RAO-BLACKWELLISED PARTICLE FILTERS: EXAMPLES OF APPLICATIONS

Frédéric Mustière
School of Information Technology and Engineering, University of Ottawa
e-mail: mustiere@site.uottawa.ca

Miodrag Bolić
School of Information Technology and Engineering, University of Ottawa
e-mail: mbolic@site.uottawa.ca

Martin Bouchard
School of Information Technology and Engineering, University of Ottawa
e-mail: bouchard@site.uottawa.ca

Abstract

In this work, we present some examples of applications of the so-called Rao-Blackwellised Particle Filter (RBPF). RBPFs are an extension to Particle Filters (PFs) which are applicable to conditionally linear-Gaussian state-space models.

Although RBPF introductions and reviews may be found in many existing sources, going through the specific vocabulary and concepts of particle filtering can sometimes prove to be time-consuming for the non-initiated reader willing to experiment with alternative algorithms.

The goal of the paper is to introduce RBPF-based methods in an accessible manner via a main algorithm, which is detailed enough to be readily applied to a wide range of problems. To illustrate the practicality and the convenience of the approach, the algorithm is then tailored to two examples from different fields. The first example is related to system identification, and the second is an application of speech enhancement.

Keywords — *Particle filters, RBPF, Rao Blackwellised particle filters, tutorial, system identification, speech enhancement.*

1 Introduction

Particle filters are a family of algorithms operating on systems which can be modelled by discrete time state-space equations, where the evolution of the state $\mathbf{x}$ and its relationship to the measurement $\mathbf{z}$ is conveniently represented as follows:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_k) \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (2)$$

PFs can compute online an approximation of the state through equations (1) and (2) for weak assumptions. Functions f and h are assumed to be known, and may depend on time, $\mathbf{v}_k$ and $\mathbf{w}_k$ are noises which may be, theoretically, of any nature (although non-Gaussian noises may complicate substantially the derivation of the PF algorithm). Provided that the state $\mathbf{x}$ contains enough information, a large class of problems can be written in the form of equations (1) and (2). Rao-Blackwellised Particle Filters (RBPFs), can be seen as a form of constrained PFs applicable to a subclass of state-space models, where equations (1) and (2) can be written in a conditionally linear-Gaussian form (see section 2 below). In such cases, RBPFs allow the use of less particles to obtain a performance similar to that of PFs, even though more computations must be carried out per particle. Reviews of RBPF algorithms can be found in many sources, including [2–4, 9], however the experience of the authors is that some sources, as detailed and rigorous as they are, can still appear quite intimidating to the reader not initiated to the nomenclature and the many concepts specific to particle filtering. The attempt here is to present the algorithm in a most accessible way.

In the following pages, a practical RBPF algorithm is presented, along with a description of the range of problems aimed. We then apply the algorithm to two detailed examples from various fields, for which a tailored RBPF is derived from the generic algorithm. In the first example, we show how RBPFs can be applied to a non-linear system identification problem, where an unknown time-varying FIR filter is applied to an unknown signal, evolving in a known non-linear fashion. As a second example, we choose to present a basic RBPF-based solution to speech denoising problems, which is the basis for more complex existing algorithms, such as the ones presented in [3, 7, 9]. Concrete simulation results are presented on both cases. We conclude on the advantages of using RBPFs in the light of the examples shown.

2 A RBPF Algorithm

We choose not to repeat here the generic PF algorithm, although we briefly present the idea. (The PF algorithm can be found in details in several sources, e.g. [1, 2]). One may see particle filtering as a genetic algorithm, a "brute force" simulation. At each step, the algorithm draws a large number of possible candidates for the state. As a measurement is received, scores (or weights) are assigned to the candidates, depending on how well they fit the measurement – and in fact, the sequence of past measurements. Only the fittest candidates (in a probabilistic sense) survive onto the next step, and the other ones are discarded.

A Rao-Blackwellised particle filter can be seen as an enhanced PF applicable to a wide range of problems in which dependencies within the state vector can be analytically exploited.

In practical terms, suppose that we are able to model the evolution of a quantity of interest, $\mathbf{x_2}_k$, using the following time-varying equations:

$$\mathbf{x_2}_k = \mathbf{A}_k \mathbf{x_2}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \quad (3)$$

$$\mathbf{z}_k = \mathbf{C}_k \mathbf{x_2}_k + \mathbf{D}_k \mathbf{u}_k + \mathbf{H}_k \mathbf{v}_k \quad (4)$$

but with one or more of the parameters $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$ or $\mathbf{u}_k$ unknown, and evolving possibly under non-linear, non-Gaussian conditions. In (3) and (4), $\mathbf{w}$ and $\mathbf{v}$ are

