \tag{17}$$

Once the pilots are drawn, the remaining steps are similar to those in the random pilot algorithm, except that there is usually no easy way to obtain a proper weight with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$, though a proper weight with respect to $\pi_t(\boldsymbol{x}_t)$ is easily available. In order to make proper inference with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$, one can generate an additional random pilot path to $x_{t+\Delta}$. Specifically, we have the following scheme.

Deterministic pilot lookahead algorithm:

- *At time* $t=0$, *for* $j=1,\cdots,m$
  - *Draw* $x_0^{(j)}$ *from distribution* $q_0(x_0)$.
  - *Set* $w_0^{(j)}=\pi_0(x_0^{(j)})/q_0(x_0^{(j)})$.
  - *Generate deterministic pilots* $\boldsymbol{x}_{1:\Delta}^{(j,*)}$ *sequentially by letting*

$$x_s^{(j,*)}=arg\max_{x_s}\pi_s(x_s\mid \boldsymbol{x}_{\Delta}^{(j)},\boldsymbol{x}_{1:s-1}^{(j,*)})$$

  for $s=1,\cdots,\Delta$. *Let* $U_0^{(j,*)}=\pi_\Delta(x_0^{(j)},\boldsymbol{x}_{1:\Delta}^{(j,*)})/\pi_0(x_0^{(j)})$.
  - *Set* $w_0^{res(j)}=w_0^{(j)}U_0^{(j,*)}$.

- *At times* $t=1,2,\cdots,,$
  - *(Optional) Resample* $\{(\boldsymbol{x}_{t-1}^{(j)},w_{t-1}^{(j)}),j=1,\cdots,m\}$ *with priority scores* $\alpha_{t-1}^{(j)}=w_{t-1}^{res(j)}$.
  - *Propagation: For* $j=1,\cdots,m$
    - *(Generating Deterministic Pilots.) For* $x_t=a_i$, $i=1,\cdots,|\mathcal{A}|$, *obtain* $\boldsymbol{x}_{t+1:t+\Delta}^{(j,i)}$ *sequentially using (17) for* $s=t+1,\cdots,t+\Delta$.
    - *(Sampling.) Draw* $x_t^{(j)}$ *from distribution* $q_t(x_t=a_i\mid \boldsymbol{x}_{t-1}^{(j)})=U_t^{(j,i)}/\sum_{i=1}^{|\mathcal{A}|}U_t^{(j,i)}$, *where*

$$U_t^{(j,i)} = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)},x_t=a_i,\boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}.$$

    - *(Updating Weights.) We keep three sets of weights for concurrent weighting, resampling, and estimation.*
      (1) Concurrent weight: $w_t^{(j)}=w_{t-1}^{(j)}\frac{\pi_t(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})q_t(x_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}$;
      (2) Resampling weight $w_t^{res(j)}=w_{t-1}^{(j)}\sum_{i=1}^{|\mathcal{A}|}U_t^{(j,i)}$;
      (3) Auxiliary weight: draw $\boldsymbol{x}_{t+1:t+\Delta}^{aux(j)}$ *from* $\prod_{s=t+1}^{t+\Delta}q_s^{aux}(x_s\mid \boldsymbol{x}_{t-1}^{(j)},x_t^{(j)},\boldsymbol{x}_{t+1:s-1})$ *and calculate*

$$w_t^{aux(j)}=w_t^{(j)}\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)},x_t^{(j)},\boldsymbol{x}_{t+1:t+\Delta}^{aux(j)})}{\pi_t(\boldsymbol{x}_t^{(j)})\prod_{s=t+1}^{t+\Delta}q_s^{aux}(x_s^{aux(j)}\mid \boldsymbol{x}_{t-1}^{(j)},x_t^{(j)},\boldsymbol{x}_{t+1:s-1}^{aux(j)})}.$$

  - *Inference:* $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ *is estimated by* $\sum_{j=1}^m w_t^{aux(j)}h(\boldsymbol{x}_t^{(j)})/\sum_{j=1}^m w_t^{aux(j)}$.

The above algorithm requires the generation of additional random pilot $\boldsymbol{x}_{t+1:t+\Delta}^{aux(j)}$ to obtain $w_t^{aux(j)}$, which is properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{x}_t)$. Alternatively, one can combine the deterministic pilot scheme and the lookahead weighting method in Section 4.1 to obtain a consistent estimate of $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$.

The resampling weight $w_t^{res(j)}$ is served as the priority score for resampling when needed. It retains the information from the deterministic pilot and avoids the additional random variation from the additional sample path required by the auxiliary weight $w_t^{aux(j)}$.

The deterministic pilots are useful because they gather future information to guide the generation of the current state x_t. In some cases, the deterministic pilots can provide a better approximation of the distribution $\pi_{t+\Delta}(x_t \mid \boldsymbol{x}_{t-1}^{(j)})$ than the random pilots, especially when we can only afford to use a single pilot for each $(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)$. In addition, with some proper approximation, the deterministic pilot scheme may have lower computational complexity. The example in Section 6.1 uses a low complexity method to generate the deterministic pilots.

4.6 Multilevel Pilot Lookahead Sampling

In case of finite state space, when the size of the state space $\mathcal{A}$ is large, the pilot lookahead sampling method can still be too expensive. To reduce the computational cost, we introduce a multilevel method, which constructs a hierarchical structure in the state space and utilizes lookahead idea within the structure. Guo et al. (2004) developed a similar algorithm.

Specifically, at time t, we first divide the current state space $\mathcal{A}$ of x_t into disjoint subspaces on $L+1$ different levels, that is

$$\mathcal{A} = \mathcal{C}_{l,1} \cup \mathcal{C}_{l,2} \cup \cdots \cup \mathcal{C}_{l,D_l} \qquad l = 0, \cdots, L.$$

In the division, each level-l subspace $\mathcal{C}_{l,i}$ consists of several level-$(l+1)$ sets $\mathcal{C}_{l+1,j}$. On the top level (level-0), $\mathcal{C}_{0,1} = \mathcal{A}$. On the lowest level (level-L), each $\mathcal{C}_{L,i}$ only contains a single state value $a_i \in \mathcal{A}$. For example, in a 16-QAM wireless communication problem (Guo et al., 2004), the transmitted signal x_t to be decoded takes values in space $\mathcal{A} = \{a_i = (a_{i,1}, a_{i,2}) : a_{i,1}, a_{i,2} = \pm 1, \pm 2\}$. Figure 3 depicts a multilevel scheme where the state space is divided into three levels ($L = 2$)

$$\mathcal{A} = \mathcal{C}_{0,1} = \mathcal{C}_{1,1} \cup \mathcal{C}_{1,2} \cup \mathcal{C}_{1,3} \cup \mathcal{C}_{1,4} = \cup_{i=1}^{16} \mathcal{C}_{2,i} = \cup_{i=1}^{16} \{a_i\}.$$

At time t, instead of sampling $x_t^{(j)}$ directly, we generate a length L index sequence $\{I_{t,1}^{(j)}, \cdots, I_{t,L}^{(j)}\}$, in which $I_{t,l}^{(j)}$ indicates that $x_t^{(j)}$ belongs to level-l subsets $\mathcal{C}_{l,I_{t,l}^{(j)}}$. A valid index sequence $\{I_{t,1}^{(j)}, \cdots, I_{t,L}^{(j)}\}$ needs to satisfy $\mathcal{C}_{l,I_{t,l}^{(j)}} \subset \mathcal{C}_{l-1,I_{t,l-1}^{(j)}}$, $l = 1, \cdots, L$. The last indicator $I_{t,L}^{(j)}$ specifies the value of $x_t^{(j)}$ as the level-L subset $\mathcal{C}_{L,I_{t,L}^{(j)}}$ only contains one state value.

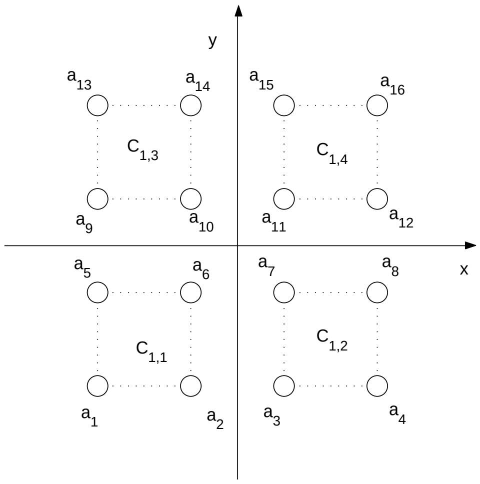

Figure 3: Illustration of multilevel structure in 16-QAM modulation.

The index sequence $\{I_{t,1}^{(j)}, \cdots, I_{t,L}^{(j)}\}$ is generated sequentially, starting from the highest level, following the trial distribution

$$\prod_{l=1}^{L} q_{t,l}(I_{t,l} \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l-1}).$$

Here we define $I_{t,0} \equiv 1$, which coincides with $x_t \in \mathcal{C}_{0,1} \equiv \mathcal{A}$. The index sampling distribution $q_{t,l}(I_{t,l} \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l-1})$ can be constructed as follows, using a pilot scheme.

For every i such that $\mathcal{C}_{l,i} \subset \mathcal{C}_{l-1,I_{t,l-1}}$, randomly draw a pilot path $(x_t^{(j,i)}, x_{t+1}^{(j,i)}, \cdots, x_{t+\Delta}^{(j,i)})$ from the trial distribution

$$q_t^{pilot}(x_t \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l} = i) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:s-1}) \tag{18}$$

where $q_t^{pilot}(x_t \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l} = i)$ indicates that x_t must be a member of $\mathcal{C}_{l,i}$, and calculate

$$U_{t,l}^{(j,i)} = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:t+\Delta}^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t^{pilot}(x_t^{(j,i)} \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l} = i) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{x}_{t:s-1}^{(j,i)})} \tag{19}$$

Then sample $I_{t,l}^{(j)}$ is generated from distribution

$$q_{t,l}(I_{t,l} = i \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l-1}^{(j)}) = \frac{U_{t,l}^{(j,i)}}{\sum_{k:\, \mathcal{C}_{l,k} \subset \mathcal{C}_{l-1,I_{t,l-1}}} U_{t,l}^{(j,k)}}. \tag{20}$$

Specifically, the algorithm is as follows.

Multilevel pilot algorithm:

- *At time* $t = 0$, *for* $j = 1, \cdots, m$
  - *Draw* $x_0^{(j)}$ *from distribution* $q_0(x_0)$.
  - *Set* $w_0^{(j)} = \pi_0(x_0^{(j)})/q_0(x_0^{(j)})$.
  - *Generate pilot path* $\boldsymbol{x}_{1:\Delta}^{(j,*)}$ *from* $\prod_{s=1}^{\Delta} q_s^{pilot}(x_s \mid x_0^{(j)}, \boldsymbol{x}_{1:s-1})$ *and calculate*

$$U_0^{(j,*)} = \frac{\pi_\Delta(x_0^{(j)}, x_{1:\Delta}^{(j,*)})}{\pi_0(x_0^{(j)}) \prod_{s=1}^{\Delta} q_s^{pilot}(x_s \mid x_0^{(j)}, \boldsymbol{x}_{1:s-1})}.$$

  - *Set* $w_0^{aux(j)} = w_0^{(j)} U_0^{(j,*)}$.

