Global Sampling for Sequential Filtering over Discrete State Space

Pascal Cheung-Mon-Chan

École Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris Cédex 13, France
Email: pcheung@tsi.enst.fr

Eric Moulines

École Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris Cédex 13, France
Email: moulines@tsi.enst.fr

In many situations, there is a need to approximate a sequence of probability measures over a growing product of finite spaces. Whereas it is in general possible to determine analytic expressions for these probability measures, the number of computations needed to evaluate these quantities grows exponentially thus precluding real-time implementation. Sequential Monte Carlo techniques (SMC), which consist in approximating the flow of probability measures by the empirical distribution of a finite set of *particles*, are attractive techniques for addressing this type of problems. In this paper, we present a simple implementation of the sequential importance sampling/resampling (SISR) technique for approximating these distributions; this method relies on the fact that, the space being finite, it is possible to consider every offspring of the trajectory of particles. The procedure is straightforward to implement, and well-suited for practical implementation. A limited Monte Carlo experiment is carried out to support our findings.

Keywords and phrases: particle filters, sequential importance sampling, sequential Monte Carlo sampling, sequential filtering, conditionally linear Gaussian state-space models, autoregressive models.

1. INTRODUCTION

State-space models have been around for quite a long time to model dynamic systems. State-space models are used in a variety of fields such as computer vision, financial data analysis, mobile communication, radar systems, among others. A main challenge is to design efficient methods for online estimation, prediction, and smoothing of the hidden state given the continuous flow of observations from the system. Except in a few special cases, including linear state-space models (see [1]) and hidden finite-state Markov chain (see [2]), this problem does not admit computationally tractable exact solutions.

From the mid 1960s, considerable research efforts have been devoted to develop computationally efficient methods to approximate these distributions; in the last decade, a great deal of attention has been devoted to sequential Monte Carlo (SMC) algorithms (see [3] and the references therein). The basic idea of SMC method consists in approximating the conditional distribution of the hidden state with the empirical distribution of a set of random points, called *particles*. These particles can either give birth to offspring particles or die,

depending on their ability to represent the distribution of the hidden state conditional on the observations. The main difference between the different implementations of the SMC algorithms depends on the way this population of particles evolves in time. It is no surprise that most of the efforts in this field has been dedicated to finding numerically efficient and robust methods, which can be used in real-time implementations.

In this paper, we consider a special case of state-space model, often referred to in the literature as *conditionally Gaussian linear state-space models* (CGLSSMs), which has received a lot of attention in the recent years (see, e.g., [4, 5, 6, 7]). The main feature of a CGLSSM is that, conditionally on a set of indicator variables, here taking their values in a finite set, the system becomes linear and Gaussian. Efficient recursive procedures—such as the Kalman filter/smoother—are available to compute the distribution of the state variable conditional on the indicator variable and the observations. By embedding these algorithms in the sequential importance sampling/resampling (SISR) framework, it is possible to derive computationally efficient sampling procedures which focus their attention on the space of indicator variables.

These algorithms are collectively referred to as *mixture Kalman filters (MKFs)*, a term first coined by Chen and Liu [8] who have developed a generic sampling algorithm; closely related ideas have appeared earlier in the automatic control/signal processing and computational statistics literature (see, e.g., [9, 10] for early work in this field; see [5] and the references therein for a tutorial on these methods; see [3] for practical implementations of these techniques). Because these sampling procedures operate on a lower-dimensional space, they typically achieve lower Monte Carlo variance than "plain" particle filtering methods.

In the CGLSSM considered here, it is assumed that the indicator variables are discrete and take a finite number of different values. It is thus feasible to consider every possible offspring of a trajectory, defined here as a particular realization of a sequence of indicator variables from initial time 0 to the current time t. This has been observed by the authors in [5, 7, 8], among many others, who have used this property to design appropriate proposal distributions for improving the accuracy and performance of SISR procedures.

In this work, we use this key property in a different way, along the lines drawn in [11, Section 3]; the basic idea consists in considering the population of every possible offspring of every trajectory and *globally* sampling from this population. This algorithm is referred to as the *global sampling (GS)* algorithm. This algorithm can be seen as a simple implementation of the SISR algorithm for the so-called *optimal* importance distribution.

Some limited Monte Carlo experiments on prototypal examples show that this algorithm compares favorably with state-of-the-art implementation of MKF; in a joint symbol estimation and channel equalization task, we have in particular achieved extremely encouraging performance with as few as 5 particles, making the proposed algorithm amenable to real-time applications.

2. SEQUENTIAL MONTE CARLO ALGORITHMS

2.1. Notations and definitions

Before going further, some additional definitions and notations are required. Let $\mathbf{X}$ (resp., $\mathbf{Y}$) be a general set and let $\mathcal{B}(\mathbf{X})$ (resp., $\mathcal{B}(\mathbf{Y})$) denote a σ-algebra on $\mathbf{X}$ (resp., $\mathbf{Y}$). If Q is a nonnegative function on $\mathbf{X} \times \mathcal{B}(\mathbf{Y})$ such that

(i) for each $B \in \mathcal{B}(\mathbf{Y})$, $Q(\cdot, B)$ is a nonnegative measurable function on $\mathbf{X}$,
(ii) for each $x \in \mathbf{X}$, $Q(x, \cdot)$ is a measure on $\mathcal{B}(\mathbf{Y})$,

then we call Q a transition kernel from $(\mathbf{X}, \mathcal{B}(\mathbf{X}))$ to $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$ and we denote $Q : (\mathbf{X}, \mathcal{B}(\mathbf{X})) \prec (\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$. If for each $x \in \mathbf{X}$, $Q(x, \cdot)$ is a finite measure on $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$, then we say that the transition is finite. If for all $x \in \mathbf{X}$, $Q(x, \cdot)$ is a probability measure on $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$, then Q is said to be a *Markov transition kernel*.

Denote by $\mathcal{B}(\mathbf{X}) \otimes \mathcal{B}(\mathbf{Y})$ the product σ-algebra (the smallest σ-algebra containing all the sets $A \times B$, where $A \in \mathcal{B}(\mathbf{X})$ and $B \in \mathcal{B}(\mathbf{Y})$). If μ is a measure on $(\mathbf{X}, \mathcal{B}(\mathbf{X}))$ and Q is a transition kernel, $Q : (\mathbf{X}, \mathcal{B}(\mathbf{X})) \prec (\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$, we denote

by $\mu \otimes Q$ the measure on the product space $(\mathbf{X} \times \mathbf{Y}, \mathcal{B}(\mathbf{X}) \otimes \mathcal{B}(\mathbf{Y}))$ defined by

$$\mu \otimes Q(A \times B) = \int_A \mu(dx) Q(x, B) \quad \forall A \in \mathcal{B}(\mathbf{X}), B \in \mathcal{B}(\mathbf{Y}). \quad (1)$$

Let $X : (\Omega, \mathcal{F}) \to (\mathbf{X}, \mathcal{B}(\mathbf{X}))$ and $Y : (\Omega, \mathcal{F}) \to (\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$ be two random variables and μ and ν two measures on $(\mathbf{X}, \mathcal{B}(\mathbf{X}))$ and $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}))$, respectively. Assume that the probability distribution of (X, Y) has a density denoted by $f(x, y)$ with respect to $\mu \otimes \nu$. We denote by $f(y|x) = f(x, y) / \int_{\mathbf{Y}} f(x, y) \nu(dy)$ the conditional density of Y given X.

2.2. Sequential importance sampling

Let $\{F_t\}_{t \geq 0}$ be a sequence of probability measures on $(\mathbf{Z}^{t+1}, \mathcal{P}(\mathbf{Z})^{\otimes(t+1)})$, where $\mathbf{Z} \stackrel{\text{def}}{=} \{z_1, \dots, z_M\}$ is a finite set with cardinal equal to M. It is assumed in this section that for any $\lambda_{0:t-1} \in \mathbf{Z}^t$ such that $f_{t-1}(\lambda_{0:t-1}) = 0$, we have

$$f_t([\lambda_{0:t-1}, \lambda]) = 0 \quad \forall \lambda \in \mathbf{Z}, \quad (2)$$

where for any $\tau \geq 0$, f_τ denotes the density of F_τ with respect to the counting measure. For any $t \geq 1$, there exists a finite transition kernel $Q_t : (\mathbf{Z}^t, \mathcal{P}(\mathbf{Z})^{\otimes t}) \prec (\mathbf{Z}, \mathcal{P}(\mathbf{Z}))$ such that

$$F_t = F_{t-1} \otimes Q_t. \quad (3)$$

We denote by q_t the density of the kernel Q_t with respect to the counting measure, which can simply be expressed as

$$q_t(\lambda_{0:t-1}, \lambda) = \begin{cases} \dfrac{f_t([\lambda_{0:t-1}, \lambda])}{f_{t-1}(\lambda_{0:t-1})} & \text{if } f_{t-1}(\lambda_{0:t-1}) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In the SIS framework (see [5, 8]), the probability distribution F_t on $\mathbf{Z}^{t+1}$ is approximated by particles $(\Lambda^{(1,t)}, \dots, \Lambda^{(N,t)})$ associated to nonnegative weights $(w^{(1,t)}, \dots, w^{(N,t)})$; the estimator of the probability measure associated to this weighted particle system is given by

$$F_t^N = \frac{\sum_{i=1}^N w^{(i,t)} \delta_{\Lambda^{(i,t)}}}{\sum_{i=1}^N w^{(i,t)}}. \quad (5)$$

