

Fast Filtering in Switching Approximations of Nonlinear Markov Systems With Applications to Stochastic Volatility

Ivan Gorynin, Stéphane Derrode, *Senior Member, IEEE*, Emmanuel Monfrini, and Wojciech Pieczynski

Abstract—We consider the problem of optimal statistical filtering in general nonlinear non-Gaussian Markov dynamic systems. The novelty of the proposed approach consists in approximating the nonlinear system by a recent Markov switching process, in which one can perform exact and optimal filtering with a linear time complexity. All we need to assume is that the system is stationary (or asymptotically stationary), and that one can sample its realizations. We evaluate our method using two stochastic volatility models and results show its efficiency.

Index Terms—Conditionally Gaussian linear state-space model, filtering in switching systems, Kalman filter, nonlinear systems, optimal statistical filter, stochastic volatility model.

I. INTRODUCTION

LET US consider two random sequences $\mathbf{X}_1^N = (\mathbf{X}_1, \dots, \mathbf{X}_N)$ and $\mathbf{Y}_1^N = (\mathbf{Y}_1, \dots, \mathbf{Y}_N)$, taking their values in \mathbb{R}^m and \mathbb{R}^q , respectively. \mathbf{X}_1^N is hidden, while \mathbf{Y}_1^N is observed. In this paper, we focus on the optimal filtering problem, which consists in the sequential search of \mathbf{X}_1^N from \mathbf{Y}_1^N . More precisely, we present a nonstochastic iterative algorithm which computes the expectation of \mathbf{X}_n conditional on \mathbf{Y}_1^n .

We present a workable approach for filtering in general stationary (or asymptotically stationary) Markov dynamic systems, provided that one can sample a realization of the systems. This approach makes use of a recent switching model, in which fast exact optimal filtering is computationally feasible.

The idea is to approximate a given nonlinear non-Gaussian system by a switching Gaussian system. Such ideas are not new, as it is well known that a wide range of probability distributions can be approximated by a Gaussian mixture. In the standard switching models, e.g., in jump Markov linear systems (JMLSs), there is no known fast exact optimal filtering algorithm [1]–[4]; therefore, such approximations are useless in their context. However, there are some recent switching models, in which fast exact optimal filtering is computationally feasible, e.g., the con-

ditionally Markov switching hidden linear model (CMSHLM [5]) and the conditionally Gaussian observed Markov switching model (CGOMSM [6], [7]). The novelty of the paper is to use them as an approximation of some given model, then to apply the corresponding filtering algorithms. Besides, let us notice that any standard JMLS can be approximated by a CMSHLM [8], [9].

To be more precise, CMSHLMs are Markov triplet $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ models, where \mathbf{R}_1^N is a chain of switches, and where there exists an exact fast filtering algorithm which is as fast as the standard Kalman filter. The main difference between CMSHLMs and JMLSs is that in CMSHLMs $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markovian and $(\mathbf{X}_1^N, \mathbf{R}_1^N)$ is not necessarily Markovian, contrary to JMLSs where $(\mathbf{X}_1^N, \mathbf{R}_1^N)$ is Markovian and $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is not necessarily Markovian.

The CGOMSM is a sub-model of CMSHLM, in which $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is Gaussian conditional on \mathbf{R}_1^N . As the main novelty, deriving from its prior simplified version introduced in [10], we put forward a method for approximating any stationary nonlinear non-Gaussian Markov model by a CGOMSM, in order to perform fast filtering in this approximation. The other main novelty compared with [10] is the design of a new expectation-maximization (EM)-based approximation algorithm, as well as several new experiments. In particular, we present consistent results that we obtained for a recent asymmetric stochastic volatility model.

Our method can be used as an alternative to the particle filter (PF) based methods, which are widely used in different areas like finance [11]–[15] and tracking [2]–[4], [16]. Although PFs are asymptotically optimal, their use may be problematic due to the particle degeneracy phenomenon, or to the need of a large amount of particles when the dimension of the hidden space is high. The method proposed is fundamentally different and is free of these impediments. Besides, our filtering algorithm is exact and is as fast as the standard Kalman filter.

Let us mention some other nonlinear non-Gaussian filters which similarly use a mixture model to represent the filtering pdf. Such examples include the Gaussian sum filter (GSF) [17], [18], the unscented Kalman filter (UKF) [19], the unscented Gaussian sum filter (UGSF) [20], and the Gaussian sum unscented Kalman filter (GSUKF) with adaptive scaling parameters [21]. The main difference between these methods and ours is that they rely on a supplementary approximation to prevent the number of mixands to growth exponentially with time (this number remains constant in our representation). We show through some experiments that the GSF may be disadvantaged due to this additional approximation, whereas our

Manuscript received November 4, 2015; revised March 26, 2016 and March 31, 2016; accepted May 4, 2016. Date of publication May 17, 2016; date of current version January 26, 2017. Recommended by Associate Editor H. S. Chang.

I. Gorynin, E. Monfrini, and W. Pieczynski are with SAMOVAR, Telecom Sudparis, CNRS, Université Paris-Saclay, 91000 Evry, France (e-mail: ivan.gorynin@telecom-sudparis.eu).

S. Derrode is with Ecole Centrale de Lyon, LIRIS, CNRS UMR 5205, 69130 Écully, France (e-mail: stephane.derrode@ec-lyon.fr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2016.2569417

method attains the optimal accuracy. Moreover, contrary to PF and GSUKF, our filter is designed for stationary (or asymptotically stationary) systems only. However, let us notice that the context of stationary systems is relevant for many applications, and different efficient approaches have been recently proposed [22], [23].