- *At time* $t = 1, 2, \cdots,$
  - *(Optional) Resample* $\{\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)}, j = 1, \cdots, m\}$ *with priority scores* $\alpha_{t-1}^{(j)} = w_{t-1}^{aux(j)}$.
  - *Propagation: For* $j = 1, \cdots, m,$
    - *Set* $I_{t,0}^{(j)} \equiv 1$. *For level* $l = 1, 2, \cdots, L,$
      - *(Generating Pilots.) For each* i *such that* $\mathcal{C}_{l,i} \subset \mathcal{C}_{l-1, I_{t,l-1}^{(j)}}$, *generate pilot* $(x_t^{(j,i)}, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})$ *from distribution (18) and* $U_{t,l}^{(j,i)}$ *is calculated as in (19).*
      - *(Sampling.) Draw* $I_{t,l-1}^{(j)}$ *from the trial distribution (20).*
    - *(Updating Weights.) If* $x_t^{(j)} = a_{i_0}$ *is chosen at last, i.e.,* $\mathcal{C}_{L, I_{t,L}^{(j)}} = \{a_{i_0}\}$, *let*

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \frac{\pi_t(\boldsymbol{x}_t^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{l=1}^{L} q_{t,l}(I_{t,l}^{(j)} \mid I_{t,l-1}^{(j)}, \boldsymbol{x}_{t-1}^{(j)})}, \\ w_t^{aux(j)} &= w_{t-1}^{(j)} \frac{U_{t,L}^{(j,i_0)}}{\prod_{l=1}^{L} q_{t,l}(I_{t,l}^{(j)} \mid I_{t,l-1}^{(j)}, \boldsymbol{x}_{t-1}^{(j)})}. \end{aligned}$$

  - *Inference:* $E_{\pi_{t+\Delta}}(h(\boldsymbol{x}_t))$ *is estimated by* $\sum_{j=1}^{m} w_t^{aux(j)} h(\boldsymbol{x}_t^{(j)}) / \sum_{j=1}^{m} w_t^{aux(j)}$.

The advantage of the multilevel method is that it reduces the total number of probability calculations involved in generating $x_t^{(j)}$. For example, generating $x_t^{(j)}$ directly from trial distribution $q_t(x_t \mid \boldsymbol{x}_{t-1}^{(j)})$ requires a total of $|\mathcal{A}|$ evaluations of $q_t(x_t = a_i \mid \boldsymbol{x}_{t-1}^{(j)})$, $i = 1, \cdots, |\mathcal{A}|$. On the other hand, generating $\{I_{t,1}^{(j)}, \cdots, I_{t,L}^{(j)}\}$ only requires $\sum_{l=1}^{L} n(I_{t,l-1}^{(j)})$ such evaluations, where $n(I_{t,l-1}^{(j)})$ is the number of level-l subsets contained in level-$(l-1)$ subset $\mathcal{C}_{l-1, I_{t,l-1}^{(j)}}$. In the example illustrated by Figure 3, $I_{t,1}^{(j)}$ is chosen from a set of four subgroups at the first step. Given a selected $I_{t,1}^{(j)}$, $I_{t,2}^{(j)}$ is drawn from a set of four elements under $I_{t,1}^{(j)}$. Hence, $n(I_{t,0}) = 4$ and $n(I_{t,1}) = 4$. In this example, a total of 8 probabilities need to be evaluated, reduced from 16 if $x_t^{(j)}$ were generated directly. More generally, if $|\mathcal{A}| = 4^L$, we can reduce the computation to $4L$ evaluations based on such a multilevel structure.

As discussed in Section 4.5, deterministic pilot can also be used in the multilevel method. A multilevel pilot lookahead sampling method using deterministic pilot is applied to the signal detection example in Section 6.1.

4.7 Resampling with lookahead and piloting

As discussed in Liu and Chen (1995, 1998), although a resampling step introduces additional Monte Carlo variations for estimating the current state, it enables the sampler to focus on important regions of "future" spaces and can improve the effectiveness of samples in future steps. Liu and Chen (1998) suggested that one can perform resampling according to either a deterministic schedule or an adaptive schedule. In the following, we consider the problem of finding the optimal resampling priority score if resampling only takes place at time $T, 2T, 3T, \cdots$ (i.e., a deterministic schedule).

Suppose we perform a standard SMC procedure. At time $t = nT$, samples $\{(\boldsymbol{x}_t^{(j)}, w_t^{(j)}), j = 1, \cdots, m\}$ properly weighted with respect to $\pi_t(\boldsymbol{x}_t)$ are generated, in which $\boldsymbol{x}_t^{(j)}$ follows the distribution $r_t(\boldsymbol{x}_t)$, and $w_t^{(j)} = w_t(\boldsymbol{x}_t^{(j)}) = \pi_t(\boldsymbol{x}_t^{(j)})/r_t(\boldsymbol{x}_t^{(j)})$. We perform a resampling step with priority score $b(\boldsymbol{x}_t^{(j)})$, then the new samples $\boldsymbol{x}_t^{*(j)}, j = 1, \cdots, m$, approximately follow the distribution $\psi(\boldsymbol{x}_t)$ that is proportional to $r_t(\boldsymbol{x}_t)b(\boldsymbol{x}_t)$. In the following T steps, $x_{t+1}^{*(j)}, \cdots, x_{t+T}^{*(j)}$ is generated sequentially from distribution $q_s(x_s \mid \boldsymbol{x}_{s-1}^{*(j)})$, $s = t+1, \cdots, t+T$, then the corresponding weight of $\boldsymbol{x}_{t+T}^{*(j)}$ with respect to $\pi_{t+T}(\boldsymbol{x}_{t+T})$ is

$$\begin{aligned} w_{t+T}(\boldsymbol{x}_{t+T}^{*(j)}) &= \frac{\pi_t(\boldsymbol{x}_t^{*(j)})}{\psi_t(\boldsymbol{x}_t^{*(j)})} \frac{\pi_{t+T}(\boldsymbol{x}_{t+T}^{*(j)})}{\pi_t(\boldsymbol{x}_t^{*(j)}) \prod_{s=t+1}^{t+T} q_s(x_s^{*(j)} \mid \boldsymbol{x}_{s-1}^{*(j)})} \\ &\propto \frac{\pi_t(\boldsymbol{x}_t^{*(j)})}{r_t(\boldsymbol{x}_t^{*(j)}) b_t(\boldsymbol{x}_t^{*(j)})} \frac{\pi_{t+T}(\boldsymbol{x}_{t+T}^{*(j)})}{\pi_t(\boldsymbol{x}_t^{*(j)}) \prod_{s=t+1}^{t+T} q_s(x_s^{*(j)} \mid \boldsymbol{x}_{s-1}^{*(j)})}. \end{aligned}$$

The following proposition concerns the choice of priority score $b(\boldsymbol{x}_t)$ that minimizes the variance of weight $w_{t+T}(\boldsymbol{x}_{t+T}^{*(j)})$.

Proposition 6. *The variance of weight $w_{t+T}(\boldsymbol{x}_{t+T}^{*(j)})$ is minimized when*

$$b_t(\boldsymbol{x}_t) \propto w_t(\boldsymbol{x}_t) \eta_{t,T}^{1/2}(\boldsymbol{x}_t). \tag{21}$$

where

$$\eta_{t,T}(\boldsymbol{x}_t) = \int \left[\frac{\pi_{t+T}(\boldsymbol{x}_{t+T})}{\pi_t(\boldsymbol{x}_t) \prod_{s=t+1}^{t+T} q_s(x_s \mid \boldsymbol{x}_{s-1})} \right]^2 \prod_{s=t+1}^{t+T} q_s(x_s \mid \boldsymbol{x}_{s-1}) \, dx_{t+1} \ldots dx_{t+T}.$$

The proof is in the Appendix.

Specifically, if we perform resampling at every step ($T = 1$), and the trial distribution is $q_s(x_s \mid \boldsymbol{x}_{s-1}) = \pi_s(x_s \mid \boldsymbol{x}_{s-1})$, the optimal priority score becomes

$$b_t(\boldsymbol{x}_t) = w_t(\boldsymbol{x}_t)\frac{\pi_{t+1}(\boldsymbol{x}_t)}{\pi_t(\boldsymbol{x}_t)},$$

which is the priority score used in the sequential imputation of Kong et al. (1994) and Liu and Chen (1995), and the auxiliary particle filter proposed by Pitt and Shephard (1999).

When $T > 1$, the exact value of $\eta_{t,T}(\boldsymbol{x}_t)$ in (21) is difficult to calculate. In this case, one can use pilot method to find an approximation. For each sample $\boldsymbol{x}_t^{(j)}$, multiple pilots $\boldsymbol{x}_{t+1:t+T}^{(j,i)}$, $i = 1, \cdots, K$, are generated following distribution $\prod_{s=t+1}^{t+T} q_s(x_s \mid \boldsymbol{x}_{s-1}^{(j,i)})$ with the cumulative incremental weight

$$U_t^{(j,i)} = \frac{\pi_{t+T}(\boldsymbol{x}_t^{(j)}, \boldsymbol{x}_{t+1:t+T}^{(j,i)})}{\pi_t(\boldsymbol{x}_t^{(j)}) \prod_{s=t+1}^{t+T} q_s(x_s^{(j,i)} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{x}_{t+1:s-1}^{(j,i)})}.$$

Then $\eta(\boldsymbol{x}_t^{(j)})$ can be estimated by $K^{-1}\sum_{i=1}^{K}\left(U_t^{(j,i)}\right)^2$.

4.8 Combined methods

The lookahead schemes discussed so far can be combined to further improve the efficiency. For example, Wang et al. (2002) considered a combination of the exact lookahead sampling and the pilot lookahead sampling methods. In this approach, the space of the immediate future states is explored exhaustively, and the space of further future states is explored using pilots.

5 Adaptive Lookahead

Many systems have structures with different local complexity. In these systems, it may be beneficial to have different lookahead schemes based on local information. For example, in one of the wireless communication applications, the received signal y_t can be considered as following

$$y_t = \xi_t x_t + v_t,$$

where $\{v_t\}$ is white noise with variance σ^2, $\{x_t\}$ is the transmitted discrete symbol sequence, $\{\xi_t\}$ is the fading channel coefficient that varies over time. Since $\{\xi_t\}$ varies, the signal-to-noise ratio in the system also changes. When $|\xi_t|$ is large, the current observation $\boldsymbol{y}_t$ contains sufficient information to decode x_t accurately. In this case, lookahead is not needed. When $|\xi_t|$ is small, the signal-to-noise ratio is low and lookahead becomes very important to bring in future observations to help the estimation of ξ_t and x_t.