These trajectories and weights are obtained by drawing N independent trajectories $\Lambda^{(i,t)}$ under an instrumental probability distribution G_t on $(\mathbf{Z}^{t+1}, \mathcal{P}(\mathbf{Z})^{\otimes(t+1)})$ and computing the importance weights as

$$w^{(i,t)} = \frac{f_t(\Lambda^{(i,t)})}{g_t(\Lambda^{(i,t)})}, \quad i \in \{1, \dots, N\}, \quad (6)$$

where g_t is the density of the probability measure G_t with respect to the counting measure on $(\mathbf{Z}^{t+1}, \mathcal{P}(\mathbf{Z})^{(t+1)})$. It is assumed that for each t, F_t is absolutely continuous with respect to the instrumental probability G_t, that is, for all $\lambda_{0:t} \in \mathbf{Z}^{t+1}$ such that $g_t(\lambda_{0:t}) = 0$, $f_t(\lambda_{0:t}) = 0$. In the SIS

framework, these weighted trajectories are updated by drawing at each time step an offspring of each particle and then computing the associated importance weight. It is assumed in the sequel that the instrumental probability measure satisfies a decomposition similar to (3), that is,

$$G_t = G_{t-1} \otimes K_t, \qquad (7)$$

where $K_t : (\mathbf{Z}^t, \mathcal{P}(\mathbf{Z})^{\otimes t}) \prec (\mathbf{Z}, \mathcal{P}(\mathbf{Z}))$ is a Markov transition kernel: $\sum_{j=1}^M K_t(\lambda_{0:t-1}, \{z_j\}) = 1$. Hence, for all $\lambda_{0:t-1} \in \mathbf{Z}^t$, $\sum_{j=1}^M g_t([\lambda_{0:t-1}, z_j]) = g_{t-1}(\lambda_{0:t-1})$, showing that whenever $g_{t-1}(\lambda_{0:t-1}) = 0$, $g_t([\lambda_{0:t-1}, z_j]) = 0$ for all $j \in \{1, \ldots, M\}$. Define by k_t the density of the Markov transition kernel K_t with respect to the counting measure:

$$k_t(\lambda_{0:t-1}, \lambda) = \begin{cases} \dfrac{g_t([\lambda_{0:t-1}, \lambda])}{g_{t-1}(\lambda_{0:t-1})} & \text{if } g_{t-1}(\lambda_{0:t-1}) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (8)$$

In the SIS framework, at each time t, for each particle $\Lambda^{(i,t-1)}$, $i \in \{1, \ldots, N\}$, and then for each particular offspring $j \in \{1, \ldots, M\}$, we evaluate the weights

$$\rho^{(i,j,t)} = k_t(\Lambda^{(i,t-1)}, z_j) \qquad (9)$$

and we draw an index $J^{(i,t)}$ from a multinomial distribution with parameters $(\rho^{(i,1,t-1)}, \ldots, \rho^{(i,M,t-1)})$ conditionally independently from the past:

$$\mathbb{P}[J^{(i,t)} = j \mid \mathcal{G}_{t-1}] = \rho^{(i,j,t)}, \quad i \in \{1, \ldots, N\}, \ j \in \{1, \ldots, M\}, \qquad (10)$$

where $\mathcal{G}_t$ is the history of the particle system at time t,

$$\mathcal{G}_t = \sigma((\Lambda^{(j,\tau)}, w^{(j,\tau)}), \ 1 \leq j \leq N, \ 1 \leq \tau \leq t). \qquad (11)$$

The updated system of particles then is

$$\Lambda^{(i,t)} = [\Lambda^{(i,t-1)}, z_{J^{(i,t)}}]. \qquad (12)$$

If $(\Lambda^{(1,0)}, \ldots, \Lambda^{(N,0)})$ is an independent sample from the distribution G_0, it is then easy to see that at each time t, the particles $(\Lambda^{(1,t)}, \ldots, \Lambda^{(N,t)})$ are independent and distributed according to G_t; the associated (unnormalized) importance weights $w^{(i,t)} = f_t(\Lambda^{(i,t)})/g_t(\Lambda^{(i,t)})$ can be written as a product $w^{(i,t)} = u_t(\Lambda^{(i,t-1)}, z_{J^{(i,t)}}) w^{(i,t-1)}$, where the incremental weight $u_t(\Lambda^{(i,t-1)}, Z_{J^{(i,t)}})$ is given by

$$u_t([\lambda_{0:t-1}, \lambda]) \overset{\text{def}}{=} \frac{q_t(\lambda_{0:t-1}, \lambda)}{k_t(\lambda_{0:t-1}, \lambda)} \quad \forall \lambda_{0:t-1} \in \mathbf{Z}^t, \ \lambda \in \mathbf{Z}. \quad (13)$$

It is easily shown that the instrumental distribution k_t which minimizes the variance of the importance weights conditionally to the history of the particle system (see [5, Proposition 2]) is given by

$$k_t(\lambda_{0:t-1}, \cdot) = \frac{q_t(\lambda_{0:t-1}, \cdot)}{\sum_{j=1}^M q_t(\lambda_{0:t-1}, z_j)} \quad \text{for any } \lambda_{0:t-1} \in \mathbf{Z}^t. \qquad (14)$$

The choice of the optimal instrumental distribution (14) has been introduced in [12] and has since then been used and/or rediscovered by many authors (see [5, Section II-D] for a discussion and extended references). Using this particular form of the importance kernel, the incremental importance sampling weights (13) are given by

$$u_t(\Lambda^{(i,t-1)}, z_{J^{(i,t)}}) = \sum_{j=1}^M q_t(\Lambda^{(i,t-1)}, z_j), \quad i \in \{1, \ldots, N\}. \qquad (15)$$

It is worthwhile to note that $u_t([\Lambda^{(i,t-1)}, z_j]) = u_t([\Lambda^{(i,t-1)}, z_l])$ for all $j, l \in \{1, \ldots, M\}$; the incremental importance weights do not depend upon the particular offspring of the particle which is drawn.

2.3. Sequential importance sampling/resampling

The normalized importance weights $\bar{w}^{(i,t)} \overset{\text{def}}{=} w^{(i,t)} / \sum_{i=1}^N w^{(i,t)}$ reflect the contribution of the imputed trajectories to the importance sampling estimate F_t^N. A weight close to zero indicates that the associated trajectory has a "small" contribution. Such trajectories are thus ineffective and should be eliminated.

Resampling is the method usually employed to combat the degeneracy of the system of particles. Let $[\Lambda^{(1,t-1)}, \ldots, \Lambda^{(N,t-1)}]$ be a set of particles at time $t - 1$ and let $[w^{(1,t-1)}, \ldots, w^{(N,t-1)}]$ be the associated importance weights. An SISR iteration, in its most elementary form, produces a set of particles $[\Lambda^{(1,t)}, \ldots, \Lambda^{(N,t)}]$ with equal weights $1/N$. The SISR algorithm is a two-step procedure. In the first step, each particle is updated according to the importance transition kernel k_t and the incremental importance weights are computed according to (12) and (13), exactly as in the SIS algorithm. This produces an intermediate set of particles $\tilde{\Lambda}^{(i,t)}$ with associated importance weights $\tilde{w}^{(i,t)}$ defined as

$$\tilde{\Lambda}^{(i,t)} = [\Lambda^{(i,t-1)}, z_{\tilde{J}^{(i,t)}}],$$
$$\tilde{w}^{(i,t)} = w^{(i,t-1)} u_t(\Lambda^{(i,t-1)}, z_{\tilde{J}^{(i,t)}}), \quad i \in \{1, \ldots, N\}, \qquad (16)$$

where the random variables $\tilde{J}^{(i,t)}$, $i \in \{1, \ldots, N\}$, are drawn conditionally independently from the past according to a multinomial distribution with parameters

$$\mathbb{P}[\tilde{J}^{(i,t)} = j \mid \mathcal{G}_{t-1}] = k_t(\Lambda^{(i,t-1)}, z_j),$$
$$i \in \{1, \ldots, N\}, \ j \in \{1, \ldots, M\}. \qquad (17)$$

We denote by $\bar{S}_t = ((\tilde{\Lambda}^{(i,t)}, \tilde{w}^{(i,t)}), \ i \in \{1, \ldots, N\})$, this intermediate set of particles. In the second step, we resample the intermediate particle system. Resampling consists in transforming the weighted approximation of the probability measure F_t, $F_t^N = \sum_{i=1}^N \tilde{w}^{(i,t)} \delta_{\tilde{\Lambda}^{(i,t)}}$, into an unweighted one, $\tilde{F}_t^N = N^{-1} \sum_{i=1}^N \delta_{\Lambda^{(i,t)}}$. To avoid introducing bias during the resampling step, an unbiased resampling procedure should be used. More precisely, we draw with replacements N indices $I^{(1,t)}, \ldots, I^{(N,t)}$ in such a way that $N^{(i,t)} = \sum_{k=1}^N \delta_{i,I^{(k,t)}}$,

the number of times the ith trajectory is chosen satisfies

$$\sum_{i=1}^{N} N^{(i,t)} = N, \quad \mathbb{E}[N^{(i,t)} \mid \tilde{\mathcal{G}}_t] = N\tilde{w}^{(i,t)}$$
$$\text{for any } i \in \{1,\dots,N\}, \tag{18}$$

where $\tilde{\mathcal{G}}_t$ is the history of the particle system *just before* the resampling step (see (11)), that is, $\tilde{\mathcal{G}}_t$ is the σ-algebra generated by the union of $\mathcal{G}_{t-1}$ and $\sigma(\tilde{J}^{(1,t)},\dots,\tilde{J}^{(N,t)})$:

$$\tilde{\mathcal{G}}_t = \mathcal{G}_{t-1} \vee \sigma(\tilde{J}^{(1,t)},\dots,\tilde{J}^{(N,t)}). \tag{19}$$

Then, we set, for $k \in \{1,\dots,N\}$,

$$(I^{(k,t)}, J^{(k,t)}) = (I^{(k,t)}, \tilde{J}^{(I^{(k,t)},t)}),$$
$$\Lambda^{(k,t)} = [\Lambda^{(I^{(k,t)},t-1)}, z_{J^{(k,t)}}], \quad w^{(k,t)} = \frac{1}{N}. \tag{20}$$

Note that the sampling is done with replacement in the sense that the same particle can be either eliminated or copied several times in the final updated sample. We denote by $S_t = ((\Lambda^{(i,t)}, w^{(i,t)}), i \in \{1,\dots,N\})$ this set of particles.

There are several options to obtain an unbiased sample. The most obvious choice consists in drawing the N particles conditionally independently on $\tilde{G}_t$ according to a multinomial distribution with normalized weights $(\tilde{w}^{(1,t)},\dots,\tilde{w}^{(N,t)})$. In the literature, this is referred to as multinomial sampling. As a result, under multinomial sampling, the particles $\Lambda^{(i,t)}$ are conditional on $\tilde{G}_t$ independent and identically distributed (i.i.d.). There are however better algorithms which reduce the added variability introduced during the sampling step (see the appendix).