1-4244-0038-4 2006

IEEE CCECE/CCGEI, Ottawa, May 2006

zero mean, unit covariance Gaussian random vectors, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{G}$, and $\mathbf{H}$ have the dimension of matrices, and $\mathbf{u}$ that of a vector[1]. This type of system is sometimes termed conditionally linear-Gaussian. In this situation, a possibility to solve the problem is to form another set of variables describing those parameters, say $\mathbf{x_1}_k$, and then to apply a RBPF algorithm on the whole state $\mathbf{x}_k = \{\mathbf{x_1}_k; \mathbf{x_2}_k\}$. To be able to do so, there are two pre-requisites. First, we must be able to define the probability density $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$, that is, we must either know or conjecture the evolution of the unknown variables $\mathbf{x_1}_k$[2]. Secondly, it must be possible to draw samples from $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$. Note that the elements of $\mathbf{x_1}_k$ do not have to be identical to the unknown parameters among $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$, and $\mathbf{u}_k$ – the only requirement is that of a one-to-one relationship. Also note that a Kalman filter alone cannot be used to solve this problem, although it may be possible and convenient to use a combination of algorithms – an algorithm estimating $\mathbf{x_1}_k$, serving a Kalman filter running on $\mathbf{x_2}_k$. In contrast, RBPFs can be seen as a holistic solution.

A generic RBPF algorithm, applicable to problems of the form (3) and (4) is presented in Algorithm 1[3]. A justificative explanation for the algorithm is presented in appendix B. At every iteration of the algorithm, a set of N particles is maintained and updated. At instant k, the i^{th} particle is defined as $\{\mathbf{x_1}_{k,i}\,;\,\mathbf{x_2}_{k,i}\,;\,\mathbf{K}_{k,i}\}$, where $\mathbf{K}_{k,i}$ corresponds to the covariance of $\mathbf{x_2}_{k,i}$ given the set[4] $\mathbf{X_1}_{k,i} \triangleq \{\mathbf{x_1}_{l,i}\}_{l=0}^{k}$. It is from this set of particles that the state estimates can be extracted, as seen at the bottom of Algorithm 1. The set of particles form the only variables that must be stored in memory (all the other ones can be generated from them at a given instant k). Within each iteration, each $\mathbf{x_1}_{k,i}$ corresponds to a unique $\mathbf{A}_{k,i}$, $\mathbf{B}_{k,i}$, $\mathbf{C}_{k,i}$, $\mathbf{D}_{k,i}$, $\mathbf{G}_{k,i}$, $\mathbf{H}_{k,i}$, and $\mathbf{u}_{k,i}$ (we understate here that all of these parameters are unknown). Observe that in the algorithm, as in every PF, one must resample the particles, using the weights $w_{k,i}$ that are assigned to them at each step. Resampling algorithms can be found in many sources. For example, a detailed resampling scheme is presented in [1].

Note that it would also be perfectly possible to use a regular particle filter to solve the given problem. The corresponding algorithm is given in appendix A. Given a number of particles, using a PF instead of an RBPF will lead to a faster execution, but will in general require more particles to achieve similar accuracy.

[1]The situation described does not cover the entire theoretical range of applications of the most general form of RBPF (given in Appendix B), however it does cover most of the practical range of applications, since this is a situation where the algorithm applies most conveniently.

[2]In this paper, we assume that $\mathbf{x_1}_k$ is independent of $\mathbf{x_2}_{k-1}$, conditioned upon $\mathbf{x_1}_{k-1}$

[3]The reader familiar with particle filtering will notice that we only use here the suboptimal importance density $q(\mathbf{x}_k|\mathbf{X}_{k-1,i}, \mathbf{Z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and that we apply resampling at each step.

[4]In this paper, we use the notation $a \triangleq b$ to mean "a is defined to be equal to b"

Algorithm 1 RBPF algorithm

***1.* Define $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$, choose the number of particles N**

***2.* Define and initialize the value of $\{\mathbf{x_1}_{0,i}\,;\,\mathbf{x_2}_{0,i}\,;\,\mathbf{K}_{0,i}\}_{i=1}^{N}$, where $\mathbf{K}$ is the covariance matrix of $\mathbf{x_2}$, according to any a priori belief.**

***3.* For every k, update the set $\{\mathbf{x_1}_{k-1,i}\,;\,\mathbf{x_2}_{k-1,i}\,;\,\mathbf{K}_{k-1,i}\}_{i=1}^{N}$ as follows:**

∘ For every $i \in \{1, 2, \ldots, N\}$:
- Draw $\mathbf{x_1}_{k,i} \sim p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1,i})$
- From $\mathbf{x_1}_{k,i}$, obtain the corresponding $\mathbf{A}_{k,i}$, $\mathbf{B}_{k,i}$, $\mathbf{C}_{k,i}$, $\mathbf{D}_{k,i}$, $\mathbf{G}_{k,i}$, $\mathbf{H}_{k,i}$, and $\mathbf{u}_{k,i}$ as applicable.
- Compute the following:

$$\begin{aligned}
\mathbf{K}_{k|k-1,i} &= \mathbf{G}_{k,i}\mathbf{G}_{k,i}{}^T + \mathbf{A}_{k,i}\mathbf{K}_{k-1,i}\mathbf{A}_{k,i}{}^T \\
\mathbf{T}_{k,i} &= \mathbf{H}_{k,i}\mathbf{H}_{k,i}{}^T + \mathbf{C}_{k,i}\mathbf{K}_{k|k-1,i}\mathbf{C}_{k,i}{}^T \\
\mathbf{x_2}_{k|k-1,i} &= \mathbf{A}_{k,i}\mathbf{x_2}_{k-1,i} + \mathbf{B}_{k,i}\mathbf{u}_{k,i} \\
\mathbf{y}_{k,i} &= \mathbf{C}_{k,i}\mathbf{x_2}_{k|k-1,i} + \mathbf{D}_{k,i}\mathbf{u}_{k,i} \\
\widetilde{w}_{k,i} &= \mathcal{N}(\mathbf{z}_k|\mathbf{y}_{k,i}, \mathbf{T}_{k,i}) \\
\mathbf{J}_{k,i} &= \mathbf{K}_{k|k-1,i}\mathbf{C}_{k,i}{}^T\mathbf{T}_{k,i}^{-1} \\
\mathbf{x_2}_{k,i} &= \mathbf{x_2}_{k|k-1,i} + \mathbf{J}_{k,i}(\mathbf{z}_k - \mathbf{y}_{k,i}) \\
\mathbf{K}_{k,i} &= (\mathbf{I} - \mathbf{J}_{k,i}\mathbf{C}_{k,i})\mathbf{K}_{k|k-1,i}
\end{aligned}$$

∘ Compute the normalizing factor $\sum_{i=1}^{N} \widetilde{w}_{k,i}$

∘ Normalize the weights (obtain $\{w_{k,i}\}_{i=1}^{N}$)

∘ Using the weights, resample the set $\{\mathbf{x_1}_{k,i}\,;\,\mathbf{x_2}_{k,i}\,;\,\mathbf{K}_{k,i}\}_{i=1}^{N}$

∘ Obtain the state estimates:
- $\hat{\mathbf{x}}_{\mathbf{1}k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_1}_{k,i}$
- $\hat{\mathbf{x}}_{\mathbf{2}k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_2}_{k,i}$

3 A First Example Of Application

3.1 Problem Formulation

In the first example, a signal x_k is passed through an unknown, time-varying FIR filter. We are given a set of noisy measurements at the output of the filter. The evolution of the signal in time is described by a known function (which is not restricted to be linear).

Let $\mathbf{x_2}_k \in \mathbb{R}^p$ represent the p coefficients of the FIR filter. We define $\mathbf{x_1}_k \in \mathbb{R}^p$ as the last p values of the signal to be estimated, from time $k-p+1$ to time k. This way[5], $\mathbf{x_1}_k(1) \equiv x_k$ and $\mathbf{x_1}_k(l) \equiv x_{k-l+1}$ (for $l \leq p$). In addition, let $g(.)$ be a function from $\mathbb{R}^p$ to $\mathbb{R}^p$ defined as follows:

$$g(\mathbf{x_1}_k) \triangleq [f(\mathbf{x_1}_k) \;\; \mathbf{x_1}_k(1) \;\; \mathbf{x_1}_k(2) \;\; \ldots \;\; \mathbf{x_1}_k(p-1)]^T \quad (5)$$

In equation (5), $f(.)$ represents a function from $\mathbb{R}^p$ to $\mathbb{R}$, which defines the evolution of the signal $x_k \equiv \mathbf{x_1}_k(1)$. Finally, we let $\mathbf{F}_k$ be a transition matrix for the FIR coefficients, and define the following noises: $\mathbf{w_1}_k$, $\mathbf{w_2}_k$ and $\mathbf{v}_k$ are all $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we choose to constrain $\mathbf{S} = \operatorname{diag}\{\sigma_\mathbf{1}, \mathbf{0}_{p-1\times p-1}\}$, such that only the first component of $\mathbf{x_1}_k$ is stochastic. The problem setting can be summarized by the following equations:

$$\begin{aligned}
\mathbf{x_1}_k &= g(\mathbf{x_1}_{k-1}) + \mathbf{S}\mathbf{w_1}_k && (6) \\
\mathbf{x_2}_k &= \mathbf{F}_k\mathbf{x_2}_{k-1} + \mathbf{G}\mathbf{w_2}_k && (7) \\
\mathbf{z}_k &= \mathbf{x_1}_k{}^T\mathbf{x_2}_k + \mathbf{H}\mathbf{v}_k && (8)
\end{aligned}$$

It is readily seen that conditioned upon the values of the substate $\mathbf{x_1}_k$, the system formed by equations (7) and (8) is

[5]By $a \equiv b$, we mean "a is identical to b"

linear-Gaussian. We can therefore solve this problem using RBPFs.