Lookahead strategies always result in a better estimator provided that the Monte Carlo sample size is sufficiently large so that $I(\Delta)$ in (2) is negligible. To control computational cost, however, Monte Carlo sample size used may not be large enough to make $I(\Delta)$ negligible. For a fixed sample size, $I(\Delta)$ can increase as Δ increases. Hence, it is possible that lookahead make the performance worse with finite Monte Carlo sample size. The following proposition provides the condition under which one additional lookahead step in the pilot lookahead sampling method makes the estimator less accurate.

Specifically, suppose in a finite state system, a sample set $\{(\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)}), j = 1, \cdots, m\}$ properly weighted with respect to $\pi_{t-1}(\boldsymbol{x}_{t-1})$ is available at time $t-1$. At time t, Δ-step pilots $\boldsymbol{x}_{t+\Delta}^{(j,i)} = (\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i, \boldsymbol{x}_{t+1:t+\Delta}^{(j,i)})$, $j = 1, \cdots, m$, $i = 1, \cdots, \mathcal{A}$, are generated from distribution (12) with cumulative incremental weight

$$U_{t,\Delta}^{(j,i)} = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}^{(j,i)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{s=t+1}^{t+\Delta} q_s^{pilot}(x_s^{(j,i)} \mid \boldsymbol{x}_{s-1}^{(j,i)}, \boldsymbol{y}_s)}.$$

Then the Δ-step pilot lookahead sampling estimator of $h(\boldsymbol{x}_t)$ is

$$\widehat{h} = \frac{1}{m} \sum_{j=1}^{m} w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i) \rightarrow E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}).$$

If we lookahead one more step and draw $x_{t+\Delta+1}^{(j,i)}$ from trial distribution $q_{t+\Delta+1}(x_{t+\Delta+1} \mid \boldsymbol{x}_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta+1})$, the proper $(\Delta+1)$-step cumulative incremental weight is

$$U_{t,\Delta+1}^{(j,i)} = U_{t,\Delta}^{(j,i)} \frac{\pi_{t+\Delta+1}(\boldsymbol{x}_{t+\Delta+1}^{(j,i)})}{\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}^{(j,i)})\, q_{t+\Delta+1}(x_{t+\Delta+1}^{(j,i)} \mid \boldsymbol{x}_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta+1})}.$$

Then the $(\Delta+1)$-step pilot lookahead sampling estimator of $h(\boldsymbol{x}_t)$ is

$$\widehat{h}^* = \frac{1}{m} \sum_{j=1}^{m} w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i) \rightarrow E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}).$$

Proposition 7. *Let* $\boldsymbol{x}_{t+\Delta}^{(j,i=1:\mathcal{A})} = \{\boldsymbol{x}_{t+\Delta}^{(j,i)}, i = 1, \cdots, \mathcal{A}\}$ *and suppose* $\boldsymbol{x}_{t-1}^{(j)}$, $j = 1, \cdots, m$, *are i.i.d. given* $\boldsymbol{y}_{t+\Delta}$. *When*

$$\begin{aligned} \frac{1}{m} E\left[var\left(w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \mid \boldsymbol{x}_{t+\Delta}^{(j,i=1:\mathcal{A})}, \boldsymbol{y}_{t+\Delta} \right) \middle| \boldsymbol{y}_{t+\Delta} \right] \\ \geq \left(1 + \frac{1}{m}\right) var\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta} \Big], \end{aligned} \tag{22}$$

we have

$$E\left[\left(\widehat{h}^* - h(\boldsymbol{x}_t)\right)^2 \middle| \boldsymbol{y}_{t+\Delta} \right] \geq E\left[\left(\widehat{h} - h(\boldsymbol{x}_t)\right)^2 \middle| \boldsymbol{y}_{t+\Delta} \right].$$

The proof is in the Appendix.

Condition (22) may be difficult to check in practice. However, when $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+\Delta}) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+\Delta+1})$, *i.e.*, $y_{t+\Delta+1}$ is independent of the current state $\boldsymbol{x}_t$ given $\boldsymbol{y}_{t+\Delta}$, the condition always holds since $var\left[E\big(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\big) \,\Big|\, \boldsymbol{y}_{t+\Delta}\right] = 0$.

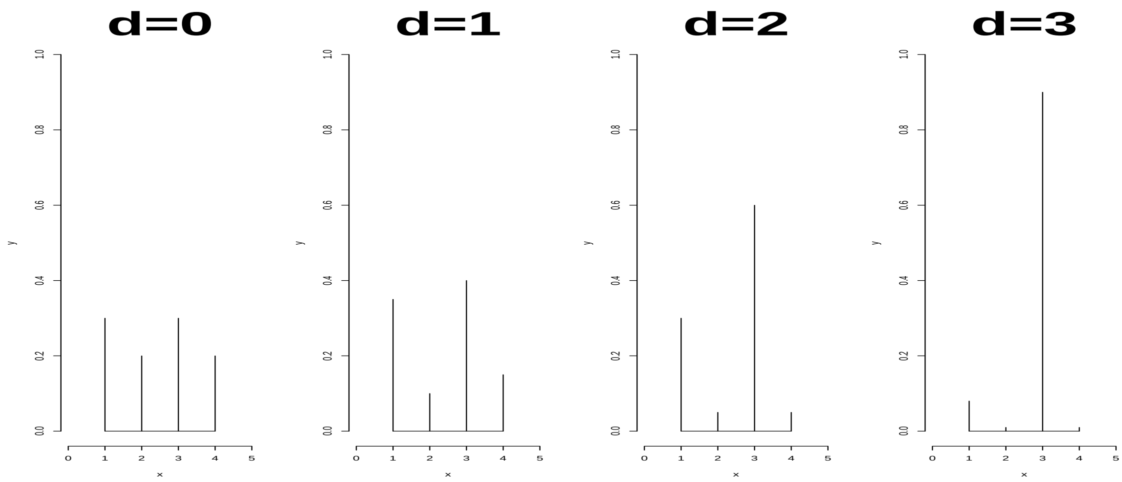

Figure 4: Illustration of adaptive lookahead criterion.

Proposition 7 suggests that, with a fixed number of samples, the performance of the SMC estimator can be optimized by choosing a proper lookahead step. Here we use a heuristic criteria, depicted in Figure 4. Suppose that the state space of x_t takes four possible values and the distribution $\pi_{t+d}(x_t)$ for different lookahead $d = 0, 1, 2, 3$ is as shown in Figure 4, we can conclude that the information available at t (*i.e.* $\boldsymbol{y}_t$) is not sufficiently strong for making inference on x_t, and the samples we generate for x_t at this time ($d = 0$) may not be useful as the system propagates. However, as d increases, the distribution becomes less diffused, showing the accumulation of information about x_t from the future y_{t+d}. It also shows that further lookahead beyond $d = 3$ is probably not necessary. The details of this adaptive criteria is as follows

- In a finite state space model, consider lookahead steps $\Delta = 0, 1, 2, \cdots$. Stop if $\Delta \geq N$ or the estimated posterior distribution satisfies

$$max_i\Big\{\widehat{\pi}_{t+\Delta}(x_t = a_i)\Big\} = max_i\Big\{\frac{\sum_j w_{t-1}^{(j)} U_{t,\Delta}^{(j,i)}}{\sum_{l,j} w_{t-1}^{(j)} U_{t,\Delta}^{(j,l)}}\Big\} > p_0, \tag{23}$$

  where N is the maximum number of lookahead steps we will perform, $0 < p_0 < 1$ is a threshold close to 1, and $U_t^{(j,i)}$ are the cumulative incremental weights defined in (13).

- In a continuous state space model, try lookahead steps $\Delta = 0, 1, 2, \cdots$. Stop if $\Delta \geq N$ or the estimated variance $var_{\pi_{t+\Delta}}(x_t)$ satisfies

$$\widehat{var}_{\pi_{t+\Delta}}(x_t) = \frac{\sum_{i,j} w_{t-1}^{(j)} U_{t,\Delta}^{(j,i)} \left(x_t^{(j,i)}\right)^2}{\sum_{i,j} w_{t-1}^{(j)} U_{t,\Delta}^{(j,i)}} - \left(\frac{\sum_{i,j} w_{t-1}^{(j)} U_{t,\Delta}^{(j,i)} x_t^{(j,i)}}{\sum_{i,j} w_{t-1}^{(j)} U_{t,\Delta}^{(j,i)}}\right)^2 < \sigma_0^2, \qquad (24)$$

where σ_0^2 is a given threshold, $x_t^{(j,i)}$ are samples of current state generated from each $\boldsymbol{x}_{t-1}^{(j)}$ under the pilot scheme, $U_{t,\Delta}^{(j,i)}$ are the corresponding cumulative incremental weights.

Some examples of using adaptive lookahead in finite state space model and continuous state space model are presented in Sections 6.

6 Applications

In this section, we demonstrate the property of lookahead and make performance comparisons. In all cases, δ, Δ, and Δ' are used to denote the numbers of lookahead steps in lookahead weighting, exact lookahead sampling and pilot lookahead sampling, respectively.

6.1 Signal Detection over Flat-fading Channel

In a digital wireless communication problem (Chen and Liu, 2000; Wang et al., 2002), the received signal sequence $\{y_t\}$ is modelled as

$$y_t = \xi_t x_t + v_t,$$

where $\{x_t\}$ is the transmitted complex digital symbol sequence, $\{v_t\}$ is white complex Gaussian noise with variance σ^2 and independent real and complex components, $\{\xi_t\}$ is the transmitted channel, which can be modelled as an ARMA process

$$\xi_t + \phi_1 \xi_{t-1} + \cdots + \phi_r \xi_{t-r} = \theta_0 u_t + \theta_1 u_{t-1} + \cdots + \theta_r u_{t-r},$$

where $\{u_t\}$ is a unit white complex Gaussian noise. In this example, we assume $\{\xi_t\}$ follows the ARMA(3,3) process (Guo et al., 2004)

$$\begin{aligned} & \xi_t - 2.37409\xi_{t-1} + 1.92936\xi_{t-2} - 0.53208\xi_{t-3} \\ = \; & 10^{-2}\Big(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} + 0.89409u_{t-3}\Big). \end{aligned}$$

This system can be turned into a conditional dynamic linear model (CDLM) as follows

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{F}\boldsymbol{z}_{t-1} + \boldsymbol{g}u_t, \\ y_t &= \xi_t x_t + v_t = \boldsymbol{h}^H \boldsymbol{z}_t \, x_t + v_t, \end{aligned}$$

where

$$\boldsymbol{F} = \begin{pmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \quad \boldsymbol{g} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$\boldsymbol{h} = [\theta_0 \, \theta_1 \, \cdots \, \theta_r]^H.$$

Here we consider a high-constellation system with 256-QAM modulation, thus the symbol space is $\mathcal{A} = \{a_i = (a_{i,1}, a_{i,2}) : a_{i,1}, a_{i,2} = \pm 1, \pm 3, \cdots, \pm 15\}$, where $a_{i,1}$ and $a_{i,2}$ are the real and imaginary parts of symbol a_i, respectively. We decode $\{x_t\}$ from received $\{y_t\}$ under the framework of mixture Kalman filter of Chen and Liu (2000) and the "optimal-resampling" scheme of Fearnhead and Clifford (2003).