This procedure is referred to as the SISR procedure. The particles with large normalized importance weights are likely to be selected and will be kept alive. On the contrary, the particles with low normalized importance weights are eliminated. Resampling provides more efficient samples of *future* states but increases sampling variation in the *past* states because it reduces the number of distinct trajectories.

The SISR algorithm with multinomial sampling defines a Markov chain on the path space. The transition kernel of this chain depends upon the choice of the proposal distribution and of the unbiased procedure used in the resampling step. These transition kernels are, except in a few special cases, involved. However, when the "optimal" importance distribution (14) is used in conjunction with multinomial sampling, the transition kernel has a simple and intuitive expression. As already mentioned above, the incremental weights for all the possible offsprings of a given particle are, in this case, identical; as a consequence, under multinomial sampling, the indices $I^{(k,t)}$, $k \in \{1,\dots,N\}$, are i.i.d. with multinomial distribution for all $k \in \{1,\dots,N\}$,

$$\mathbb{P}[I^{(k,t)} = i \mid \tilde{\mathcal{G}}_t]$$
$$= \frac{(\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)) w^{(i,t-1)}}{\sum_{i=1}^{N} (\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)) w^{(i,t-1)}}, \quad i \in \{1,\dots,N\}. \tag{21}$$

Recall that, when the optimal importance distribution is used, for each particle $i \in \{1,\dots,N\}$, the random variables $\tilde{J}^{(i,t)}$, $i \in \{1,\dots,M\}$, are conditionally independent from $\mathcal{G}_{t-1}$ and are distributed with multinomial random variable with parameters

$$\mathbb{P}[\tilde{J}^{(i,t)} = j \mid \mathcal{G}_{t-1}] = \frac{q_t(\Lambda^{(i,t-1)}, z_j)}{\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)}, \tag{22}$$
$$i \in \{1,\dots,N\}, \; j \in \{1,\dots,M\}.$$

We may compute, for $i,k \in \{1,\dots,N\}$ and $j \in \{1,\dots,M\}$,

$$\mathbb{P}[(I^{(k,t)}, J^{(k,t)}) = (i,j) \mid \mathcal{G}_{t-1}]$$
$$= \mathbb{E}[\mathbb{P}[I^{(k,t)} = i, \tilde{J}^{(i,t)} = j \mid \tilde{\mathcal{G}}_t] \mid \mathcal{G}_{t-1}]$$
$$= \mathbb{E}[\mathbb{P}[I^{(k,t)} = i \mid \tilde{\mathcal{G}}_t]\mathbf{1}(\tilde{J}^{(i,t)} = j) \mid \mathcal{G}_{t-1}]$$
$$= \frac{(\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)) w^{(i,t-1)}}{\sum_{i=1}^{N} (\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)) w^{(i,t-1)}} \tag{23}$$
$$\times \mathbb{P}[\tilde{J}^{(i,t)} = j \mid \mathcal{G}_{t-1}]$$
$$= \frac{q_t(\Lambda^{(i,t-1)}, z_j) w^{(i,t-1)}}{\sum_{i=1}^{N} (\sum_{j=1}^{M} q_t(\Lambda^{(i,t-1)}, z_j)) w^{(i,t-1)}} = \bar{w}^{(i,j,t)},$$

showing that the SISR algorithm is equivalent to drawing, conditionally independently from $\mathcal{G}_{t-1}$, N random variables out of $N \times M$ possible offsprings of the system of particles, with weights $(\bar{w}^{(i,j,t)}, i \in \{1,\dots,N\}, j \in \{1,\dots,N\})$.

Resampling can be done at any time. When resampling is done at every time step, it is said to be systematic. In this case, the importance weights at each time t, $w^{(i,t)}$, $i \in \{1,\dots,N\}$, are all equal to $1/N$. Systematic resampling is not always recommended since resampling is costly from the computational point of view and may result in loss of statistical efficiency by introducing some additional randomness in the particle system. However, the effect of resampling is not necessarily negative because it allows to control the degeneracy of the particle systems, which has a positive impact on the quality of the estimates. Therefore, systematic resampling yields in some situations better estimates than the standard SIS procedure (without resampling); in some cases (see Section 4.2 for an illustration), it compares favorably with more sophisticated versions of the SISR algorithm, where resampling is done at random times (e.g., when the entropy or the coefficient of variations of the normalized importance weights is below a threshold).

2.4. The global sampling algorithm

When the instrumental distribution is the so-called optimal sampling distribution (14), it is possible to combine the sampling/resampling step above into a single sampling step. This idea has already been mentioned and worked out in [11, Section 3] under the name of deterministic/resample low weights (RLW) approach, yet the algorithm given below is not given explicitly in this reference.

Let $[\Lambda^{(1,t-1)},\dots,\Lambda^{(N,t-1)}]$ be a set of particles at time $t-1$ and let $[w^{(1,t-1)},\dots,w^{(N,t-1)}]$ be the associated importance weights. Similar to the SISR step, the GS algorithm produces

a set of particles $[\Lambda^{(1,t)}, \ldots, \Lambda^{(N,t)}]$ with equal weights. The GS algorithm combines the two-stage sampling procedure (first, samples a particular offspring of a particle, updates the importance weights, and then resamples from the population) into a single one.

(i) We first compute the weights

$$w^{(i,j,t)} = w^{(i,t-1)} q_t\big(\Lambda^{(i,t-1)}, z_j\big), \quad i \in \{1, \ldots, N\}, \ j \in \{1, \ldots, M\}. \tag{24}$$

(ii) We then draw N random variables $((I^{(1,t)}, J^{(1,t)}), \ldots, (I^{(N,t)}, J^{(N,t)}))$ in $\{1, \ldots, N\} \times \{1, \ldots, M\}$ using an unbiased sampling procedure, that is, for all $(i, j) \in \{1, \ldots, N\} \times \{1, \ldots, M\}$, the number of times of the particles (i, j) is

$$N^{(i,j,t)} \overset{\text{def}}{=} \big| \{k \in \{1, \ldots, N\}, \ (I^{(k,t)}, J^{(k,t)}) = (i,j)\} \big| \tag{25}$$

thus satisfying the following two conditions:

$$\sum_{i'=1}^{N} \sum_{j'=1}^{M} N^{(i',j',t)} = N,$$

$$\mathbb{E}\big[N^{(i,j,t)} \,\big|\, \mathcal{G}_{t-1}\big] = N \frac{w^{(i,j,t)}}{\sum_{i'=1}^{N} \sum_{j'=1}^{M} w^{(i',j',t)}}. \tag{26}$$

The updated set of particles is then defined as

$$\Lambda^{(k,t)} = \big[\Lambda^{(I^{(k,t)},t-1)}, z_{J^{(k,t)}}\big], \quad w^{(k,t)} = \frac{1}{N}. \tag{27}$$

If multinomial sampling is used, then the GS algorithm is a simple implementation of the SISR algorithm, which combines the two-stage sampling into a single one. Since the computational cost of drawing L random variables grows linearly with L, the cost of simulations is proportional to NM for the GS algorithm and $NM + N$ for the SISR algorithm. There is thus a (slight) advantage in using the GS implementation. When sampling is done using a different unbiased method (see the appendix), then there is a more substantial difference between these two algorithms. As illustrated in the examples below, the GS may outperform the SISR algorithm.

3. GLOBAL SAMPLING FOR CONDITIONALLY GAUSSIAN STATE-SPACE MODELS

3.1. Conditionally linear Gaussian state-space model

As emphasized in the introduction, CGLSSMs are a particular class of state-space models which are such that, conditional to a set of indicator variables, the system becomes linear and Gaussian. More precisely,

$$\begin{aligned} S_t &= \Psi_t(\Lambda_{0:t}), \\ \vec{X}_t &= A_{S_t} \vec{X}_{t-1} + C_{S_t} \vec{W}_t, \\ \vec{Y}_t &= B_{S_t} \vec{X}_t + D_{S_t} \vec{V}_t, \end{aligned} \tag{28}$$

where

(i) $\{\Lambda_t\}_{t \geq 0}$ are the *indicators variables*, here assumed to take values in a finite set $\mathbf{Z} = \{z_1, z_2, \ldots, z_M\}$, where M denotes the cardinal of the set $\mathbf{Z}$; the law of $\{\Lambda_t\}_{t \geq 0}$ is assumed to be known but is otherwise not specified;

(ii) for any $t \geq 0$, Ψ_t is a function $\Psi_t : \mathbf{Z}^{t+1} \to \mathbf{S}$, where $\mathbf{S}$ is a finite set;

(iii) $\{\vec{X}_t\}_{t \geq 0}$ are the $(n_x \times 1)$ *state vectors*; these state variables are not directly observed;

(iv) the distribution of $\vec{X}_0$ is complex Gaussian with mean $\vec{\mu}_0$ and covariance Γ_0;

(v) $\{\vec{Y}_t\}_{t \geq 0}$ are the $(n_y \times 1)$ *observations*;

(vi) $\{\vec{W}_t\}_{t \geq 0}$ and $\{\vec{V}_t\}_{t \geq 0}$ are (complex) n_w- and n_v-dimensional (complex) Gaussian white noise, $\vec{W}_t \sim \mathcal{N}_c(0, I_{n_w \times n_w})$ and $\vec{V}_t \sim \mathcal{N}_c(0, I_{n_v \times n_v})$, where $I_{p \times p}$ is the $p \times p$ identity matrix; $\{\vec{W}_t\}_{t \geq 0}$ is referred to as the *state noise*, whereas $\{\vec{V}_t\}_{t \geq 0}$ is the *observation noise*;

(vii) $\{A_s, s \in \mathbf{S}\}$ are the state transition matrices, $\{B_s, s \in \mathbf{S}\}$ are the observation matrices, and $\{C_s, s \in \mathbf{S}\}$ and $\{D_s, s \in \mathbf{S}\}$ are Cholesky factors of the covariance matrix of the state noise and measurement noise, respectively; these matrices are assumed to be known;

(viii) the indicator process $\{\Lambda_t\}_{t \geq 0}$ and the noise observation processes $\{\vec{V}_t\}_{t \geq 0}$ and $\{\vec{W}_t\}_{t \geq 0}$ are independent.