3.2 Application Of The Algorithm

We can directly use Algorithm 1, with $\forall\{k,i\}, \mathbf{A}_{k,i} = \mathbf{F}_k$, $\mathbf{B}_{k,i} = \mathbf{0}$, $\mathbf{C}_{k,i} = {\mathbf{x_1}_{k,i}}^T$, $\mathbf{D}_{k,i} = \mathbf{0}$, $\mathbf{u}_{k,i} = \mathbf{0}$, and $\mathbf{H}_{k,i} = \mathbf{H}$. In addition, we have $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1}) = \mathcal{N}(\mathbf{x_1}_k|g(\mathbf{x_1}_{k-1}); \mathbf{SS}^T)$. The RBPF algorithm tailored to the current example is shown in Algorithm 2.

Algorithm 2 First example of tailored RBPF algorithm

∘ For every $i \in 1, 2, .., N$

- Draw $\mathbf{x_1}_{k,i} \sim \mathcal{N}(\mathbf{x_1}_k|g(\mathbf{x_1}_{k-1,i}); \mathbf{SS}^T)$
- Compute the following:

$$
\begin{aligned}
\mathbf{K}_{k|k-1,i} &= \mathbf{GG}^T + \mathbf{F}_k\mathbf{K}_{k-1,i}\mathbf{F}_k^T \\
\mathbf{T}_{k,i} &= \mathbf{HH}^T + {\mathbf{x_1}_{k,i}}^T\mathbf{K}_{k|k-1,i}\mathbf{x_1}_{k,i} \\
\mathbf{x_2}_{k|k-1,i} &= \mathbf{F}_k\mathbf{x_2}_{k-1,i} \\
\mathbf{y}_{k,i} &= {\mathbf{x_1}_{k,i}}^T\mathbf{x_2}_{k|k-1,i} \\
\widetilde{w}_{k,i} &= \mathcal{N}(\mathbf{z}_k|\mathbf{y}_{k,i}, \mathbf{T}_{k,i}) \\
\mathbf{J}_{k,i} &= \mathbf{K}_{k|k-1,i}\mathbf{x_1}_{k,i}\mathbf{T}_{k,i}^{-1} \\
\mathbf{x_2}_{k,i} &= \mathbf{x_2}_{k|k-1,i} + \mathbf{J}_{k,i}(\mathbf{z}_k - \mathbf{y}_{k,i}) \\
\mathbf{K}_{k,i} &= (\mathbf{I} - \mathbf{J}_{k,i}{\mathbf{x_1}_{k,i}}^T)\mathbf{K}_{k|k-1,i}
\end{aligned}
$$

∘ Compute the normalizing factor $\sum_{i=1}^N \widetilde{w}_{k,i}$, and normalize the weights

∘ Resample the set $\{\mathbf{x_1}_{k,i}, \mathbf{x_2}_{k,i}, \mathbf{K}_{k,i}\}_{i=1}^N$, and compute the state estimates

3.3 Simulation Results

We present results here for a very simple case, in which a signal is passed through a time-varying gain. We thus have $p = 1$ and $g(.) \equiv f(.)$. We choose $\mathbf{F}_k = \mathbf{I}$, and $f(\mathbf{x_1}_{k-1}) \triangleq \cos(\mathbf{x_1}_{k-1}) + \sin(\mathbf{x_1}_{k-1})$. In this simple case, $\mathbf{w_1}_k$ and $\mathbf{w_2}_k$ are $\sim \mathcal{N}(0, 1)$, and we let $\mathbf{SS}^T = 0.09$ and $\mathbf{GG}^T = 0.04$. We also set $\mathbf{HH}^T = 10^{-3}$. Using $N = 500$, the results of a random simulation run is shown in Figure 1.

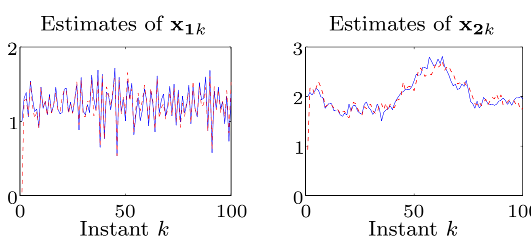

Figure 1: Simulation results for the first example
The blue lines are the true values and the red, dotted lines are the estimates.

The estimates appear to follow the true state convincingly.

4 A Second Example Of Application

4.1 Problem Formulation

We are interested here in the application of an RBPF algorithm to the problem of speech denoising. Such a procedure was introduced and applied with success by several researchers (e.g. [3, 7, 9]), and we refer the reader to their work for a thorough problem formulation, detailed results, and extensions to smoothing (more details below). We are here only focusing on the derivation of the base algorithm itself from Algorithm 1.