Because the symbol space is large ($|\mathcal{A}| = 256$), we use a combination of the multilevel pilot lookahead sampling method and the lookahead weighting method. The multilevel structure used is similar to that of 16-QAM presented in Figure 3. The symbol space is divided into subspaces of five different levels ($L = 4$). Hence, at time t, we generate $(I_{t,1}^{(j)}, I_{t,2}^{(j)}, I_{t,3}^{(j)}, I_{t,4}^{(j)})$ to obtain $x_t^{(j)}$ for given $\boldsymbol{x}_{t-1}^{(j)}$ sequentially.

To construct the conditional trial distribution $q_{t,l}(I_{t,l} \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l-1}^{(j)})$, we generate deterministic pilot $(x_t^{(j,I_{t,l})}, \cdots, x_{t+\Delta'}^{(j,I_{t,l})})$ for every possible $I_{t,l}$ given $(\boldsymbol{x}_{t-1}^{(j)}, I_{t,1}^{(j)}, \cdots, I_{t,l-1}^{(j)})$ generated. The steps to generate deterministic pilot are as follows:

- Predict channel ξ_t by $\widehat{\xi}_t^{(j)} = E(\xi_t \mid \boldsymbol{x}_{t-1}^{(j)}, Y_{t-1})$. Let $x_t^{(j,I_{t,l})}$ be the symbol $a_i \in \mathcal{C}_{l,I_{t,l}}$ closest to $y_t / \widehat{\xi}_t^{(j)}$.
- For $s = t+1, \cdots, t+\Delta'$, repeat the following:
  - Predict channel ξ_s by $\widehat{\xi}_s^{(j,I_{t,l})} = E(\xi_s \mid \boldsymbol{x}_{t-1}^{(j)}, x_t^{(j,I_{t,l})}, \cdots, x_{s-1}^{(j,I_{t,l})}, \boldsymbol{y}_{s-1})$.
  - Choose symbol $a_i \in \mathcal{A}$ closest to $y_s / \widehat{\xi}_s^{(j,I_{t,l})}$ as $x_s^{(j,I_{t,l})}$.

Let $U_t^{(j,I_{t,l})} = \pi_{t+\Delta'}(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(j,I_{t,l})}, \boldsymbol{x}_{t+1:t+\Delta'}^{(j,I_{t,l})}) / \pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})$, the trial distribution is

$$q_{t,l}(I_{t,l} \mid \boldsymbol{x}_{t-1}^{(j)}, I_{t,l-1}^{(j)}) = \frac{U_t^{(j,I_{t,l})}}{\sum_{k:\, \mathcal{C}_{l,k} \subset \mathcal{C}_{l-1, I_{t,l-1}^{(j)}}} U_t^{(j,k)}}$$

For comparison, SMC without using multilevel structure and lookahead pilot is also considered. More computational details of this problem can be found in Wang et al. (2002).

In the simulation, the length of transmitted symbol sequences is 500. To avoid phase ambiguities, differential decoding is used. Specifically, suppose the information symbol sequence is $\{d_t\}$. The actual transmitted symbol sequence $\{x_t\}$ is constructed as follows: Given the 256 QAM transmitted symbol x_{t-1} and information symbol d_t, we first map them to four QPSK symbols $(r_{x_{t-1},1}, r_{x_{t-1},2}, r_{x_{t-1},3}, r_{x_{t-1},4})$ and $(r_{d_t,1}, r_{d_t,2}, r_{d_t,3}, r_{d_t,4})$ respectively. Let $r_{x_t,i} = r_{d_t,i} r_{x_{t-1},i}$, $i = 1, 2, 3, 4$, and we map these four QPSK symbols $(r_{x_t,1}, r_{x_t,2}, r_{x_t,3}, r_{x_t,4})$ back to 256-QAM as the transmitted symbol x_t. The differential receiver calculates $r_{\widehat{d}_t,i} = r_{\widehat{x}_t,i} r^*_{\widehat{x}_{t-1},i}$, where $(\widehat{x}_{t-1}, \widehat{x}_t)$ are estimated (x_{t-1}, x_t) at the receiver, then decodes the information symbol d_t as the 256-QAM symbol corresponding to $(r_{\widehat{d}_t,1}, r_{\widehat{d}_t,2}, r_{\widehat{d}_t,3}, r_{\widehat{d}_t,4})$. To improve the decoding accuracy of this high-constellation system, we also insert 10% symbols that are known to the receiver into the transmitted symbol sequences periodically. The experiment is repeated 100 times. Total 50,000 symbols (400,000 bit information) are decoded.

Figure 5 reports the bit-error-ratio (BER) performance of different lookahead step δ of the lookahead weighting method with standard concurrent SMC sampling ($\Delta = 0, \Delta' = 0$). $m = 200$ samples are used. It is seen that the BER performance does not improve further after $\delta \geq 8$ lookahead steps. We use $\delta = 10$ in the following comparison.

BER performance of pilot lookahead sampling methods with different lookahead steps Δ' is shown in Figure 6. The number of Monte Carlo samples is adjusted so that each method takes approximately the same CPU time. From the result, it is seen that the multilevel pilot lookahead sampling method with $\Delta' = 1$ has smaller BER than SMC without using lookahead pilot. But when we use $\Delta' = 2$, the performance is worse. One of the reasons is that we use predicted channel to construct the pilot, which could be very different from the true channel and severely mislead the sampling, especially when the number of lookahead steps is large. We also implement the adaptive method. Here we use adaptive stop criteria (23) with $p_0 = 0.90$. The resulting average number of lookahead steps is 0.195. Due to the saving in smaller number of lookahead steps, larger Monte Carlo sample size is used with the same computational time. Its BER performance is slightly better than using fixing pilot lookahead step $\Delta' = 1$.

6.2 Nonlinear Filtering

Consider the following nonlinear state space model (Gordon et al., 1993):

$$\begin{aligned} \text{state equation} \quad &: \quad x_t = 0.5x_{t-1} + 25x_{t-1}/(1 + x_{t-1}^2) + 8\cos(1.2(t-1)) + u_t, \\ \text{observation equation} \quad &: \quad y_t = x_t^2/20 + v_t, \end{aligned}$$

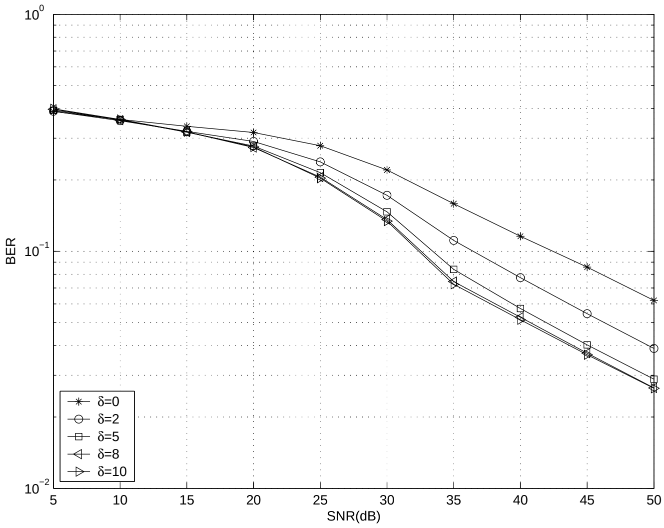

Figure 5: BER performance of the lookahead weighting method with $\Delta = 0$, $\Delta' = 0$, $m = 200$ and different δ in 256-QAM system.

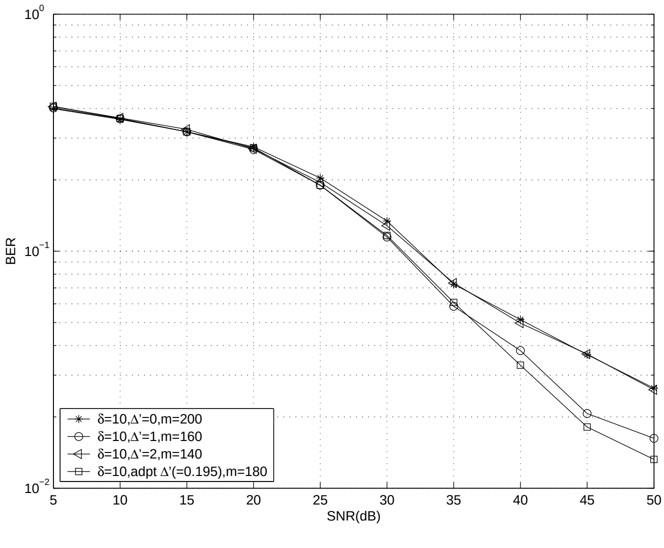

Figure 6: BER performance of the multilevel pilot lookahead sampling method with $\Delta = 0$, $\delta = 10$ but different Δ' and number of samples m in 256-QAM system. The number of samples are chosen so that each of the method takes approximately the same CPU time.

where $u_t \sim N(0, \sigma^2)$, $v_t \sim N(0, \eta^2)$ are Gaussian white noise. In the simulation, we let $\sigma = 1$ and $\eta = 1$, and the length of observations is $T = 100$. We compare the performance of different lookahead strategies.

In this nonlinear system, $\pi_t(x_t|\boldsymbol{x}_{t-1})$ can not be easily sampled from. Here we use the simple trial distribution

$$q_t(x_t|\boldsymbol{x}_{t-1}) = \pi_{t-1}(x_t|\boldsymbol{x}_{t-1}) = g_t(x_t \mid x_{t-1}) \quad \text{and} \quad q_s^{pilot}(x_s|\boldsymbol{x}_{s-1}) = g_s(x_s \mid x_{s-1}).$$

We use SMC to denote the pilot lookahead sampling method for continuous state space case. The implementation with smoothing step presented in Section 4.4 is denoted as SMC-S. A simple piecewise constant function with interval width 0.5 is used for smoothing. Resampling is applied at every step.