This model has been considered by many authors, following the pioneering work in [13, 14] (see [5, 7, 8, 15] for authoritative recent surveys). Despite its simplicity, this model is flexible enough to describe many situations of interests including linear state-space models with non-Gaussian state noise or observation noise (heavy-tail noise), jump linear systems, linear state space with missing observations; of course, digital communication over fading channels, and so forth.

Our aim in this paper is to compute recursively in time an estimate of the conditional probability of the (unobserved) indicator variable Λ_n given the observation up to time $n + \Delta$, that is, $\mathbb{P}(\Lambda_n \mid \vec{Y}_{0:n+\Delta} = \vec{y}_{0:n+\Delta})$, where Δ is a nonnegative integer and for any sequence $\{\lambda_t\}_{t \geq 0}$ and any integer $0 \leq i < j$, we denote $\lambda_{i:j} \overset{\text{def}}{=} \{\lambda_i, \ldots, \lambda_j\}$. When $\Delta = 0$, this distribution is called the *filtering distribution*; when $\Delta > 0$, it is called the fixed-lag smoothing distribution, and Δ is the lag.

3.2. Filtering

In this section, we describe the implementation of the GS algorithm to approximate the filtering probability of the indicator variables given the observations

$$f_t(\lambda_{0:t}) = \mathbb{P}[\Lambda_{0:t} = \lambda_{0:t} \mid Y_{0:t} = y_{0:t}] \tag{29}$$

in the CGLSSM (28). We will first show that the filtering probability F_t satisfies condition (3), that is, for any $t \geq 1$, $F_t = F_{t-1} \otimes Q_t$; we then present an efficient recursive algorithm to compute the transition kernel Q_t using the Kalman filter update equations. For any $t \geq 1$ and for any $\lambda_{0:t} \in \mathbf{Z}^{t+1}$,

under the conditional independence structure implied by the CGLSSM (28), the Bayes formula shows that

$$q_t(\lambda_{0:t-1}; \lambda_t) \propto f(\vec{y}_t | \vec{y}_{0:t-1}, \lambda_{0:t}) f(\lambda_t | \lambda_{0:t-1}). \quad (30)$$

The predictive distribution of the observations given the indicator variables $f(\vec{y}_t | \vec{y}_{0:t-1}, \lambda_{0:t})$ can be evaluated along each trajectory of indicator variables $\lambda_{0:t}$ using the Kalman filter recursions. Denote by $g_c(\cdot; \mu, \Gamma)$ the density of a complex circular Gaussian random vector with mean $\vec{\mu}$ and covariance matrix Γ, and for A a matrix, let $A^{\dagger}$ be the transpose conjugate of A; we have, with $s_t = \Psi_t(\lambda_{0:t})$ (and Ψ_t is defined in (28)),

$$
\begin{aligned}
& f(\vec{y}_t | \lambda_{0:t}, \vec{y}_{0:t-1}) \\
& = g_c(\vec{y}_t; B_{s_t} \mu_{t|t-1}[\lambda_{0:t}], B_{s_t} \Gamma_{t|t-1}[\lambda_{0:t}] B_{s_t}^{\dagger} + D_{s_t} D_{s_t}^{\dagger}),
\end{aligned}
\quad (31)
$$

where $\vec{\mu}_{t|t-1}[\lambda_{0:t}]$ and $\Gamma_{t|t-1}[\lambda_{0:t}]$ denote the filtered mean and covariance of the state, that is, the conditional mean and covariance of the state given the indicators variables $\lambda_{0:t}$ and the observations up to time $t-1$ (the dependence of the predictive mean $\vec{\mu}_{t|t-1}[\lambda_{0:t}]$ on the observations $y_{0:t-1}$ is implicit). These quantities can be computed recursively using the following Kalman one-step prediction/correction formula. Denote by $\vec{\mu}_{t-1}([\lambda_{0:t-1}])$ and $\Gamma_{t-1}([\lambda_{0:t-1}])$ the mean and covariance of the filtering density, respectively. These quantities can be recursively updated as follows:

(i) predictive mean:

$$\vec{\mu}_{t|t-1}[\lambda_{0:t}] = A_{s_t} \vec{\mu}_{t-1}[\lambda_{0:t}]; \quad (32)$$

(ii) predictive covariance:

$$\Gamma_{t|t-1}[\lambda_{0:t}] = A_{s_t} \Gamma_{t-1}[\lambda_{0:t}] A_{s_t}^T + C_{s_t} C_{s_t}^T; \quad (33)$$

(iii) innovation covariance:

$$\Sigma_t[\lambda_{0:t}] = B_{s_t} \Gamma_{t|t-1}[\lambda_{0:t}] B_{s_t}^T + D_{s_t} D_{s_t}^T; \quad (34)$$

(iv) Kalman Gain:

$$K_t[\lambda_{0:t}] = \Gamma_{t|t-1}[\lambda_{0:t}] B_{s_t} (\Sigma_t[\lambda_{0:t}])^{-1}; \quad (35)$$

(v) filtered mean:

$$\vec{\mu}_t[\lambda_{0:t}] = \vec{\mu}_{t-1}[\lambda_{0:t-1}] + K_t[\lambda_{0:t}] (\vec{y}_t - B_{s_t} \vec{\mu}_{t|t-1}[\lambda_{0:t-1}]); \quad (36)$$

(vi) filtered covariance:

$$\Gamma_t[\lambda_{0:t}] = (I - K_t[\lambda_{0:t-1}] B_{s_t}) \Gamma_{t|t-1}[\lambda_{0:t}]. \quad (37)$$

Note that the conditional distribution of the state vector $\vec{X}_t$ given the observations up to time t, $y_{0:t}$, is a mixture of Gaussian distributions with a number of components equal to M^{t+1} which grows exponentially with t. We have now at

hands all the necessary ingredients to derive the GS approximation of the filtering distribution. For any $t \in \mathbb{N}$ and for any $\lambda_{0:t} \in \mathbf{Z}^{t+1}$, denote

$$\gamma_t(\lambda_{0:t}) \stackrel{\text{def}}{=} \begin{cases} f(\vec{y}_0 | \lambda_0) f(\lambda_0) & \text{for } t = 0, \\ f(\vec{y}_t | \lambda_{0:t}, \vec{y}_{0:t-1}) f(\lambda_t | \lambda_{0:t-1}) & \text{for } t > 0. \end{cases} \quad (38)$$

With these notations, (30) reads $q_t(\lambda_{0:t-1}; \lambda_t) \propto \gamma_t(\lambda_{0:t})$.

The first step consists in initializing the particle tracks. For $t = 1$ and $i \in \{1, \ldots, N\}$, set $\vec{\mu}^{(i,0)} = \vec{\mu}_0$ and $\Gamma^{(i,0)} = \Gamma_0$, where $\vec{\mu}_0$ and Γ_0 are the initial mean and variance of the state vector (which are assumed to be known); then, compute the weights

$$w_j = \frac{\gamma_0(z_j)}{\sum_{j'=1}^{M} \gamma_0(z_{j'})}, \quad j \in \{1, \ldots, M\}, \quad (39)$$

and draw $\{I_i, i \in \{1, \ldots, N\}\}$ in such a way that, for $j \in \{1, \ldots, M\}$, $\mathbb{E}[N_j] = Nw_j$, where $N_j = \sum_{i=1}^{N} \delta_{I_i, j}$. Then, set $\Lambda^{(i,0)} = z_{I_i}, i \in \{1, \ldots, N\}$.

At time $t \geq 1$, assume that we have N trajectories $\Lambda^{(i,t-1)} = (\Lambda_0^{(i,t-1)}, \ldots, \Lambda_{t-1}^{(i,t-1)})$ and that, for each trajectory, we have stored the filtered mean $\vec{\mu}^{(i,t-1)}$ and covariance $\Gamma^{(i,t-1)}$ defined in (36) and (37), respectively.

(1) For $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$, compute the predictive mean $\vec{\mu}_{t|t-1}[\Lambda^{(i,t-1)}, z_j]$ and covariance $\Gamma_{t|t-1}[\Lambda^{(i,t-1)}, z_j]$ using (32) and (33), respectively. Then, compute the innovation covariance $\Sigma_t[\Lambda^{(i,t-1)}, z_j]$ using (34) and evaluate the likelihood $\gamma^{(i,j,t)}$ of the particle $[\Lambda^{(i,t-1)}, z_j]$ using (31). Finally, compute the filtered mean and covariance $\vec{\mu}_t([\Lambda^{(i,t-1)}, z_j])$ and $\Gamma_t([\Lambda^{(k,t-1)}, z_j])$.

(2) Compute the weights

$$
\begin{aligned}
w^{(i,j,t)} &= \frac{\gamma^{(i,j,t)}}{\sum_{i'=1}^{N} \sum_{j'=1}^{M} \gamma^{(i',j',t)}}, \\
& i \in \{1, \ldots, N\}, \, j \in \{1, \ldots, M\}.
\end{aligned}
\quad (40)
$$

(3) Draw $\{(I_k, J_k), k \in \{1, \ldots, N\}\}$ using an unbiased sampling procedure (see (26)) with weights $\{w^{(i,j,t)}\}$, $i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}$; set, for $k \in \{1, \ldots, N\}$, $\Lambda^{(k,t)} = (\Lambda^{(I_k,t-1)}, z_{J_k})$. Store the filtered mean and covariance $\vec{\mu}_t([\Lambda^{(k,t)}])$ and $\Gamma_t([\Lambda^{(k,t)}])$ using (36) and (37), respectively.

Remark 1. From the trajectories and the computed weights it is possible to evaluate, for any $\delta \geq 0$ and $t \geq \delta$, the posterior probability of $\Lambda_{t-\delta}$ given $\vec{Y}_{0:t} = \vec{y}_{0:t}$ as

$$\hat{\mathbb{P}}[\Lambda_{t-\delta} = z_k \mid \vec{Y}_{0:t} = \vec{y}_{0:t}]$$

$$
\propto \begin{cases} \sum_{i=1}^{N} w^{(i,k,t)}, & \delta = 0, \text{ filtering}, \\ \sum_{i=1}^{N} \left(\sum_{j=1}^{M} w^{(i,j,t)} \right) \delta_{\Lambda_{t-\delta}^{(i,t-1)}, z_k}, & \delta > 0, \text{ fixed-lag smoothing}. \end{cases}
\quad (41)
$$

Similarly, we can approximate the filtering and the smoothing distribution of the state variable as a mixture of Gaussians. For example, we can estimate the filtered mean and variance of the state as follows:

(i) *filtered mean:*

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w^{(i,j,t)} \vec{\mu}_t([\Lambda^{(i,t-1)}, z_j]);\qquad(42)$$

(ii) *filtered covariance:*

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w^{(i,j,t)} \Gamma_t([\Lambda^{(k,t-1)}, z_j]).\qquad(43)$$

3.3. Fixed-lag smoothing

Since the state process is correlated, the future observations contain information about the current value of the state; therefore, whenever it is possible to delay the decision, fixed-lag smoothing estimates yield more reliable information on the indicator process than filtering estimates.