The model chosen for the speech signal, denoted here $x_{\mathbf{2}k}$, is an auto-regression of order M:

$$
\begin{aligned}
\mathbf{x_2}_k &= \mathbf{A}_k\mathbf{x_2}_{k-1} + \mathbf{G}_k\mathbf{w}_k && (9) \\
z_{\mathbf{2}k} &= \mathbf{C}\mathbf{x_2}_k + \sigma_{v,k}v_k && (10)
\end{aligned}
$$

where:

$$
\left\{
\begin{array}{ll}
\mathbf{x_2}_k & \triangleq [x_{\mathbf{2}k} \;\; x_{\mathbf{2}k-1} \;\; \ldots \;\; x_{\mathbf{2}k-M+1}]^T \\
\mathbf{a}_k & \triangleq [a_{1,k} \;\; a_{2,k} \;\; \ldots \;\; a_{M,k}]^T \\
\mathbf{A}_k & = \begin{bmatrix} \mathbf{a}_k^T & \\ \mathbf{I}_{M-1} & \mathbf{0}_{M-1\times 1} \end{bmatrix} \qquad \mathbf{C} = [1 \;\; \mathbf{0}_{1\times M-1}] \\
\mathbf{w}_k & \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \text{ and } \mathbf{G}_k = \operatorname{diag}\{\sigma_{w,k}, \mathbf{0}_{M-1\times M-1}\} \\
v_k & \sim \mathcal{N}(0, 1)
\end{array}
\right.
$$

As indicated in section 2, the set of parameters that form $\mathbf{x_1}_k$ are the variables which make the system (9),(10) time-varying: $\mathbf{x_1}_k = [\mathbf{a}_k \,; \sigma_{w,k} \,; \sigma_{v,k}]$. It is also clear that conditioned on $\mathbf{X_1}_k$, the problem is linear-Gaussian. The density $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$ must now be defined, to describe the evolution of the elements of $\mathbf{x_1}_k$. There are different ways to do so. First, we can reasonably state that $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1}) = p(\mathbf{a}_k|\mathbf{a}_{k-1})p(\sigma_{w,k}|\sigma_{w,k-1})p(\sigma_{v,k}|\sigma_{v,k-1})$. But as mentioned before, the elements of $\mathbf{x_1}_k$ can also be defined differently, as long as it is via a one-to-one correspondence with $\mathbf{a}_k$, $\sigma_{w,k}$, and $\sigma_{v,k}$. For example, it was found in [3] that a constrained Gaussian random-walk on the partial correlation coefficients, rather than the AR coefficients $\mathbf{a}_k$, yields better results. The one-to-one relation is then defined by the Levinson-Durbin algorithm. Similarly, the evolution of the noise variances may be defined on their logarithm, in order to ensure their positiveness [3, 7, 9]. Once $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$ is defined, we can start deriving the RBPF algorithm.

The sources cited employ a basis RBPF algorithm which corresponds to the one presented here. In [9], a constrained random-walk on the AR coefficients is employed, with additional smoothing strategies including a detailed MCMC step implementation. In [3], a complete RBP smoother is derived, based on partial correlation coefficients, and the resulting algorithm is shown to outperform several other algorithms, including a regular particle smoother. In [7], an extension of the model to α-stable observation noises is presented, with an additional parameter included in $\mathbf{x_1}_k$. For a presentation of efficient smoothing methods, which are independent on the type of filtering used, we cite [5] (in which simulation results of speech enhancement are shown for a regular PF implementation).

4.2 Application Of The Algorithm

In Algorithm 3, we present the RBPF algorithm. The algorithm is directly obtained from Algorithm 1 with $\mathbf{B}$, $\mathbf{D}$, and $\mathbf{u}$ all being equated to 0, and with $\forall\{k,i\}, \mathbf{C}_{k,i} = \mathbf{C}$.

Algorithm 3 Second example of tailored RBPF algorithm

∘ For every $i \in 1, 2, .., N$

• Draw $\mathbf{x}_{k,i}$ as:

$$\begin{aligned}\mathbf{a}_{k,i} &\sim p(\mathbf{a}_k|\mathbf{a}_{k-1,i}) \\ \sigma_{w,k,i} &\sim p(\sigma_{w,k}|\sigma_{w,k-1,i}) \\ \sigma_{v,k,i} &\sim p(\sigma_{v,k}|\sigma_{v,k-1,i})\end{aligned}$$

• Form the matrices $\mathbf{A}_{k,i}$ and $\mathbf{G}_{k,i}$

• Compute the following:

$$\begin{aligned}\mathbf{K}_{k|k-1,i} &= \mathbf{G}_{k,i}{\mathbf{G}_{k,i}}^T + \mathbf{A}_{k,i}\mathbf{K}_{k-1,i}{\mathbf{A}_{k,i}}^T \\ \mathbf{T}_{k,i} &= {\sigma_{v,k,i}}^2 + \mathbf{C}\mathbf{K}_{k|k-1,i}\mathbf{C}^T \\ \mathbf{x_2}_{k|k-1,i} &= \mathbf{A}_{k,i}\mathbf{x_2}_{k-1,i} \\ \mathbf{y}_{k,i} &= \mathbf{C}\mathbf{x_2}_{k|k-1,i} \\ \widetilde{w}_{k,i} &= \mathcal{N}(\mathbf{z}_k|\mathbf{y}_{k,i}, \mathbf{T}_{k,i}) \\ \mathbf{J}_{k,i} &= \mathbf{K}_{k|k-1,i}\mathbf{C}^T\mathbf{T}_{k,i}^{-1} \\ \mathbf{x_2}_{k,i} &= \mathbf{x_2}_{k|k-1,i} + \mathbf{J}_{k,i}(\mathbf{z}_k - \mathbf{y}_{k,i}) \\ \mathbf{K}_{k,i} &= (\mathbf{I} - \mathbf{J}_{k,i}\mathbf{C})\mathbf{K}_{k|k-1,i}\end{aligned}$$