We repeat the experiment 1,000 times. The goodness of the fit measures used are

$$RMSE_1 = \left[\frac{1}{T}\sum_{t=1}^{T}(\widehat{x}_t - x_t)^2\right]^{1/2} \quad \text{and} \quad RMSE_2 = \left[\frac{1}{T}\sum_{t=1}^{T}\left(\widehat{x}_t - \widetilde{E}_{\pi_{t+\delta+\Delta'}}(x_t)\right)^2\right]^{1/2},$$

where $RMSE_2$ is a measurement of estimation variance, $I(\delta + \Delta')$ in (2). Here $\widetilde{E}_{\pi_{\delta+\Delta'}}(x_t)$ is obtained by SMC ($\Delta' = 0$) with a large number of samples ($m = 200,000$) and the lookahead weighting method with lookahead steps $\delta^* = \delta + \Delta'$. Tables 1 and 2 report average $RMSE_1$ and $RMSE_2$ and the associated CPU time of using different sampling methods and $m = 3,000$ samples. It can be seen that the delayed methods can greatly reduce $RMSE_1$ for small $\delta + \Delta'$, but no further improvement can be found when $\delta + \Delta' \geq 3$. SMC with $\Delta' = 1, A = 10, K = 16$ is an approximation of the exact lookahead sampling method with $\Delta = 1$. It has the smallest $RMSE_1$ at the cost of extensive computation, which confirms Proposition 3. The performance of SMC with single pilot ($K = 1$) is poor because the future state space can not be efficiently explored by small number of pilots. With the smoothing step, SMC-S can achieve better performance than the simple lookahead weighting method (SMC, $\Delta' = 0$). SMC-S with $A = 3$ has better performance than SMC-S with $A = 1$, because when using $A = 1$, the pilot only affects resampling and estimation, but not the sampling procedure. However, SMC-S with $A = 3$ also takes longer CPU time.

We also use the adaptive stop criteria (24) (adpt) to choose the lookahead steps adaptively. In the criteria, we let $\sigma_0^2 = 4$. The adaptive method has similar performance to the fixed-step pilot lookahead sampling method, but much fewer average lookahead steps (average lookahead steps are only 0.244) and less CPU time.

For a fair comparison, Tables 3 and 4 report average $RMSE_1$ and $RMSE_2$ of different methods with different numbers of samples, which are chosen so that each method used approximately the same CPU time. In this table, SMC with $A = 1$ and adaptive lookahead scheme has the smallest

$RMSE_1 \quad \backslash \quad \Delta' + \delta$	0	1	2	3	5	7	time(sec.)
SMC($\Delta' = 0$)	3.128	1.011	0.828	0.817	0.818	0.819	0.113
SMC($\Delta' = 1, A = 10, K = 16$)	–	1.009	0.824	0.813	0.812	0.813	5.952
SMC($\Delta' = 1, A = 3$)	–	1.011	0.831	0.826	0.831	0.839	0.319
SMC($\Delta' = 2, A = 3$)	–	–	0.838	0.844	0.860	0.876	0.405
SMC($\Delta' = 3, A = 3$)	–	–	–	0.846	0.885	0.913	0.504
SMC-S($\Delta' = 1, A = 1$)	–	1.009	0.825	0.815	0.814	0.815	0.170
SMC-S($\Delta' = 2, A = 1$)	–	–	0.825	0.815	0.815	0.815	0.197
SMC-S($\Delta' = 3, A = 1$)	–	–	–	0.816	0.816	0.816	0.224
SMC-S(adpt $\Delta'(0.244), A = 1$)	0.995	0.834	0.815	0.814	0.816	0.817	0.147
SMC-S($\Delta' = 1, A = 3$)	–	1.009	0.824	0.813	0.813	0.813	0.421
SMC-S($\Delta' = 2, A = 3$)	–	–	0.824	0.813	0.813	0.813	0.498
SMC-S($\Delta' = 3, A = 3$)	–	–	–	0.814	0.814	0.813	0.576

Table 1: Average $RMSE_1$ for SMC with different lookahead methods. The same numbers of samples ($m = 3,000$) are used in different methods. We use single pilot lookahead ($K = 1$) unless stated otherwise. Average lookahead steps in the adaptive lookahead method are reports in the parentheses.

$RMSE_1$, which demonstrates the effectiveness of the adaptive lookahead strategy. It also shows that SMC-1 with $\Delta' = 1, A = 10, K = 16$ has a large $RMSE_1$, because of its high computational cost per sample.

6.3 Target Tracking in Clutter

Consider the problem of tracking a single target in clutter (Avitzour, 1995). In this example, the target moves with random acceleration in one dimension. The state equation can be written as

$$\begin{pmatrix} x_{t,1} \\ x_{t,2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1,1} \\ x_{t-1,2} \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} u_t,$$

where $x_{t,1}$ and $x_{t,2}$ denote the one dimensional location and velocity of the target, respectively; $u_t \sim N(0, \sigma^2)$ is the random acceleration.

At each time t, the target can be observed with probability p_d independently. If the target is observed, the observation is

$$z_t = x_{t,1} + v_t,$$

$RMSE_2 \quad \backslash \quad \Delta' + \delta$	0	1	2	3	5	7	time(sec.)
SMC($\Delta' = 0$)	0.137	0.055	0.057	0.066	0.078	0.090	0.113
SMC($\Delta' = 1, A = 10, K = 16$)	–	0.023	0.027	0.032	0.038	0.043	5.952
SMC($\Delta' = 1, A = 3$)	–	0.070	0.105	0.138	0.174	0.203	0.319
SMC($\Delta' = 2, A = 3$)	–	–	0.156	0.220	0.278	0.326	0.405
SMC($\Delta' = 3, A = 3$)	–	–	–	0.240	0.356	0.417	0.504
SMC-S($\Delta' = 1, A = 1$)	–	0.043	0.048	0.053	0.062	0.072	0.170
SMC-S($\Delta' = 2, A = 1$)	–	–	0.051	0.063	0.066	0.075	0.197
SMC-S($\Delta' = 3, A = 1$)	–	–	–	0.073	0.081	0.090	0.224
SMC-S($\Delta' = 1, A = 3$)	–	0.029	0.032	0.036	0.041	0.048	0.421
SMC-S($\Delta' = 2, A = 3$)	–	–	0.031	0.039	0.042	0.047	0.498
SMC-S($\Delta' = 3, A = 3$)	–	–	–	0.045	0.050	0.055	0.576

Table 2: Average $RMSE_2$ for SMC with different lookahead methods. The same numbers of samples (m=3,000) are used in different methods.

where $v_t \sim N(0, r^2)$.

In additional to the true observation, there are false signals. Observation of false signals follows a spatially homogeneous Poisson process with rate λ. Suppose the observation window is wide and centers around the predicted location of the target. Let Δ be the range of the observation window. The actual observation y_t includes n_t detected signals, among which at most one is the true observation. Therefore, n_t follows a Bernoulli(p_d)+Poisson($\lambda\Delta$) distribution.

Define an indicator variable I_t as follows

$$I_t = \begin{cases} 0, & \text{if the target is not detected at time } t, \\ k, & \text{if the } k\text{-th signal in } y_t \text{ is the true observation,} \end{cases}$$

then we have

$$p(y_t, I_t \mid x_t) \propto \begin{cases} (1 - p_d)\lambda, & \text{if } I_t = 0, \\ p_d(2\pi r^2)^{-1/2} exp\{-(y_{t,k} - x_t)^2/2r^2\}, & \text{if } I_t = k > 0. \end{cases}$$

In this system, given $\boldsymbol{I}_t = (I_1, \cdots, I_t)$, it becomes a linear Gaussian state space model. In such a system, the mixture Kalman filter (MKF) can be applied. The mixture Kalman filter only generates samples of the indicators $\boldsymbol{I}_t^{(j)}$ and considers the state space as discrete. Conditional on $\boldsymbol{I}_t^{(j)}$ and $\boldsymbol{y}_t$, the state variable x_t is normally distributed. The mean and the variance of $p(x_{t-\delta} \mid$

$RMSE_1 \quad \backslash \quad \Delta' + \delta$	0	1	2	3	5	7	time(sec.)
SMC ($m = 3,000, \Delta' = 0$)	3.128	1.011	0.828	0.817	0.818	0.819	0.113
SMC ($m = 60, \Delta' = 1, A = 10, K = 16$)	–	1.079	0.911	0.906	0.912	0.920	0.125
SMC-S($m = 2,000, \Delta' = 1, A = 1$)	–	1.010	0.826	0.817	0.817	0.818	0.117
SMC-S($m = 1,700, \Delta' = 2, A = 1$)	–	–	0.827	0.818	0.817	0.819	0.116
SMC-S($m = 1,500, \Delta' = 3, A = 1$)	–	–	–	0.820	0.822	0.823	0.118
SMC-S($m = 2,400$, adpt $\Delta'(0.245), A = 1$)	0.994	0.835	0.816	**0.815**	0.817	0.818	0.104
SMC-S($m = 800, \Delta' = 1, A = 3$)	–	1.015	0.832	0.821	0.821	0.822	0.108
SMC-S($m = 700, \Delta' = 2, A = 3$)	–	–	0.827	0.817	0.816	0.817	0.111
SMC-S($m = 600, \Delta' = 3, A = 3$)	–	–	–	0.819	0.819	0.820	0.119

Table 3: Average $RMSE_1$ for SMC with different lookahead methods. The numbers of samples are chosen so that each method used approximately the same CPU time. Average lookahead steps in the adaptive lookahead method are reports in the parentheses.

$\boldsymbol{I}_t^{(j)}, \boldsymbol{y}_t)$ can be exactly calculated through the Kalman filter. To perform lookahead strategies in MKF, suppose we can obtain samples $\{(\boldsymbol{I}_{t+\Delta}^{(j)}, w_t^j), j = 1, \cdots, m\}$ properly weighted with respect to $\pi_{t+\Delta}(\boldsymbol{I}_{t+\Delta}) = p(\boldsymbol{I}_{t+\Delta} \mid \boldsymbol{y}_{t+\Delta})$, then

$$\frac{\sum_{j=1}^m w_t^{(j)} E_{\pi_{t+\Delta}}(x_{t-\delta} \mid \boldsymbol{I}_{t+\Delta} = \boldsymbol{I}_{t+\Delta}^{(j)})}{\sum_{j=1}^m w_t^{(j)}}$$

is a consistent estimator of $E_{\pi_{t+\Delta}}(x_{t-\delta})$, $\delta = 0, 1, \cdots$. More details of MKF and MKF with lookahead can be found in Chen and Liu (2000) and Wang et al. (2002).