As pointed out above, it is possible to determine an estimate of the fixed-lag smoothing distribution for any delay δ from the trajectories and the associated weights produced by the SISR or GS method described above; nevertheless, we should be aware that this estimate can be rather poor when the delay δ is large, as a consequence of the impoverishment of the system of particles (the system of particle "forgets" its past). To address this well-known problem in all particle methods, it has been proposed by several authors (see [11, 16, 17, 18]) to sample at time t from the conditional distribution of Λ_t given $\vec{Y}_{0:t+\Delta} = \vec{y}_{0:t+\Delta}$ for some $\Delta > 0$. The computation of fixed-lag smoothing distribution is also amenable to GS approximation.

Consider the distribution of the indicator variables $\Lambda_{0:t}$ conditional to the observations $\vec{Y}_{0:t+\Delta} = \vec{y}_{0:t+\Delta}$, where Δ is a positive integer. Denote by $\{F_t^\Delta\}_{t \geqslant 0}$ this sequence of probability measures; the dependence on the observations $\vec{y}_{0:t+\Delta}$ being, as in the previous section, implicit. This sequence of distributions also satisfies (3), that is, there exists a finite transition kernel $Q_t^\Delta : (\mathbf{Z}^t, \mathcal{P}(\mathbf{Z})^{\otimes t}) \prec (\mathbf{Z}, \mathcal{P}(\mathbf{Z}))$ such that $F_t^\Delta = F_{t-1}^\Delta \otimes Q_t^\Delta$ for all $t \geqslant 1$. Elementary conditional probability calculations exploiting the conditional independence structure of (28) show that the transition kernel Q_t^Δ can be determined, up to a normalization constant, by the relation

$$Q_t^\Delta(\lambda_{0:t-1}; \lambda_t) \propto \frac{\sum_{\lambda_{t+1:t+\Delta}} \prod_{\tau=t}^{t+\Delta} f(\vec{y}_\tau | \vec{y}_{0:\tau-1}, \lambda_{0:\tau}) f(\lambda_\tau | \lambda_{0:\tau-1})}{\sum_{\lambda_{t:t+\Delta-1}} \prod_{\tau=t}^{t+\Delta-1} f(\vec{y}_\tau | \vec{y}_{0:\tau-1}, \lambda_{0:\tau}) f(\lambda_\tau | \lambda_{0:\tau-1})},\qquad(44)$$

where, for all $\lambda_{0:t-1} \in \mathbf{Z}^t$, the terms $f(\vec{y}_\tau | \vec{y}_{0:\tau-1}, \lambda_{0:\tau})$ can be determined recursively using Kalman filter fixed-lag smoothing update formula.

Below, we describe a straightforward implementation of the GS method to approximate the smoothing distribution by the delayed sampling procedure; more sophisticated techniques, using early pruning of the possible prolonged trajectories, are currently under investigation. For any $t \in \mathbb{N}$ and for any $\lambda_{0:t} \in \mathbf{Z}^{t+1}$, denote

$$D_t^\Delta(\lambda_{0:t}) \stackrel{\text{def}}{=} \sum_{\lambda_{t+1:t+\Delta}} \prod_{\tau=t+1}^{t+\Delta} \gamma_\tau(\lambda_{0:\tau}),\qquad(45)$$

where the function γ_τ is defined in (38). With this notation, (44) may be rewritten as

$$Q_t^\Delta(\lambda_{0:t-1}; \lambda_t) \propto \gamma_t(\lambda_{0:t}) \frac{D_t^\Delta(\lambda_{0:t})}{D_{t-1}^\Delta(\lambda_{0:t-1})}.\qquad(46)$$

We now describe one iteration of the algorithm. Assume that for some time instant $t \geqslant 1$, we have N trajectories $\Lambda^{(j,t-1)} = (\Lambda_0^{(j,t-1)}, \ldots, \Lambda_{t-1}^{(j,t-1)})$; in addition, for each trajectory $\Lambda^{(j,t-1)}$, the following quantities are stored:

(1) the factor $D_{t-1}^\Delta(\Lambda^{(j,t-1)})$ defined in (45);
(2) for each prolongation $\lambda_{t:\tau} \in \mathbf{Z}^{\tau-t+1}$ with $\tau \in \{t, t+1, \ldots, t+\Delta-1\}$, the conditional likelihood $\gamma_\tau(\Lambda^{(j,t-1)}, \lambda_{t:\tau})$ given in (38);
(3) for each prolongation $\lambda_{t:t+\Delta-1} \in \mathbf{Z}^\Delta$, the filtering conditional mean $\vec{\mu}_{t+\Delta-1}([\Lambda^{(j,t-1)}, \lambda_{t:t+\Delta-1}])$ and covariance $\Gamma_{t+\Delta-1}(\Lambda^{(j,t-1)}, \lambda_{t:t+\Delta-1})$.

One iteration of the algorithm is then described below.

(1) For each $i \in \{1, \ldots, N\}$ and for each $\lambda_{t:t+\Delta} \in \mathbf{Z}^{\Delta+1}$, compute the predictive conditional mean and covariance of the state, $\vec{\mu}_{t+\Delta|t+\Delta-1}([\Lambda^{(i,t-1)}, \lambda_{t:t+\Delta}])$ and $\Gamma_{t+\Delta|t+\Delta-1}([\Lambda^{(i,t-1)}, \lambda_{t:t+\Delta}])$, using (32) and (33), respectively. Then compute the innovation covariance $\Sigma_{t+\Delta}[(\Lambda^{(i,t-1)}, \lambda_{t:t+\Delta})]$ using (34) and the likelihood $\gamma_{t+\Delta}(\Lambda^{(j,t-1)}, \lambda_{t:t+\Delta})$ using (31).
(2) For each $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$, compute

$$D_t^\Delta(\Lambda^{(i,t-1)}, z_j) = \sum_{\lambda_{t+1:t+\Delta}} \prod_{\tau=t+1}^{t+\Delta} \gamma_\tau([\Lambda^{(i,t-1)}, z_j, \lambda_{t+1:t+\tau}]),$$

$$\gamma^{(i,j,t)} = \gamma_t(\Lambda^{(i,t-1)}, z_j) \frac{D_t^\Delta(\Lambda^{(i,t-1)}, z_j)}{D_{t-1}^\Delta(\Lambda^{(i,t-1)})},$$

$$w^{(i,j,t)} = \frac{\gamma^{(i,j,t)}}{\sum_{i'=1}^{M} \sum_{j=1}^{N} \gamma^{(i',j',t)}}.\qquad(47)$$

(3) Update the trajectory of particles using an unbiased sampling procedure $\{(I_k, J_k), k \in \{1, \ldots, N\}\}$ with weights $\{w^{(i,j,t)}\}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}$, and set $\Lambda^{(k,t)} = (\Lambda^{(I_k,t-1)}, z_{J_k}), k \in \{1, \ldots, N\}$.

4. SOME EXAMPLES

4.1. *Autoregressive model with jumps*

To illustrate how the GS method works, we consider the state-space model

$$X_t = a_{\Lambda_t} X_{t-1} + \sigma_{\Lambda_t} \epsilon_t,$$
$$Y_t = X_t + \rho \eta_t,$$
(48)

where $\{\epsilon_t\}_{t \geq 0}$ and $\{\eta_t\}_{t \geq 0}$ are i.i.d. unit-variance Gaussian noise. We assume that $\{\Lambda_t\}_{t \geq 0}$ is an i.i.d. sequence of random variables taking their values in $\mathbf{Z} \stackrel{\text{def}}{=} \{1, 2\}$, which is independent from both $\{\epsilon_t\}_{t \geq 0}$ and $\{\eta_t\}_{t \geq 0}$, and such that $\mathbb{P}[\Lambda_0 = i] = \pi_i$, $i \in \mathbf{Z}$. This can easily be extended to deal with the Markovian case. This simple model has been dealt with, among others, in [19] and [20, Section 5.1]. We focus in this section on the filtering problem, that is, we approximate the distribution of the hidden state X_t given the observations up to time t, $Y_{0:t} = y_{0:t}$. For this model, we can carry out the computations easily. The transition kernel q_t defined in (30) is given, for all $\lambda_{0:t-1} \in \mathbf{Z}^t$, $\lambda_t \in \mathbf{Z}$, by

$$q_t(\lambda_{0:t-1}, \lambda_t) \propto \frac{\pi_{\lambda_t}}{\sqrt{2\pi\Sigma_t[\lambda_{0:t}]}} \exp\left(-\frac{(y_t - \mu_{t|t-1}[\lambda_{0:t}])^2}{2\Sigma_t[\lambda_{0:t}]} \right),$$
(49)

where the mean $\mu_{t|t-1}[\lambda_{0:t}]$ and covariance $\Sigma_t[\lambda_{0:t}]$ are computed recursively from the filtering mean $\mu_{t-1}([\lambda_{0:t-1}])$ and covariance $\Gamma_{t-1}([\lambda_{0:t-1}])$ according to the following one-step Kalman update equations derived from (32), (33), and (34):

(i) predictive mean:

$$\mu_{t|t-1}[\lambda_{0:t}] = a_{\lambda_t}\mu_{t-1}[\lambda_{0:t}];$$
(50)

(ii) predictive covariance:

$$\Gamma_{t|t-1}[\lambda_{0:t}] = a_{\lambda_t}^2 \Gamma_{t-1}[\lambda_{0:t}] + \sigma_{\lambda_t}^2;$$
(51)