∘ Compute the normalizing factor $\sum_{i=1}^{N} \widetilde{w}_{k,i}$, and normalize the weights

∘ Resample the set $\{\mathbf{x_1}_{k,i}, \mathbf{x_2}_{k,i}, \mathbf{K}_{k,i}\}_{i=1}^{N}$, and compute the state estimates

4.3 Simulation Results

In the second example, we show in Figure 2 a portion of a section of speech corrupted by white noise (input SNR at 2.20 dB), and the estimated clean speech (output SNR at 9.62 dB). Even though the method is quite heavier than spectral subtraction or other classical KF algorithms (see for example [8]), a comparison of the average segmental SNR is found to be favorable to RBPF-based speech enhancement. Note also that the resulting enhanced speech is not corrupted by the "musical noise" typically introduced by spectral subtraction. Again, more information and results can be found in the references of this paper (see [3, 5, 7, 9]).

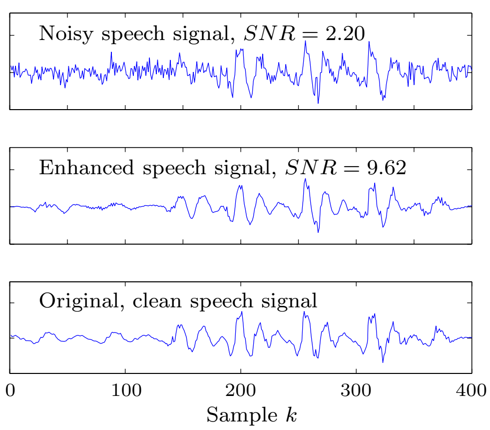

Figure 2: Simulation results for the second example

5 Conclusion

In this paper, we presented simple guidelines to derive basic RBPF algorithms. The advantage of such algorithms is that they provide a simple, holistic solution to a wide range of problems. In addition, problem modelizations are often simplified since, once it is seen that one part of the state is (conditionally) linear-Gaussian, then the remaining task is reduced to describing the evolution in time of the rest of the state. The main downside of the use of RBPFs lies in the fact that they are computationally demanding.

References

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *"A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, February 2002.

[2] A. Doucet, J.F.G. Freitas, and N. Gordon, *Sequential Monte Carlo Methods In Practice.* New York: Springer-Verlag, 2001.

[3] W. Fong, S.J. Godsill, A. Doucet, and M. West, *"Monte Carlo smoothing with application to audio signal enhancement," IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 438–49, February 2002.

[4] J.F.G. Freitas, *"Rao-Blackwellised particle filtering for fault diagnosis," Proceedings of the IEEE Aerospace Conference*, no. 4, pp. 1767–1772, March 2002.

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

[6] R.E. Kalman, *"A new approach to linear filtering and prediction problems," Transactions of the ASME–Journal of Basic Engineering*, vol. 82, Series D, pp. 35–45, 1960.

[7] M. Lombardi, *Simulation-based Estimation Methods for α-Stable Distributions and Processes.* Ph.D. thesis, Universita Degli Studi Di Firenze, 2004.

[8] K.K. Paliwal, and A. Basu, *"A speech enhancement method based on Kalman filtering," Proceedings of the ICASSP'87*, pp. 177–180, 1987.

[9] J. Vermaak, C. Andrieu, A. Doucet, and S.J. Godsill, *"Particle methods for Bayesian modeling and enhancement of speech signals," IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 173–85, March 2002.

Appendix A: A PF Solution

We present in Algorithm 4 a PF based algorithm to solve the problem of equations (3) and (4).

Appendix B: Derivation Of Algorithm 1

In this appendix, we give a possible method of obtaining Algorithm 1 from a RBPF presented in a more "standard", or general form. We begin by presenting this standard form in Algorithm 5.

First, a so-called *importance distribution*, denoted $q(\mathbf{x}_k|\mathbf{X}_{k-1,i}, \mathbf{Z}_k)$, must be chosen, and it is required to be easy to evaluate and to draw samples from. The notation $q(\mathbf{x}_k|\mathbf{X}_{k-1,i}, \mathbf{Z}_k)$ is generic but nevertheless understates that the distribution is conditional upon the set of

Algorithm 4 PF algorithm

***1.* Define $p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1})$, choose the number of particles N**
***2.* Define and initialize the value of $\{\mathbf{x}_{10,i}\,;\,\mathbf{x}_{20,i}\}_{i=1}^{N}$.**
***3.* For every k, update the set $\{\mathbf{x_1}_{k-1,i}\,;\,\mathbf{x_2}_{k-1,i}\}_{i=1}^{N}$ as follows:**