In this example, we can also use the smoothing step presented in Section 4.4 to improve the performance of the pilot lookahead sampling method. It can be shown that $\overline{V}_{t+1:t+\Delta}^{(j,i)} = E(V_{t+1:t+\Delta}^{(j,i)} \mid \boldsymbol{I}_{t-1}^{(j)}, I_t = i, \boldsymbol{y}_{t+\Delta})$ in (16) only depends on the mean $\mu_t^{(j,i)}$ and the variance $\Sigma_t^{(j,i)}$ of the normal distribution $p(x_t \mid \boldsymbol{I}_{t-1}^{(j)}, I_t = i, \boldsymbol{y}_t)$. For simplicity, we approximately assume $\overline{V}_{t+1:t+\Delta}^{(j,i)}$ only depends on $\mu_t^{(j,i)} = (\mu_{t,1}^{(j,i)}, \mu_{t,2}^{(j,i)})$, that is,

$$V_{t+1:t+\Delta}^{(j,i)} \approx \overline{V}_{t+1:t+\Delta}(\mu_t^{(j,i)}) + e_t^{(j,i)}.$$

We then use the smoothed $\overline{V}_{t+1:t+\Delta}^{(j,i)}$ to reduce the variation introduced by random pilots. We denoted this method by MKF-S. We used the piecewise constant smoother to estimate $\overline{U}_t(\mu_t^{(j,i)})$. In the smoother, the space $\left[\min\{\mu_{t,1}^{(j,i)}\}, \max\{\mu_{t,1}^{(j,i)}\}\right] \times \left[\min\{\mu_{t,2}^{(j,i)}\}, \max\{\mu_{t,2}^{(j,i)}\}\right]$ is divided into 10×10 equal parts.

$RMSE_2 \quad \backslash \quad \Delta' + \delta$	0	1	2	3	5	7	time(sec.)
SMC($m = 3000, \Delta' = 0$)	0.137	0.055	0.057	0.066	0.078	0.090	0.113
SMC($m = 60, \Delta' = 1, A = 10, K = 16$)	–	0.228	0.254	0.277	0.306	0.334	0.125
SMC-S($m = 2,000, \Delta' = 1, A = 1$)	–	0.054	0.058	0.064	0.075	0.087	0.117
SMC-S($m = 1,700, \Delta' = 2, A = 1$)	–	–	0.066	0.083	0.085	0.098	0.116
SMC-S($m = 1,500, \Delta' = 3, A = 1$)	–	–	–	0.103	0.114	0.126	0.118
SMC-S($m = 800, \Delta' = 1, A = 3$)	–	0.062	0.067	0.074	0.084	0.096	0.108
SMC-S($m = 700, \Delta' = 2, A = 3$)	–	–	0.061	0.078	0.082	0.094	0.111
SMC-S($m = 600, \Delta' = 3, A = 3$)	–	–	–	0.097	0.109	0.121	0.119

Table 4: Average $RMSE_2$ for SMC with different lookahead methods. The numbers of samples are chosen so that each method used approximately the same CPU time.

In this example, we let $\sigma^2 = 0.1$, $r^2 = 1.0$, $p_d = 0.8$, $\lambda = 0.1$, and $\Delta = 100\,r$. The length of the observation period is $T = 100$. We repeat the experiment 500 times. The resmapling step is applied when the effective sample size is less than $0.1m$.

Following Avitzour (1995), we use the median absolute error (MAE) as the performance measurement. Define

$$MAE_1 = \text{median}\left\{|\widehat{x}_{t,1} - x_{t,1}|\right\} \quad \text{and} \quad MAE_2 = \text{median}\left\{|\widehat{x}_{t,1} - \widetilde{E}_{\pi_{t+\Delta+\delta}}(x_{t,1})|\right\},$$

where $\widehat{x}_{t,1}$ is the consistent estimation of $E_{\pi_{t+\Delta+\delta}}(x_{t,1})$ using different lookahead methods, and $\widetilde{E}_{\pi_{t+\Delta+\delta}}(x_{t,1})$ is obtained by the lookahead weighting method using large number of samples (m=20,000).

We first compare the performance of different lookahead methods using the same number of samples (m=200). Table 5 reports MAE_1 and MAE_2 for the lookahead weighting method (MKF, $\Delta = 0$), the exact lookahead sampling method (MKF, $\Delta = 3$), and the single pilot lookahead sampling method (MKF, $\Delta' = 3$ and MKF-S, $\Delta' = 3$). From the result, MAE_1 decreases as the number of lookahead steps increases, which shows the effectiveness of the lookahead strategies. The exact lookahead sampling method (MKF, $\Delta = 3$) has the smallest MAE_2, which confirms Propositions 3 and 5, although its computational cost is the highest. We can also see that MKF-S ($\Delta' = 3$) performs better than MKF ($\Delta' = 3$).

Then we compare the performance of different methods under similar computational cost. The number of samples is adjusted so that each method takes approximately the same CPU time. Table 6 reports the quantiles of absolute estimation errors $|\widehat{x}_{t,1} - x_{t,1}|$ for different lookahead methods

$\Delta+\delta$ / $\Delta'+\delta$		0	1	2	3	5	8	10	13	15
MAE_1	MKF($\Delta=0$)	1.0300	0.7890	0.6560	0.5830	0.5180	0.4750	0.4590	0.4470	0.4450
	MKF($\Delta=3$)	–	–	–	0.5780	0.5150	0.4710	0.4540	0.4410	0.4370
	MKF($\Delta'=3$)	–	–	–	0.5780	0.5150	0.4730	0.4560	0.4440	0.4420
	MKF-S($\Delta'=3$)	–	–	–	0.5730	0.5120	0.4690	0.4530	0.4410	0.4370
MAE_2	MKF($\Delta=0$)	0.0932	0.0760	0.0618	0.0525	0.0460	0.0453	0.0467	0.0500	0.0520
	MKF($\Delta=3$)	–	–	–	0.0463	0.0357	0.0307	0.0298	0.0298	0.0305
	MKF($\Delta'=3$)	–	–	–	0.0575	0.0503	0.0472	0.0480	0.0503	0.0525
	MKF-S($\Delta'=3$)	–	–	–	0.0490	0.0398	0.0360	0.0353	0.0365	0.0375

Table 5: MAE_1 and MAE_2 for different lookahead methods. The same numbers of samples (m=200) are used. The CPU time used in each experiment is 0.341 seconds for the lookahead weighting method (MKF, $\Delta = 0$); 3.554 seconds for the exact lookahead sampling method (MKF, $\Delta = 3$); 0.783 seconds for the pilot lookahead sampling method (MKF, $\Delta' = 3$), and 0.788 seconds for MKF-S ($\Delta' = 3$).

with lookahead steps $\Delta + \delta = 15$ (or $\Delta' + \delta = 15$). The performance does not improve further when $\Delta' + \delta \geq 15$. Under the same CPU time, the lookahead sampling method has the largest absolute estimation error because of its high computational cost. The pilot lookahead sampling method (MKF-S, $\Delta' = 3$) has better performance than the simple lookahead weighting method (MKF, $\Delta = 0$). We then use the stop criteria (24) to choose lookahead steps in the pilot lookahead sampling method adaptively (MKF-S, adpt Δ'). When we set $\sigma_0^2 = 1.5r^2$ in criteria (24), the average number of lookahead steps is 1.572. The result shows that the adaptive pilot lookahead sampling method performs the best under the same CPU time.

Appendix

Proof of Proposition 1: For any $\Delta_2 > \Delta_1 \geq 0$, we have

$$\begin{aligned}
& E_{\pi_t}\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_1}\right) - h(\boldsymbol{x}_t)\Big]^2 \\
=& E_{\pi_t}\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_1}\right) - E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right)\Big]^2 + E_{\pi_t}\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right) - h(\boldsymbol{x}_t)\Big]^2 \\
& + 2E_{\pi_t}\left\{\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_1}\right) - E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right)\Big]\Big[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right) - h(\boldsymbol{x}_t)\Big]\right\}.
\end{aligned}$$

Quantiles ($\Delta + \delta = 15$)	0.05	0.25	0.50	0.75	0.95	time (sec.)
MKF($m = 450, \Delta = 0$)	0.0400	0.2040	0.4420	0.7910	1.7885	0.791
MKF($m = 50, \Delta = 3$)	0.0400	0.2080	0.4490	0.8120	2.1610	0.851
MKF-S($m = 200, \Delta' = 3$)	0.0390	0.2030	0.4370	0.7790	1.6590	0.788
MKF-S($m = 280$, adpt Δ')	0.0390	0.2020	0.4340	0.7700	1.6295	0.802

Table 6: Quantiles of absolute estimation errors $|\widehat{x}_{t,1} - x_{t,1}|$ for different lookahead methods. The numbers of samples are chosen so that each method used approximately the same CPU time. In the adaptive pilot lookahead sampling method (MKF-S, adpt Δ'), the average number of lookahead steps is 1.572.

Because

$$E_{\pi_t}\left\{\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_1}\right) - E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right)\right]\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right) - h(\boldsymbol{x}_t)\right]\right\}$$

$$= E\left\{E\left\{\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_1}\right) - E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right)\right]\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta_2}\right) - h(\boldsymbol{x}_t)\right] \,\Big|\, \boldsymbol{y}_{t+\Delta_2}\right\}\Big|\boldsymbol{y}_t\right\} = 0,$$

the conclusion holds.

Proof of Proposition 2: It is easily seen that

$$\begin{aligned}
E_{\pi_{t+\Delta}}\left[w_t^{(2,j)} h(\boldsymbol{x}_t^{(2,j)}) \mid \boldsymbol{x}_{t-1}^{(j)}\right] &= E_{\pi_{t+\Delta}}\left[w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_t^{(2,j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(2,j)} \mid \boldsymbol{x}_{t-1}^{(j)})} h(\boldsymbol{x}_t^{(2,j)}) \mid \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})} E_{\pi_{t+\Delta}}\left[h(\boldsymbol{x}_t^{(2,j)}) \frac{\pi_{t+\Delta}(x_t^{(2,j)} \mid \boldsymbol{x}_{t-1}^{(j)})}{q_t(x_t^{(2,j)} \mid \boldsymbol{x}_{t-1}^{(j)})} \,\Big|\, \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(j)}\right).
\end{aligned}$$

Then (10) is a direct result of Rao-Blackwellization. By replacing $h(\boldsymbol{x}_t)$ with $E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1}\right)$ and $h(\boldsymbol{x}_t) = 1$ in (10), we obtain (11) and (9) respectively.

Proof of Proposition 3: Since

$$\begin{aligned}
& E_{\pi_{t+\Delta}}\left\{w_t^{(3,j)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}) \mid \boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}\right\} \\
= \; & E_{\pi_{t+\Delta}}\left\{h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}) \left[w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t+\Delta}^{(3,j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) \prod_{s=t}^{t+\Delta} q_s(x_s^{(3,j)} \mid \boldsymbol{x}_{s-1}^{(3,j)})}\right] \Big| \boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}\right\} \\
= \; & h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}) \, w_{t-1}^{(j)} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)}) q_t(x_t^{(2,j)} \mid \boldsymbol{x}_{t-1}^{(j)})} E_{\pi_{t+\Delta}}\left\{\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t+1:t+\Delta}^{(3,j)} \mid \boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)})}{\prod_{s=t+1}^{t+\Delta} q_s(x_s^{(3,j)} \mid \boldsymbol{x}_{s-1}^{(3,j)})} \,\Big|\, \boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}\right\} \\
= \; & w_t^{(2,j)} h(\boldsymbol{x}_{t-1}^{(j)}, x_t^{(2,j)}),
\end{aligned}$$

we have

$$var_{\pi_{t+\Delta}}\left[w_{t+\Delta}^{(3,j)}h(\boldsymbol{x}_{t-1}^{(j)},x_t^{(2,j)})\right] \geq var_{\pi_{t+\Delta}}\left[w_t^{(2,j)}h(\boldsymbol{x}_{t-1}^{(j)},x_t^{(2,j)})\right]$$

according to Rao-Blackwellization theorem.