(iii) innovation covariance:

$$\Sigma_t[\lambda_{0:t}] = \Gamma_{t|t-1}[\lambda_{0:t}] + \rho^2;$$
(52)

(iv) filtered mean:

$$\mu_t[\lambda_{0:t}] = \mu_{t|t-1}[\lambda_{0:t-1}] + \frac{\Gamma_{t|t-1}([\lambda_{0:t}])}{\Gamma_{t|t-1}([\lambda_{0:t}]) + \rho^2}$$
$$\times (y_t - \mu_{t|t-1}[\lambda_{0:t-1}]);$$
(53)

(v) filtered covariance:

$$\Gamma_t[\lambda_{0:t}] = \frac{\rho^2 \Gamma_{t|t-1}([\lambda_{0:t}])}{\Gamma_{t|t-1}([\lambda_{0:t}]) + \rho^2}.$$
(54)

We have used the parameters (used in the experiments carried out in [20, Section 5.1]): $a_i = 0.9$ ($i = 1, 2$), $\sigma_1 = 0.5$,

$\sigma_2 = 1.5$, $\pi_1 = 1.7$, and $\rho = 0.3$, and applied the GS and the SISR algorithm for online filtering. We compare estimates of the filtered state mean using the GS and the SIS with systematic resampling. In both case, we use the estimator (42) of the filtered mean. Two different unbiased sampling strategies are used: multinomial sampling and the modified stratified sampling (detailed in the appendix).[1] In Figure 1, we have displayed the box and whisker plot[2] of the difference between the filtered mean estimate (42) and the true value of the state variables for $N = 5, 10, 50$ particles using multinomial sampling (Figure 1a) and the modified stratified sampling (Figure 1b). These results are obtained from 100 hundred independent Monte Carlo experiments where, for each experiment, a new set of the observations and state variables are simulated. These simulations show that, for the autoregressive model, the filtering algorithm performed reasonably well even when the number of particles is small (the difference between $N = 5$ and $N = 50$ particles is negligible; $N = 50$ particles is suggested in the literature for the same simulation setting [20]). There are no noticeable differences between the standard SISR implementation and the GS implementation of the SISR. Note that the error in the estimate is dominated by the filtered variance $\mathbb{E}[(X_t - \mathbb{E}[X_t|Y_{0:t}])^2]$; the additional variations induced by the fluctuations of the particle estimates are an order of magnitude lower than this quantity.

To visualize the difference between the different sampling schemes, it is more appropriate to consider the fluctuation of the filtered mean estimates around their sample mean for a given value of the time index and of the observations. In Figure 2, we have displayed the box and whisker plot of the error at time index 25 between the filtered mean estimates and their sample mean at each time instant; these results have been obtained from 100 independent particles (this time, the set of observations and of states are held fixed over all the Monte Carlo simulations). As above, we have used $N = 5, 10, 50$ of particles and two sampling methods: multinomial sampling (Figure 2a) and modified stratified sampling (Figure 2b). This figure shows that the GS estimate of the sampled mean has a lower standard deviation than any other estimators included in this comparison, independently of the number of particles which are used. The differences between these estimators are however small compared to the filtering variance.

4.2. *Joint channel equalization and symbol detection on a flat Rayleigh-fading channel*

4.2.1. *Model description*

We consider in this section a problem arising in transmission over a Rayleigh-fading channel. Consider a communi-

---

[1]The Matlab code to reproduce these experiments is available at http://www.tsi.enst.fr/~moulines/.

[2]The lower and upper limits of the box are the quartiles; the horizontal line in the box is the sample median; the upper and lower whiskers are at 3/2 times interquartiles.
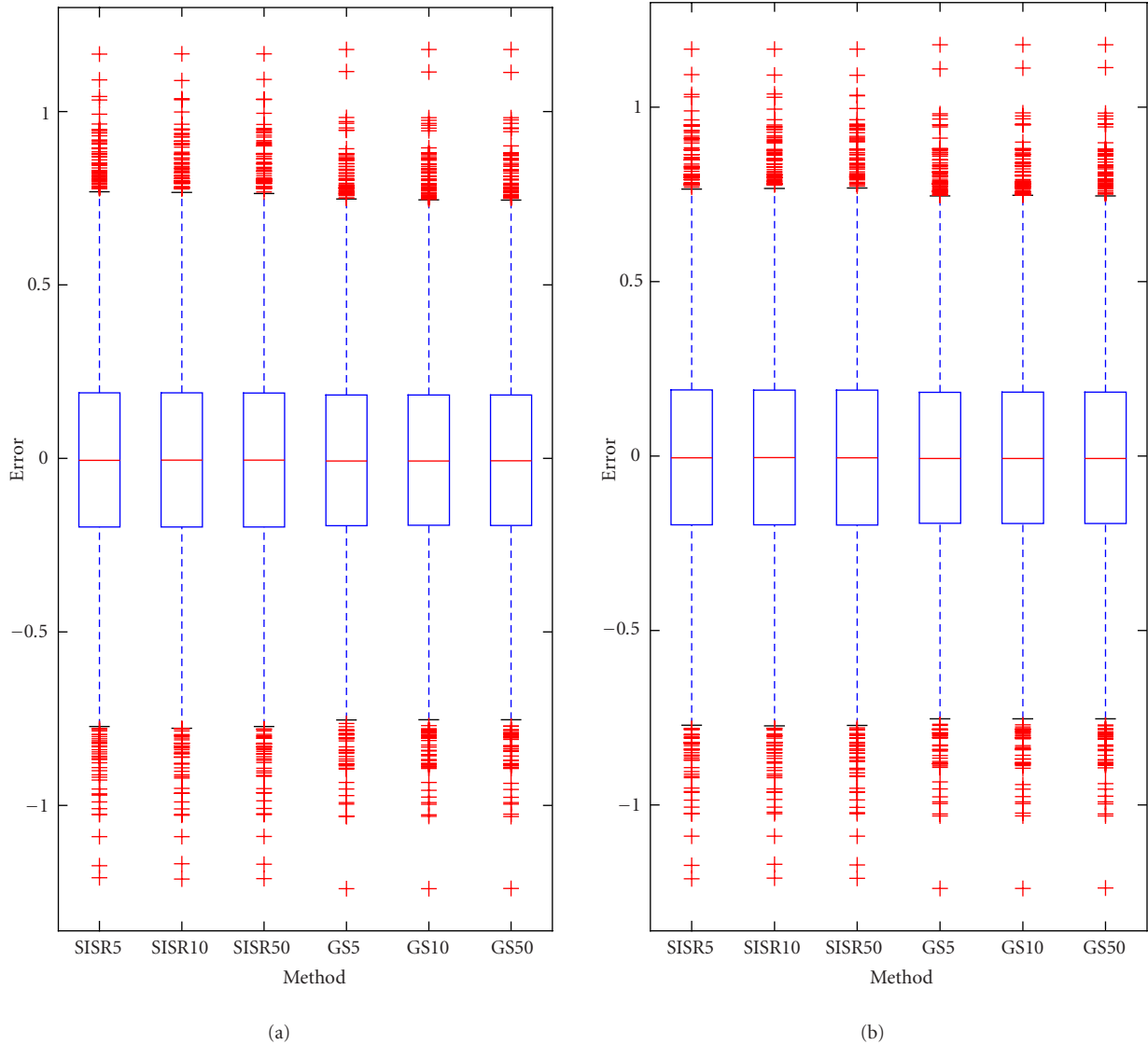
FIGURE 1: Box and whisker plot of the difference between the filtered mean estimates and the actual value of the state estimate for 100 independent Monte Carlo experiments. (a) Multinomial sampling. (b) Residual sampling with the modified stratified sampling.

cation system signaling through a flat-fading channel with additive noise. In this context, the indicator variables $\{\Lambda_t\}$ in the representation (28) are the input bits which are transmitted over the channel and $\{S_t\}_{t\geq 0}$ are the symbols generally taken into an M-ary complex alphabet. The function Ψ_t is thus the function which maps the stream of input bits into a stream of complex symbols: this function combines *channel encoding* and *symbol mapping*. In the simple example considered below, we assume binary phase shift keying (BPSK) modulation with differential encoding: $S_t = S_{t-1}(2\Lambda_t - 1)$. The input-output relationship of the flat-fading channel is described by

$$Y_t = \alpha_t S_t + V_t, \tag{55}$$

where Y_t, α_t, S_t, and V_t denote the received signal, the fading channel coefficient, the transmitted symbol, and the additive noise at time t, respectively. It is assumed in the sequel that

(i) the processes $\{\alpha_t\}_{t\geq 0}$, $\{\Lambda_t\}_{t\geq 0}$, and $\{V_t\}_{t\geq 0}$ are mutually independent;

(ii) the noise $\{V_t\}$ is a sequence of i.i.d. zero-mean complex random variables $V_t \sim \mathcal{N}_c(0, \sigma_V^2)$.