◦ For every $i \in \{1, 2, \ldots, N\}$:
- Draw $\mathbf{x_1}_{k,i} \sim p(\mathbf{x_1}_k|\mathbf{x_1}_{k-1,i})$
- From $\mathbf{x_1}_{k,i}$, obtain the corresponding $\mathbf{A}_{k,i}$, $\mathbf{B}_{k,i}$, $\mathbf{C}_{k,i}$, $\mathbf{D}_{k,i}$, $\mathbf{G}_{k,i}$, $\mathbf{H}_{k,i}$, and $\mathbf{u}_{k,i}$ as applicable.
- Draw $\mathbf{x_2}_{k,i} \sim \mathcal{N}(\mathbf{x_2}_k|\mathbf{A}_{k,i}\mathbf{x_2}_{k-1,i} + \mathbf{B}_{k,i}\mathbf{u}_{k,i}; \mathbf{G}_{k,i}{\mathbf{G}_{k,i}}^T)$
- Compute $\widetilde{w}_{k,i} = \mathcal{N}(\mathbf{z}_k|\mathbf{C}_{k,i}\mathbf{x_2}_{k,i} + \mathbf{D}_{k,i}\mathbf{u}_{k,i}; \mathbf{H}_{k,i}{\mathbf{H}_{k,i}}^T)$

◦ Compute the normalizing factor $\sum_i \widetilde{w}_{k,i}$ and normalize the weights
◦ Using the weights, resample the set $\{\mathbf{x_1}_{k,i}\,;\,\mathbf{x_2}_{k,i}\}_{i=1}^{N}$
◦ Obtain $\hat{\mathbf{x}}_{\mathbf{1}k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_1}_{k,i}$ and $\hat{\mathbf{x}}_{\mathbf{2}k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_2}_{k,i}$

Algorithm 5 Standard RBPF algorithm

For every k, do the following:
◦ For every $i \in \{1, 2, \ldots, N\}$:
- Draw $\mathbf{x_1}_{k,i} \sim q(\mathbf{x_1}_k|\mathbf{X_1}_{k-1,i}, \mathbf{Z}_k)$
- Set $\mathbf{X_1}_{k,i} = \{\mathbf{x_1}_{k,i}\,;\,\mathbf{X_1}_{k-1,i}\}$
- Compute the unnormalized weights: $\widetilde{w}_{k,i} = w_{k-1,i}\frac{p(\mathbf{z}_k|\mathbf{X_1}_{k,i},\mathbf{Z}_{k-1})p(\mathbf{x_1}_{k,i}|\mathbf{X_1}_{k-1,i},\mathbf{Z}_{k-1})}{q(\mathbf{x_1}_{k,i}|\mathbf{X_1}_{k-1,i},\mathbf{Z}_k)}$

◦ Compute the normalizing factor $\sum_{i=1}^{N} \widetilde{w}_{k,i}$
◦ Normalize the weights and resample the particles, if necessary
◦ For every $i \in \{1, 2, \ldots, N\}$:
- Update $p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_k)$ using $p(\mathbf{x_2}_{k-1}|\mathbf{X_1}_{k-1,i}, \mathbf{Z}_{k-1})$, $\mathbf{x_1}_{k,i}$, $\mathbf{x_1}_{k-1,i}$, and $\mathbf{z}_k$ (*Exact step*).

previous states and the set of all measurements. The performance and execution of the algorithm depends on the choice of $q(.)$. Optimal densities theoretically uniquely exist, however they are often intractable [1, 2]. In the examples of this paper, we only use one of the most common and simple choice, $q(\mathbf{x}_k|\mathbf{X}_{k-1,i}, \mathbf{Z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Assuming that $\mathbf{x_1}_k$ is independent of $\mathbf{x_2}_{k-1}$, conditioned upon $\mathbf{x_1}_{k-1}$, such a choice reduces the weight update equation to:

$$\widetilde{w}_{k,i} = w_{k-1,i}p(\mathbf{z}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1}) \tag{11}$$

Thus, there remains to determine $p(\mathbf{z}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1})$ in the context of this paper. We can show that:

$$p(\mathbf{z}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x_2}_k, \mathbf{x_1}_{k,i})p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1})d\mathbf{x_2}_k \tag{12}$$

Observe from equation (4) that:

$$p(\mathbf{z}_k|\mathbf{x_2}_k, \mathbf{x_1}_{k,i}) = \mathcal{N}(\mathbf{z}_k|\mathbf{C}_{k,i}\mathbf{x_2}_k + \mathbf{D}_{k,i}\mathbf{u}_{k,i}; \mathbf{H}_{k,i}{\mathbf{H}_{k,i}}^T) \tag{13}$$

The distribution $p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1})$ can be computed as:

$$p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1}) = \int p(\mathbf{x_2}_k|\mathbf{x_2}_{k-1}, \mathbf{x_1}_{k,i})p(\mathbf{x_2}_{k-1}|\mathbf{X_1}_{k-1,i}, \mathbf{Z}_{k-1})d\mathbf{x_2}_{k-1} \tag{14}$$