Proof of Proposition 4: To prove $(\boldsymbol{x}_t^{(j)}, w_t^{aux(j)})$ is properly weighted with respect to distribution $\pi_{t+\Delta}(\boldsymbol{x}_t)$, we only need to prove

$$E_{\pi_{t+\Delta}}\left[w_t^{aux(j)}h(\boldsymbol{x}_t^{(j)})\right] = E_{\pi_{t+\Delta}}\left[h(\boldsymbol{x}_t)\right].$$

According to the sampling distribution of the lookahead pilot $\boldsymbol{x}_{t+\Delta}^{(j,i)}$ and calculation of the corresponding cumulative incremental weight $U_t^{(j,i)}$,

$$\begin{aligned} E_{\pi_{t+\Delta}}\left[w_t^{aux(j)}h(\boldsymbol{x}_t^{(j)}) \mid \boldsymbol{x}_{t-1}^{(j)}\right] &= E_{\pi_{t+\Delta}}\left[w_{t-1}^{(j)}\sum_{i=1}^{\mathcal{A}} U_t^{(j,i)}h(\boldsymbol{x}_{t-1}^{(j)}, x_t=a_i) \mid \boldsymbol{x}_{t-1}^{(j)}\right] \\ &= w_{t-1}^{(j)}\sum_{i=1}^{\mathcal{A}} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t=a_i)h(\boldsymbol{x}_{t-1}^{(j)}, x_t=a_i)}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})} \\ &= w_{t-1}^{(j)}\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}E_{\pi_{t+\Delta}}\left[h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1}^{(j)}\right]. \end{aligned}$$

Because sample $(\boldsymbol{x}_{t-1}^{(j)}, w_{t-1}^{(j)})$ is properly weighted with respect to $\pi_{t-1}(\boldsymbol{x}_{t-1})$,

$$\begin{aligned} E_{\pi_{t+\Delta}}\left[w_{t-1}^{(j)}\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}\ E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1}^{(j)}\right)\right] &= E_{\pi_{t-1}}\left[\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1})}{\pi_{t-1}(\boldsymbol{x}_{t-1})}\ E_{\pi_{t+\Delta}}\left(h(\boldsymbol{x}_t) \mid \boldsymbol{x}_{t-1}\right)\right] \\ &= E_{\pi_{t+\Delta}}\left[h(\boldsymbol{x}_t)\right], \end{aligned}$$

the proposition follows.

Proof of Proposition 5: Because $w_t^{aux(j)}$ and $w_t^{(1,j)}$ both are importance weight, we have $E_{\pi_{t+\Delta}}\left(w_t^{aux(j)}\right) = E_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right) = 1$. Hence

$$\begin{aligned} var_{\pi_{t+\Delta}}\left(w_t^{aux(j)}\right) - var_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right) &= E_{\pi_{t+\Delta}}\left(w_t^{aux(j)}\right)^2 - E_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right)^2 \\ &= E_{\pi_{t+\Delta}}\left\{E_{\pi_{t+\Delta}}\left[\left(w_t^{aux(j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] - E_{\pi_{t+\Delta}}\left[\left(w_t^{(1,j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right]\right\}. \end{aligned}$$

Now we consider the difference between $E_{\pi_{t+\Delta}}\left[\left(w_t^{aux(j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right]$ and $E_{\pi_{t+\Delta}}\left[\left(w_t^{(1,j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right]$. Let

$$\varepsilon^{(j,i,k)} = U_t^{(j,i,k)} - \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t=a_i)}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}.$$

Because $E_{\pi_{t+\Delta}}\left[U_t^{(j,i,k)} \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] = \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t=a_i)}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}$, we have $E_{\pi_{t+\Delta}}\left[\varepsilon^{(j,i,k)} \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] = 0$. In addition, $\varepsilon^{(j,i,k)}$, $i = 1,\cdots,\mathcal{A}$, $k = 1,\cdots,K$ are independent conditional on $\boldsymbol{x}_{t-1}^{(j)}$, and for fixed i, $\varepsilon^{(j,i,k)}$,

$k = 1, \cdots, K$, follow the same distribution.

$$
\begin{aligned}
&E_{\pi_{t+\Delta}}\left[\left(w_t^{aux(j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= \left(w_{t-1}^{(j)}\right)^2 E_{\pi_{t+\Delta}}\left[\left(\sum_{i=1}^{\mathcal{A}} \frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i)}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})} + \sum_{i=1}^{\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} \varepsilon^{(j,i,k)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= \left(w_{t-1}^{(j)}\right)^2 E_{\pi_{t+\Delta}}\left[\left(\frac{\pi_{t+\Delta}(\boldsymbol{x}_{t-1}^{(j)})}{\pi_{t-1}(\boldsymbol{x}_{t-1}^{(j)})}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] + \left(w_{t-1}^{(j)}\right)^2 E_{\pi_{t+\Delta}}\left[\left(\sum_{i=1}^{\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} \varepsilon^{(j,i,k)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= E_{\pi_{t+\Delta}}\left[\left(w_t^{(1,j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] + \frac{1}{K^2}\left(w_{t-1}^{(j)}\right)^2 \sum_{i=1}^{\mathcal{A}} \sum_{k=1}^{K} E_{\pi_{t+\Delta}}\left[\left(\varepsilon^{(j,i,k)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] \\
&= E_{\pi_{t+\Delta}}\left[\left(w_t^{(1,j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] + \frac{1}{K}\left(w_{t-1}^{(j)}\right)^2 \sum_{i=1}^{\mathcal{A}} E_{\pi_{t+\Delta}}\left[\left(\varepsilon^{(i,j,k=1)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right].
\end{aligned}
$$

Then we have

$$
E_{\pi_{t+\Delta}}\left[\left(w_t^{aux(j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] - E_{\pi_{t+\Delta}}\left[\left(w_t^{(1,j)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right] = \frac{1}{K}\left(w_{t-1}^{(j)}\right)^2 \sum_{i=1}^{\mathcal{A}} E_{\pi_{t+\Delta}}\left[\left(\varepsilon^{(i,j,k=1)}\right)^2 \,\middle|\, \boldsymbol{x}_{t-1}^{(j)}\right],
$$

hence

$$
0 \le var_{\pi_{t+\Delta}}\left(w_t^{aux(j)}\right) - var_{\pi_{t+\Delta}}\left(w_t^{(1,j)}\right) \sim O(1/K).
$$

With similar method, we can prove

$$
\begin{aligned}
0 \le var_{\pi_{t+\Delta}}&\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} U_t^{(j,i,k)} h\left(\boldsymbol{x}_{t-1}^{(j)}, x_t = a_i\right)\right] \\
&- var_{\pi_{t+\Delta}}\left[w_t^{(1,j)} E_{\pi_{t+\Delta}}\left(h(x_{t-1}^{(j)}, x_t) \mid x_{t-1}^{(j)}\right)\right] \sim O(1/K).
\end{aligned}
$$

Proof of Proposition 6: Let $\psi_t(\boldsymbol{x}_t) \propto r_t(\boldsymbol{x}_t) b_t(\boldsymbol{x}_t)$ be the distribution of samples after resampling, Because $E_{\pi_{t+T}}\left(w_{t+T}^{*(j)}\right) \equiv 1$, we only consider minimizing $E_{\pi_{t+T}}\left(w_{t+T}^{*(j)}\right)^2$. We have

$$
\begin{aligned}
&E_{\pi_{t+T}}\left(w_{t+T}^{*(j)}\right)^2 \\
&= \int \frac{\pi_t^2(\boldsymbol{x}_t)}{\psi_t^2(\boldsymbol{x}_t)} \left[\frac{\pi_{t+T}(\boldsymbol{x}_{t+T})}{\pi_t(\boldsymbol{x}_t) \prod_{s=t+1}^{t+T} q_s(x_s \mid \boldsymbol{x}_{s-1})}\right]^2 \psi_t(\boldsymbol{x}_t) \prod_{s=t+1}^{t+T} q_s(x_s \mid \boldsymbol{x}_{s-1})\, d\boldsymbol{x}_t dx_{t+1} \dots dx_{t+T} \\
&= \int \frac{\pi_t^2(\boldsymbol{x}_t)}{\psi_t(\boldsymbol{x}_t)} \eta(\boldsymbol{x}_t) d\boldsymbol{x}_t.
\end{aligned}
$$

According to Jensen's inequality, to minimize $E_{\pi_{t+T}}\left(w_{t+T}^{*(j)}\right)^2$, $\psi_t(\boldsymbol{x}_t)$ needs to be proportional to $\pi_t(\boldsymbol{x}_t)\eta^{1/2}(\boldsymbol{x}_t)$.

Proof of Proposition 7: For estimator $\widehat{h}^*$, because $\widehat{h}^*$ and $h(\boldsymbol{x}_t)$ are independent conditional on $\boldsymbol{y}_{t+\Delta+1}$, we have

$$\begin{aligned}
&E\left[\left(\widehat{h}^* - h(\boldsymbol{x}_t)\right)^2 \mid \boldsymbol{y}_{t+\Delta}\right] \\
=&E\left\{E\left[\left[\widehat{h}^* - E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}) + E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}) - h(\boldsymbol{x}_t)\right]^2 \mid \boldsymbol{y}_{t+\Delta+1}\right] \Big| \boldsymbol{y}_{t+\Delta}\right\} \\
=&E\left[var\left(\widehat{h}^* \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] + E\left[\left[E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}) - h(\boldsymbol{x}_t)\right]^2 \mid \boldsymbol{y}_{t+\Delta}\right],
\end{aligned}$$

and for estimator $\widehat{h}$,

$$E\left[\left(\widehat{h} - h(\boldsymbol{x}_t)\right)^2 \mid \boldsymbol{y}_{t+\Delta}\right] = var\left[\widehat{h} \mid \boldsymbol{y}_{t+\Delta}\right] + E\left[\left[E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}) - h(\boldsymbol{x}_t)\right]^2 \mid \boldsymbol{y}_{t+\Delta}\right].$$

Similar to the proof of Proposition 1, we have

$$\begin{aligned}
&E\left[\left[E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta}) - h(\boldsymbol{x}_t)\right]^2 \mid \boldsymbol{y}_{t+\Delta}\right] - E\left[\left[E(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}) - h(\boldsymbol{x}_t)\right]^2 \mid \boldsymbol{y}_{t+\Delta}\right] \\
=& var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right].
\end{aligned}$$