It is further assumed that the channel fading process is Rayleigh, that is, $\{\alpha_t\}$ is a zero-mean complex Gaussian process; here modelled as an ARMA(L, L),

$$\alpha_t - \phi_1\alpha_{t-1} - \cdots - \phi_L\alpha_{t-L} = \theta_0\eta_t + \theta_1\eta_{t-1} + \cdots + \theta_L\eta_{t-L}, \tag{56}$$
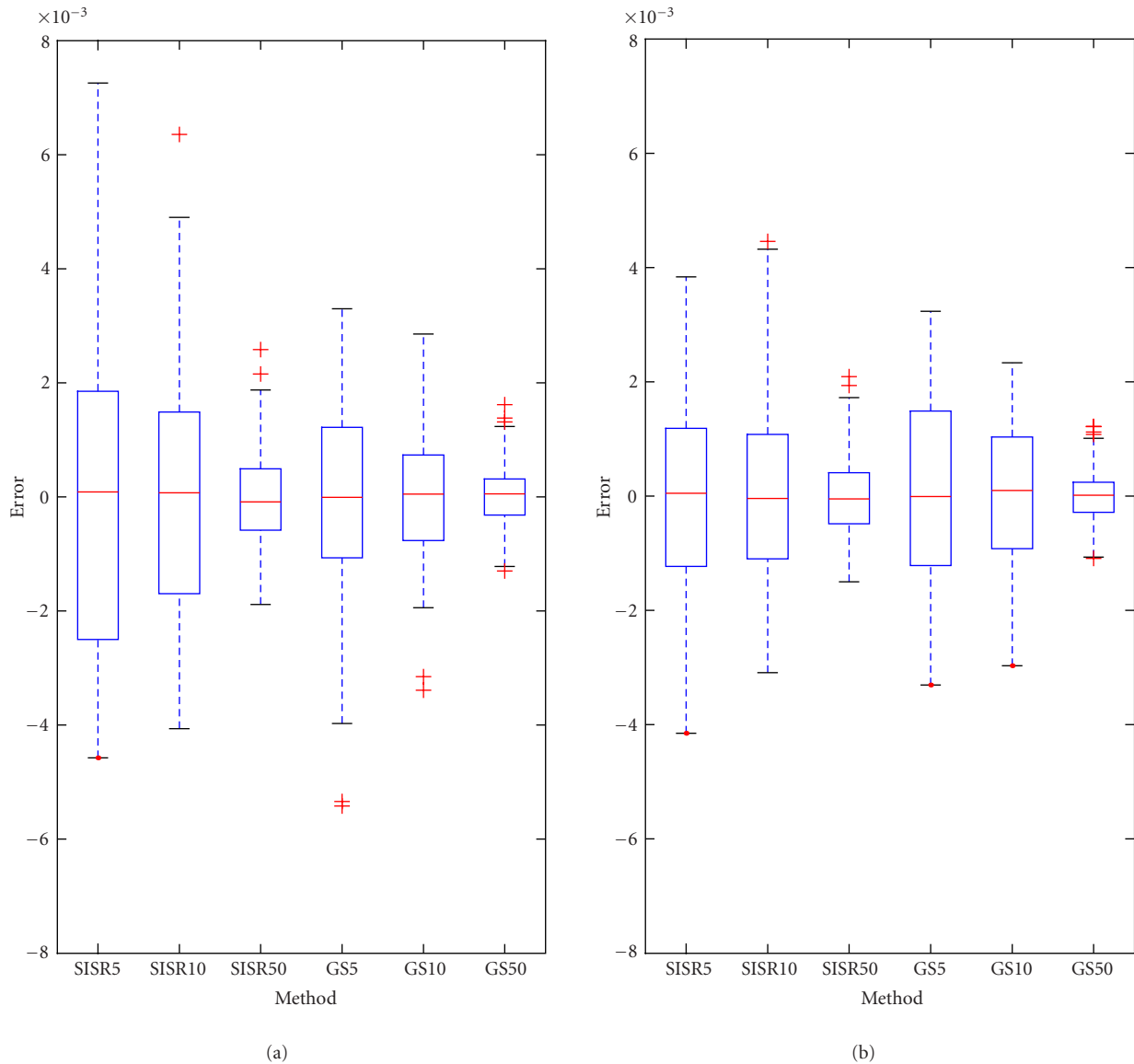
Figure 2: Box and whisker plot of the difference between the filtered mean estimates and their sample mean for 100 independent particles for a given value of the time index 25 and of the observations. (a) Multinomial sampling. (b) Residual sampling with the modified stratified sampling.

where $\phi_1, \ldots, \phi_L$ and $\theta_0, \ldots, \theta_L$ are the autoregressive and the moving average (ARMA) coefficients, respectively, and $\{\eta_t\}$ is a white complex Gaussian noise with zero mean and unit variance. This model can be written in state-space form as follows:

$$\vec{X}_{t+1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ \phi_L & \phi_{L-1} & \cdots & & \phi_1 \end{bmatrix} \vec{X}_t + \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_L \end{bmatrix} \eta_t,$$

$$\alpha_t = \begin{bmatrix} 10 & \cdots & 0 \end{bmatrix} \vec{X}_t + \eta_t,$$

(57)

where $\{\psi_k\}_{1 \le k \le m}$ are the coefficients of the expansion of $\theta(z)/\phi(z)$, for $|z| \le 1$, with

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p,$$
$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q.$$

(58)

This particular problem has been considered, among others, in [10, 16, 18, 21, 22].

4.2.2. Simulation results

To allow comparison with previously reported work, we consider the example studied in [16, Section VIII]. In this

example, the fading process is modelled by the output of a Butterworth filter of order $L = 3$ whose cutoff frequency is 0.05, corresponding to a normalized Doppler frequency $f_d T = 0.05$ with respect to the symbol rate $1/T$, which is a fast-fading scenario. More specifically, the fading process is modelled by the ARMA(3, 3) process

$$\alpha_t - 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3}$$
$$= 10^{-2}(0.89409\eta_t + 2.68227\eta_{t-1} \qquad (59)$$
$$+ 2.68227\eta_{t-2} + 0.89409\eta_{t-3}),$$

where $\eta_t \sim \mathcal{N}_c(0, 1)$. It is assumed that a BPSK modulation is used, that is, $S_t \in \{-1, +1\}$, with differential encoding and no channel code; more precisely, we assume that $S_t = S_{t-1}\Lambda_t$, where $\Lambda_t \in \{-1, +1\}$ is the bit sequence, assumed to be i.i.d. Bernoulli random variables with probability of success $\mathbb{P}(\Lambda_t = 1) = 1/2$.

The performance of the GS receiver (using the modified residual sampling algorithm) has been compared with the following receiver schemes.

(1) *Known channel lower bound.* We assume that the true fading coefficients α_t are known to the receiver and we calculate the optimal coherent detection rule $\hat{S}_t = \text{sign}(\Re\{\alpha_t^* \vec{Y}_t\})$ and $\hat{\Lambda}_t = \hat{S}_t \hat{S}_{t-1}$.

(2) *Genie-aided lower bound.* We assume that a genie allows the receiver to observe $\tilde{Y}_t = \alpha_t + \tilde{V}_t$, with $\tilde{V}_t \sim \mathcal{N}_c(0, \sigma_V^2)$. We use $\tilde{Y}_t$ to calculate an estimate $\hat{\alpha}_t$ of the fading coefficients via a Kalman filter and we then evaluate the optimal coherent detection $\hat{S}_t = \text{sign}(\Re\{\hat{\alpha}_t^* \vec{Y}_t\})$ and $\hat{\Lambda}_t = \hat{S}_t \hat{S}_{t-1}$ using the filtered fading process.

(3) *Differential detector.* In this scenario, no attempt is made to estimate the fading process and the input bits are estimated using incoherent differential detection: $\hat{\Lambda}_t = \text{sign}(\Re\{Y_t^* \vec{Y}_{t-1}\})$.

(4) *MKF detector.* The SMC filter described in [16, Sections IV and V] is used to estimate Λ_t. The MKF detector uses the SISR algorithm to draw samples in the indicator space and implements a Kalman filter for each trajectory in order to compute its trial sampling density and its importance weight. Resampling is performed when the ratio between the effective sample size defined in [16, equation (45)] and the actual sample size N is lower than a threshold β. The delayed weight method is used to obtain an estimate of Λ_t with a delay δ.

In all the simulations below, we have used only the concurrent sampling method because in the considered simulation scenarios, the use of the delayed sampling method did not bring significative improvement. This is mainly due to the fact that we have only considered, due to space limitations, the uncoded communication scenario.

Figure 3 shows the BER performance of each receiver versus the SNR. The SNR is defined as $\text{var}(\alpha_t)/\text{var}(V_t)$ and the BER is obtained by averaging the error rate over 10^6 symbols. The first 50 symbols were not taken into account in
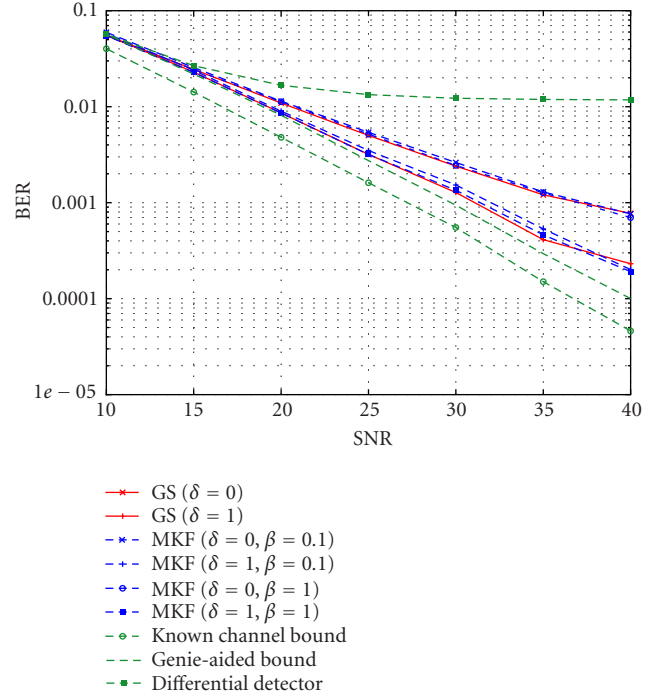

Figure 3: BER performance of the GS receiver versus the SNR. The BER corresponding to delays $\delta = 0$ and $\delta = 1$ are shown. Also shown in this figure are the BER curves for the MKF detector ($\delta = 0$ and $\delta = 1$), the known channel lower bound, the genie-aided lower bound, and the differential detector. The number of particles for the GS receiver and the MKF detector is 50.

counting the BER. The BER performance of the GS receiver is shown for estimation delays $\delta = 0$ (concurrent estimation) and $\delta = 1$. Also shown are the BER curves for the known channel lower bound, the genie-aided lower bound, the differential detector, and the MKF detector with estimation delays $\delta = 0$ and $\delta = 1$ and resampling thresholds $\beta = 0.1$ and $\beta = 1$ (systematic resampling). The number of particles for both the GS receiver and the MKF detector is set to 50. From this figure, it can be seen that with 50 particles, there is no significant performance difference between the proposed receiver and the MKF detector with the same estimation delay and $\beta = 0.1$ or $\beta = 1$. Note that, as observed in [16], the performance of the receiver is significantly improved by the delayed-weight method with $\delta = 1$ compared with concurrent estimate; there is no substantial improvement when increasing further the delay; the GS receiver achieves essentially the genie-aided bound over the considered SNR.