At this point, note from equation (3) that:

$$p(\mathbf{x_2}_k|\mathbf{x_2}_{k-1}, \mathbf{x_1}_{k,i}) = \mathcal{N}(\mathbf{x_2}_k|\mathbf{A}_{k,i}\mathbf{x_2}_{k-1} + \mathbf{B}_{k,i}\mathbf{u}_{k,i}; \mathbf{G}_{k,i}{\mathbf{G}_{k,i}}^T) \tag{15}$$

According to the algorithm, at instant k we are given the distribution $p(\mathbf{x_2}_{k-1}|\mathbf{X_1}_{k-1,i}, \mathbf{Z}_{k-1})$ – it is precisely the one that we are updating online. From the Gaussianness of the system, this distribution is Gaussian (it is in fact the *a priori* distribution of the state in the KF equations). Let us define it as:

$$p(\mathbf{x_2}_{k-1}|\mathbf{X_1}_{k-1,i}, \mathbf{Z}_{k-1}) \triangleq \mathcal{N}(\mathbf{x_2}_{k-1}|\mathbf{x_2}_{k-1,i}; \mathbf{K}_{k-1,i}) \tag{16}$$

To complete the derivation, we now use the following result. If $\mathcal{N}(\mathbf{x}|\mathbf{y}; \mathbf{Q})$ represents the density of a Gaussian random vector with mean $\mathbf{y}$ and covariance matrix $\mathbf{Q}$, then:

$$\int \mathcal{N}(\mathbf{x}|\mathbf{F}\mathbf{y}; \mathbf{Q})\mathcal{N}(\mathbf{y}|\mathbf{z}; \mathbf{K})\mathrm{d}\mathbf{y} = \mathcal{N}(\mathbf{x}|\mathbf{n}; \mathbf{N}) \tag{17}$$

where $\mathbf{N} = \mathbf{Q} + \mathbf{F}\mathbf{K}\mathbf{F}^T$ and $\mathbf{n} = \mathbf{F}\mathbf{z}$.

We can identify equation equation (17) to equation (14), with the integrand terms given by (15) and (16), and we obtain:

$$p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1}) = \mathcal{N}(\mathbf{x_2}_k|\mathbf{x_2}_{k|k-1,i}; \mathbf{K}_{k|k-1,i}) \tag{18}$$

where:

$$\begin{aligned} \mathbf{x_2}_{k|k-1,i} &\triangleq \mathbf{A}_{k,i}\mathbf{x_2}_{k-1,i} + \mathbf{B}_{k,i}\mathbf{u}_{k,i} \\ \mathbf{K}_{k|k-1,i} &\triangleq \mathbf{G}_{k,i}{\mathbf{G}_{k,i}}^T + \mathbf{A}_{k,i}\mathbf{K}_{k-1,i}{\mathbf{A}_{k,i}}^T \end{aligned}$$

Again, we use (17) but this time applied to (12), using (13) and (18), and we obtain:

$$p(\mathbf{z}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1}) = \mathcal{N}(\mathbf{z}_k|\mathbf{y}_{k,i}; \mathbf{T}_{k,i}) \tag{19}$$

where:

$$\begin{aligned} \mathbf{y}_{k,i} &\triangleq \mathbf{C}_{k,i}\mathbf{x_2}_{k|k-1,i} + \mathbf{D}_{k,i}\mathbf{u}_{k,i} \\ \mathbf{T}_{k,i} &\triangleq \mathbf{H}_{k,i}{\mathbf{H}_{k,i}}^T + \mathbf{C}_{k,i}\mathbf{K}_{k|k-1,i}{\mathbf{C}_{k,i}}^T \end{aligned}$$

In the process of determining $p(\mathbf{z}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_{k-1})$, we have taken the same road as the classical Kalman filter equations (see [6]). Half of the exact step has thus been completed, and we only need to find $p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_k)$. To do so, referring to [6], we can write:

$$p(\mathbf{x_2}_k|\mathbf{X_1}_{k,i}, \mathbf{Z}_k) = \mathcal{N}(\mathbf{x_2}_k|\mathbf{x_2}_{k,i}; \mathbf{K}_{k,i}) \tag{20}$$

where, using $\mathbf{J}_{k,i} = \mathbf{K}_{k|k-1,i}{\mathbf{C}_{k,i}}^T\mathbf{T}_{k,i}^{-1}$, we have:

$$\begin{aligned} \mathbf{x_2}_{k,i} &= \mathbf{x_2}_{k|k-1,i} + \mathbf{J}_{k,i}(\mathbf{z}_k - \mathbf{y}_{k,i}) \\ \mathbf{K}_{k,i} &= (\mathbf{I} - \mathbf{J}_{k,i}\mathbf{C}_{k,i})\mathbf{K}_{k|k-1,i} \end{aligned}$$

The KF equations and the weight computations are thus intertwined, and we can combine the two loops on i of Algorithm 5 into a single loop. Finally, if resampling is applied at each step, then there is no need, in equation (11), to multiply by $w_{k-1,i}$ since it will be equal to $1/N$ for all $\{k, i\}$.