Hence,

$$\begin{aligned}
&E\left[\left(\widehat{h}^* - h(\boldsymbol{x}_t)\right)^2 \mid \boldsymbol{y}_{t+\Delta}\right] - E\left[\left(\widehat{h} - h(\boldsymbol{x}_t)\right)^2 \mid \boldsymbol{y}_{t+\Delta}\right] \\
=&E\left[var\left(\widehat{h}^* \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] - var\left[\widehat{h} \mid \boldsymbol{y}_{t+\Delta}\right] - var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] \\
=&\frac{1}{m} E\left[var\left(w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] \\
&- \frac{1}{m}\, var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta}\right] - var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] \\
=&\frac{1}{m}\left\{var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta}\right] - var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right]\right. \\
&\left. - var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta}\right]\right\} - var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right] \\
=&\frac{1}{m}\left\{var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta}\right] - var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \Big| \boldsymbol{y}_{t+\Delta}\right]\right\} \\
&- \left(1 + \frac{1}{m}\right) var\left[E\left(h(\boldsymbol{x}_t) \mid \boldsymbol{y}_{t+\Delta+1}\right) \Big| \boldsymbol{y}_{t+\Delta}\right].
\end{aligned} \tag{25}$$

In the pilot lookahead sampling method, because

$$U_{t,\Delta+1}^{(j,i)} = U_{t,\Delta}^{(j,i)} \frac{p(\boldsymbol{x}_{t+\Delta+1}^{(j,i)} \mid \boldsymbol{y}_{t+\Delta+1})}{p(\boldsymbol{x}_{t+\Delta}^{(j,i)} \mid \boldsymbol{y}_{t+\Delta}) q_{t+\Delta+1}(x_{t+\Delta+1}^{(j,i)} \mid \boldsymbol{x}_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta+1})},$$

we have

$$
\begin{aligned}
&E\left(U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_t^{(j,i)}) \mid \boldsymbol{x}_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta}\right) \\
=&U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_t^{(j,i)}) \int \frac{p(\boldsymbol{x}_{t+\Delta+1}^{(j,i)} \mid \boldsymbol{y}_{t+\Delta+1})}{p(\boldsymbol{x}_{t+\Delta}^{(j,i)} \mid \boldsymbol{y}_{t+\Delta}) q_{t+\Delta+1}(x_{t+\Delta+1}^{(j,i)} \mid x_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta+1})} \\
&\qquad \times q_{t+\Delta+1}(x_{t+\Delta+1}^{(j,i)} \mid x_{t+\Delta}^{(j,i)}, \boldsymbol{y}_{t+\Delta+1})\, p(y_{t+\Delta+1} \mid \boldsymbol{y}_{t+\Delta})\; dx_{t+\Delta+1}^{(j,i)} dy_{t+\Delta+1} \\
=&U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_t^{(j,i)}),
\end{aligned}
$$

according to Rao-Blackwellization theorem,

$$
\begin{aligned}
&var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \,\Big|\, \boldsymbol{y}_{t+\Delta}\right] - var\left[w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \,\Big|\, \boldsymbol{y}_{t+\Delta}\right] \\
=&E\left[var\left(w_{t-1}^{(j)} \sum_{i=1}^{\mathcal{A}} U_{t,\Delta+1}^{(j,i)} h(\boldsymbol{x}_{t-1}^{(j,i)}) \mid \boldsymbol{x}_{t+\Delta}^{(j,i=1:\mathcal{A})}, \boldsymbol{y}_{t+\Delta}\right) \,\Big|\, \boldsymbol{y}_{t+\Delta}\right].
\end{aligned} \tag{26}
$$

Combine (25) and (26), the conclusion holds.

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010), "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society, Series B*, 72, 1–33.

Avitzour, D. (1995), "Stochastic simulation Bayesian approach to multitarget tracking," *IEE Proceedings on Radar, Sonar and Navigation*, 142, 41–44.

Briers, M., Doucet, A., and Maskell, S. (2010), "Smoothing algorithms for state-space models," *Annals of the Institute Statistical Mathematics*, 62, 61–89.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999), "An improved particle for non-linear problems," *IEE Proceedings on Radar, Sonar, and Navigation,*, 146, 2–7.

Carter, C. and Kohn, R. (1994), "On Gibbs sampling for state space models,," *Biometrika*, 81, 541–553.

Carvalho, C., Johannes, M., Lopes, H., and Polson, N. (2010), "Particle learning and smoothing," *Statistical Science*, 25, 88–106.

Chen, R. and Liu, J. S. (2000), "Mixture Kalman filters," *Journal of the Royal Statistical Society, Series B*, 62, 493–508.

Chen, R., Wang, X., and Liu, J. (2000), "Adaptive Joint Detection and Decoding in flat-fading channels via mixture Kalman filtering," *IEEE Transaction on Information Theory*, 46, 2079–2094.

Chopin, N. (2002), "A sequential particle filter method for static models," *Biometrika*, 89, 539–552.

— (2004), "Central limit theorem for sequential Monte Carlo methods and its application to bayesian inference," *Annals of Statistics*, 32, 2385–2411.

Clapp, T. and Godsill, S. (1997), "Bayesian blind deconvolution for mobile communications," *Proceedings of IEE Colloqium on Adaptive signal processing for mobile communication systems*, 9, 9/1–9/6.

— (1999), "Fixed-Lag smoothing using sequential importance sampling," in *Bayesian Statistics 6*, eds. Bernardo, J., Berger, J., Dawid, A., and Smith, A., Oxford: Oxford University Press, pp. 743–752.

Crisan, D. and Lyons, T. (2002), "Minimal entropy approximations and optimal algorithms," *Monte Carlo Methods and Applications*, 8, 343–355.

Del Moral, P. (2004), *Feynman-Kac Formulae Genealogical and Interacting Particle Systems with Applications*, New York: Springer-Verlag.

Douc, R., Garivier, E., Moulines, E., and Olsson, J. (2009), "On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state space models," *Working paper, Institut Télécom.*

Doucet, A., Briers, M., and Sénécal, S. (2006), "Efficient block sampling strategies for sequential Monte Carlo methods," *Journal of Computations and Graphical Statistics*, 15, 1–19.

Doucet, A., de Freitas, J. F. G., and Gordon, N. J. (2001), *Sequential Monte Carlo in Practice*, New York: Springer-Verlag.

Fearnhead, P. (2002), "Markov chain Monte Carlo, sufficient statistics, and particle filters," *Journal of Computational and Graphical Statistics*, 11, 848–862.

Fearnhead, P. and Clifford, P. (2003), "On-line inference for hidden Markov models via particle filters," *Journal of the Royal Statistical Society, Series B*, 65, 887–899.

Fearnhead, P., Wyncoll, D., and Tawn, J. (2010), "A sequential smoothing algorithm with linear computational cost," *Biometrika*, 97, 447–464.

Fong, W., Godsill, S., Doucet, A., and West, M. (2002), "Monte Carlo smoothing with application to speekch enhancement," *IEEE Transaction on Signal Processing*, 50, 438–449.

Gilks, W. R. and Berzuini, C. (2001), "Following a moving targetMonte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society, Series B*, 63, 127–146.

Godsill, S., Doucet, A., and West, M. (2004), "Monte Carlo smoothing for non-linear time series," *Journal of the American Statistical Association*, 50, 438–449.

Godsill, S. J. and Vermaak, J. (2004), "Models and algorithms for tracking using trans-dimensional sequential Monte Carlo," *Proc. IEEE ICASSP*, 3, 976–979.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993), "Novel approach to nonlinear / non-Gaussian Bayesian state estimation," *IEE Proceedings on Radar and Signal Processing*, 140, 107–113.

Guo, D., Wang, X., and Chen, R. (2004), "Multilevel Mixture Kalman filter," *EURASIP Journal on Applied Signal Processing, Special issue on Particle Filtering*, 15, 2255–2266.

Hürzeler, M. and Künsch, H. (1995), "Monte Carlo approximations for general state space models," Research Report 73, ETH, Zürich.

Ikoma, N., Ichimura, N., Higuchi, T., and Maeda, H. (2001), "Maneuvering target tracking by using particle filter," *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 4, 2223–2228.

Kantas, N., Doucet, A., Singh, S., and Maciejowski, J. (2009), "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models," *15th IFAC Symposium on System Identification.*

Kim, S., Shephard, N., and Chib, S. (1998), "Stochastic volatility: likelihood inference and comparison with ARCH models," *The Review of Economic Studies*, 65, 361–393.

Kitagawa, G. (1996), "Monte Carlo filter and smoother for non-Gaussian nonlinear State space models,," *Journal of Computational and Graphical Statistics*, 5, 1–25.

Kong, A., Liu, J., and Wong, W. (1994), "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, 89, 278–288.

Kotecha, J. and Djuric, P. (2003), "Gaussian Sum Particle Filters," *IEEE Transactions on Signal Processing*, 51, 2602–2612.

Liang, J., Chen, R., and Zhang, J. (2002), "Statistical geometry of packing defects of lattice chain polymer from enumeration and sequential Monte Carlo method," *J. Chemical Physics*, 117, 3511–3521.

Lin, M., Zhang, J., Cheng, Q., and Chen, R. (2005), "Independent particle filters," *Journal of the American Statistical Association*, 100, 1412–1421.

Liu, J. and Chen, R. (1995), "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, 90, 567–576.

— (1998), "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer, New York.

Liu, J. S., Chen, R., and Wong, W. H. (1998), "Rejection control for sequential importance sampling," *Journal of the American Statistical Association*, 93, 1022–1031.

Marshall, A. (1956), "The use of multi-stage sampling schemes in Monte Carlo computations," in *Symposium on Monte Carlo Methods*, ed. Meyer, M., Wiley, pp. 123–140.

Pitt, M. K. (2002), "Smooth particle filters for likelihood and Maximisation," Tech. rep., University of Warwick.

Pitt, M. K. and Shephard, N. (1999), "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, 94, 590–599.

Rosenbluth, M. N. and Rosenbluth, A. W. (1955), "Monte Carlo calculation of the average extension of molecular chains," *Journal of Chemical Physics*, 23, 356–359.

van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2002), "The unscented particle filter," in *Advances in Neural Information Processing Systems (NIPS13)*, ed. Leen, T. K., Dietterich, T. G. and Tresp, V., MIT Press.

Wang, X., Chen, R., and Guo, D. (2002), "Delayed Pilot Sampling for Mixture Kalman Filter with Application in Fading Channels," *IEEE Transaction on Signal Processing*, 50, 241–264.

Zhang, J. L. and Liu, J. S. (2002), "A new sequential importance sampling method and its application to the two-dimensional hydrophobicChydrophilic model," *Journal of chemical physics*, 117, 3492–3498.