Figure 4 shows the BER performance of the GS receiver versus the number of particles at SNR = 20 dB and $\delta = 1$. Also shown in this figure is the BER performance for the MKF detector with $\beta = 0.1$ and $\beta = 1$, respectively. It can be seen from this plot that when the number of particles is decreased from 50 to 10, the BER of the MKF receiver with $\beta = 0.1$ increases by 67%, whereas the BER of the GS receiver increases by 11% only. In fact, Figure 4 also shows that, for this particular example, the BER performance of the GS receiver is identical to the BER performance of an MKF with
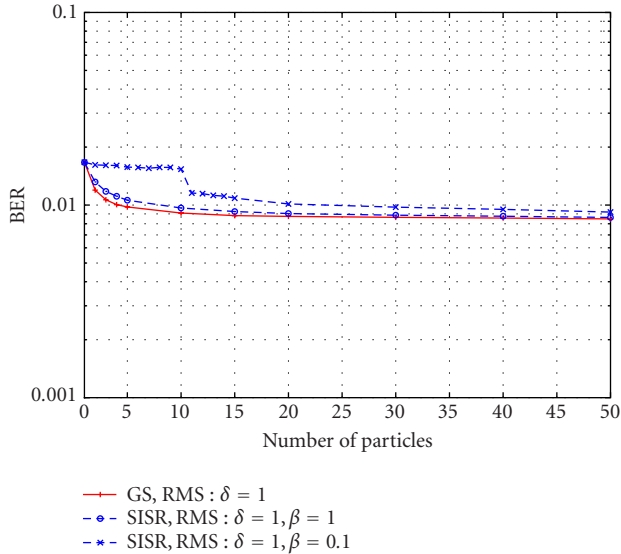
FIGURE 4: BER performance of the GS receiver versus the number of particles at SNR = 20 dB and $\delta = 1$. Also shown in this figure are the BER curves for the MKF detector with $\beta = 0.1$ and $\beta = 1$.

the same number of particles and a resampling threshold set to $\beta = 1$ (systematic resampling). This suggests that, contrary to what is usually argued in the literature [5, 16], systematic resampling of the particle seems to be, for reasons which remain yet unclear from a theoretical standpoint, more robust when the number of particles is decreased to meet the constraints of real-time implementation.

Figure 5 shows the BER performance of each receiver versus the SNR when the number of particles for both the GS receiver and the MKF detector is set to 5. For these simulations, the BER is obtained by averaging the error rate over 10^5 symbols. From this figure, it can be seen that with 5 particles, there is a significant performance difference between the proposed receiver and the MKF detector with the same estimation delay and a $\beta = 0.1$ resampling threshold. This difference remains significant even for SNR values close to 10 dB. Figure 5 also shows that, for this particular example, the BER performance of the GS receiver is identical to the BER performance of an MKF with the same estimation delay and a resampling threshold β set to 1.

5. CONCLUSION

In this paper, a sampling algorithm for conditionally linear Gaussian state-space models has been introduced. This algorithm exploits the particular structure of the flow of probability measures and the fact that, at each time instant, a global exploration of all possible offsprings of a given trajectory of indicator variables can be considered. The number of trajectories is kept constant by sampling from this set (selection step).

The global sampling algorithm appears, in the example considered here, to be robust even when a very limited number of particles is used, which is a basic requirement for
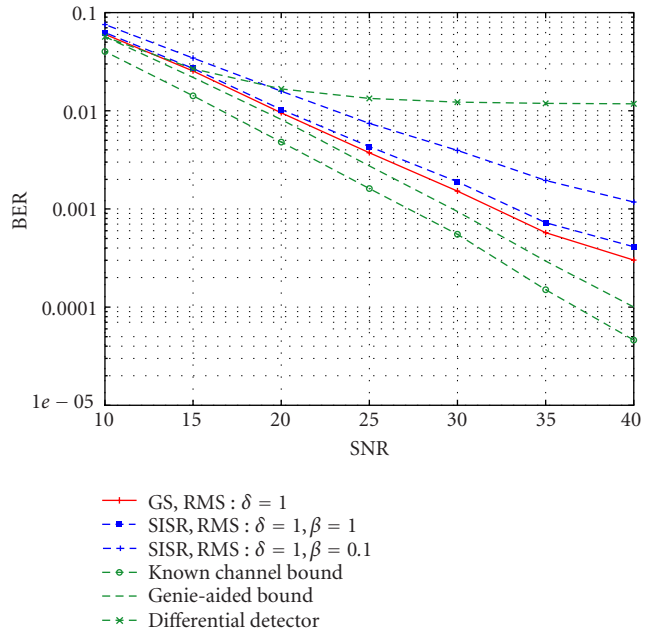

FIGURE 5: BER performance of the GS receiver versus the SNR. The BER corresponding to delay $\delta = 1$ is shown. Also shown in this figure are the BER curves for the MKF detector ($\delta = 1$, $\beta = 0.1$), the known channel lower bound, the genie-aided lower bound, and the differential detector. The number of particles for the GS receiver and the MKF detector is 5.

the implementation of such a solution in real-world applications: the global sampling algorithm is close to the optimal genie-aided bound with as few as 5 particles and thus provides a realistic alternative to the joint channel equalization and symbol detection algorithms reported earlier in the literature.

APPENDIX

MODIFIED STRATIFIED SAMPLING

In this appendix, we present the so-called modified stratified sampling strategy. Let M and N be integers and $(w_1, \ldots, w_M)$ be nonnegative weights such that $\sum_{i=1}^{M} w_i = 1$. A sampling procedure is said to be unbiased if the random vector $(N_1, \ldots, N_M)$ (where N_i is the number of times the index i is drawn) satisfies

$$\sum_{i=1}^{M} N_i = N, \quad \mathbb{E}[N_i] = N w_i, \quad i \in \{1, \ldots, M\}. \quad (A.1)$$

The modified stratified sampling is summarized as follows.

(1) For $i \in \{1, \ldots, M\}$, compute $[N w_i]$, where $[x]$ is the integer part of x; then compute the residual number $\tilde{N} = N - \sum_{i=1}^{M} [N w_i]$ and the residual weights

$$\tilde{w}_i = \frac{N w_i - [N w_i]}{\tilde{N}}, \quad i \in \{1, \ldots, M\}. \quad (A.2)$$

(2) Draw $\tilde{N}$ i.i.d. random variables $U_1, \ldots, U_{\tilde{N}}$ with a uniform distribution on $[0, 1/\tilde{N}]$ and compute, for $k \in \{1, \ldots, \tilde{N}\}$,

$$\tilde{U}_k = \frac{k-1}{\tilde{N}} + U_k. \qquad (A.3)$$

(3) For $i \in \{1, \ldots, M\}$, set $\tilde{N}_i$ as the number of indices $k \in \{1, \ldots, \tilde{N}\}$ satisfying

$$\sum_{j=1}^{i-1} w_j < \tilde{U}_k \le \sum_{j=1}^{i} w_j. \qquad (A.4)$$

REFERENCES

[1] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, Englewood Cliffs, NJ, USA, 1st edition, 2000.

[2] I. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-Valued Time Series*, vol. 70 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, London, UK, 1997.

[3] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., pp. 3–13, Springer, New York, NY, USA, January 2001.

[4] C. Carter and R. Kohn, "Markov chain Monte Carlo in conditionally Gaussian state space models," *Biometrika*, vol. 83, no. 3, pp. 589–601, 1996.

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

[6] N. Shephard, "Partial non-Gaussian state space," *Biometrika*, vol. 81, no. 1, pp. 115–131, 1994.

[7] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal American Statistical Association*, vol. 93, no. 444, pp. 1032–1044, 1998.

[8] R. Chen and J. Liu, "Mixture Kalman filter," *Journal Royal Statistical Society Series B*, vol. 62, no. 3, pp. 493–508, 2000.

[9] G. Rigal, *Filtrage non-linéaire, résolution particulaire et applications au traitement du signal*, Ph.D. dissertation, Université Paul Sabatier, Toulouse, France, 1993.

[10] J. Liu and R. Chen, "Blind deconvolution via sequential imputations," *Journal American Statistical Association*, vol. 430, no. 90, pp. 567–576, 1995.

[11] E. Punskaya, C. Andrieu, A. Doucet, and W. Fitzgerald, "Particle filtering for multiuser detection in fading CDMA channels," in *Proc. 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 38–41, Orchid Country Club, Singapore, August 2001.

[12] V. Zaritskii, V. Svetnik, and L. Shimelevich, "Monte-Carlo technique in problems of optimal information processing," *Automation and Remote Control*, vol. 36, no. 12, pp. 95–103, 1975.

[13] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, no. 4, pp. 429–434, 1977.

[14] J. Tugnait, "Adaptive estimation and identification for discrete systems with markov jump parameters," *IEEE Trans. Automatic Control*, vol. 27, no. 5, pp. 1054–1065, 1982.

[15] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

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

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

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

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

[20] J. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling and resampling," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., Springer, New York, NY, USA, 2001.

[21] F. Ben Salem, "Récepteur particulaire pour canaux mobiles évanescents," in *Journées Doctorales d'Automatique (JDA '01)*, pp. 25–27, Toulouse, France, September 2001.

[22] F. Ben Salem, *Réception particulaire pour canaux multi-trajets évanescents en communication radiomobile*, Ph.D. dissertation, Université Paul Sabatier, Toulouse, France, 2002.

Pascal Cheung-Mon-Chan graduated from the Ecole Normale Supérieure de Lyon in 1994 and received the Diplôme d'Ingénieur from the École Nationale Supérieure des Télécommunications (ENST) in Paris in the same year. After working for General Electric Medical Systems as a Research and Development Engineer, he received the Ph.D. degree from the ENST in Paris in 2003. He is currently a member of the research staff at France Telecom Research and Development.

Eric Moulines was born in Bordeaux, France, in 1963. He received the M.S. degree from Ecole Polytechnique in 1984, the Ph.D. degree from École Nationale Supérieure des Télécommunications (ENST) in 1990 in signal processing, and an "Habilitation à Diriger des Recherches" in applied mathematics (probability and statistics) from Université René Descartes (Paris V) in 1995. From 1986 until 1990, he was a member of the technical staff at Centre National de Recherche des Télécommunications (CNET). Since 1990, he was with ENST, where he is presently a Professor (since 1996). His teaching and research interests include applied probability, mathematical and computational statistics, and signal processing.