We provide two series of experiments. The first one is devoted to filtering for a standard stochastic volatility model (SV) [24]–[27]. The second is dedicated to filtering for the asymmetric stochastic volatility model (ASV) [28]–[30], which extends the SV model by incorporating a leverage effect. We conclude from these experiments that once the approximation is established, our method has the same performance in terms of the mean squared error that the particle filter, while being significantly faster.

The rest of the paper is organized as follows. In Section II, we describe and justify our approximation setting. In Section III, we recall some crucial properties of the CGOMSM, and then explain our approximation algorithm in Section IV. Section V contains experiments and Section VI gives some concluding remarks and prospects for further works.

II. SWITCHING MODEL APPROXIMATION

Let us consider two random sequences \mathbf{X}_1^N and \mathbf{Y}_1^N as previously described. The common model for $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is the hidden Markov model (HMM), where \mathbf{X}_1^N is Markovian, $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ are independent conditional on \mathbf{X}_1^N , and the distribution of each \mathbf{Y}_n depends only on \mathbf{X}_n given \mathbf{X}_1^N . We can define the joint pdf $p(\mathbf{x}_1^N, \mathbf{y}_1^N)$ via $p(\mathbf{x}_1, \mathbf{y}_1)$ and two recursions as follows:

$$\mathbf{X}_{n+1} = \mathcal{F}(\mathbf{X}_n, \mathbf{U}_n) \quad (1a)$$

$$\mathbf{Y}_n = \mathcal{G}(\mathbf{X}_n, \mathbf{V}_n) \quad (1b)$$

where \mathcal{F}, \mathcal{G} are appropriate mappings representing the Markov kernel and the observation kernel respectively, and $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$ are appropriate independent variables. Equivalently, any HMM is given by $p(\mathbf{x}_1, \mathbf{y}_1)$ and conditional densities $p(\mathbf{x}_{n+1}|\mathbf{x}_n), p(\mathbf{y}_n|\mathbf{x}_n)$ for each n in $\{1, \dots, N-1\}$ (in $\{2, \dots, N\}$, respectively). We recall that if, in addition, $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is Gaussian, then we deal with a Gaussian linear model in which the Kalman filter is statistically optimal.

The HMM can be extended to the “pairwise Markov model” (PMM), where the recursion

$$(\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}) = \mathcal{H}(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{W}_n) \quad (2)$$

replaces (1), \mathcal{H} being any mapping, and $\mathbf{W}_1, \dots, \mathbf{W}_N$ being independent variables. Likewise to the HMM, any PMM is given by $p(\mathbf{x}_1, \mathbf{y}_1)$ and the conditional pdf $p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n)$. We note that

$$p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) = p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) p(\mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}) \quad (3)$$

and thus (1) is a particular case of (2) where $p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) = p(\mathbf{x}_{n+1}|\mathbf{x}_n)$ and $p(\mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}) = p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1})$.

In our setting, we suppose that \mathbf{X}_1^N and \mathbf{Y}_1^N are stationary signals, which means that the distributions $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ do not depend on n , i.e., for any n , $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ is equal in distribution to $(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2)$, what we note by $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1}) = p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$. However, in practice, the algorithm that we put forward applies to asymptotically stationary signals as well, which is detailed in Remark II.1.

The idea proposed in [10] is to approximate $p(\mathbf{x}_1^N, \mathbf{y}_1^N)$ with a CGOMSM marginal distribution. That is to perform an exact fast filtering afterwards. More precisely, since the model (2) is stationary, its distribution derives from $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$, as it provides $p(\mathbf{x}_1, \mathbf{y}_1)$ and $p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) = p(\mathbf{x}_2, \mathbf{y}_2|\mathbf{x}_1, \mathbf{y}_1)$ for each $n = 1, \dots, N-1$. Besides, $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ can be approximated using a mixture of K^2 components

$$p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) \approx \sum_{1 \leq i, j \leq K} \alpha_{ij} p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) \quad (4)$$

where $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ are Gaussian distributions which verify some further detailed hypotheses. Then, the scalars α_{ij} are seen as a discrete distribution $\alpha_{ij} = P[R_1 = i, R_2 = j]$ of a pair of random variables (R_1, R_2) taking their values in $\Omega = \{1, \dots, K\}$ and the approximation (4) may be seen as a marginal distribution of

$$p(\mathbf{x}_1, \mathbf{y}_1, r_1, \mathbf{x}_2, \mathbf{y}_2, r_2) = p(r_1, r_2) p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2 | r_1, r_2). \quad (5)$$

Then the main idea is to consider the stationary triplet Markov model $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$, with $\mathbf{R}_1^N = (R_1, \dots, R_N)$, whose distribution is defined by (5) and which would belong to the CGOMSM family. As specified in Remark II.2, such a model approximates (5) in that for any n in $\{1, \dots, N-1\}$ the distribution $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ in the CGOMSM is as close to the distribution $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ in the PMM as the distribution $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ in the CGOMSM is close to the distribution $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ in the PMM.

Let us now specify what are the properties of the Gaussian distributions $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ in (4) needed to make the corresponding approximating switching triplet model being a CGOMSM. Let us note $\mathbf{Z}_n^T = [\mathbf{X}_n^T, \mathbf{Y}_n^T]$ and assume that for each n in $\{1, \dots, N-1\}$

$$p(r_{n+1} | \mathbf{x}_n, r_n, \mathbf{y}_n) = p(r_{n+1} | r_n) \quad (6)$$

which implies the Markovianity of \mathbf{R}_1^N . This is equivalent to $p(r_2 | \mathbf{x}_1, r_1, \mathbf{y}_1) = p(r_2 | r_1)$ by the stationarity assumption. Besides, the above equation means that \mathbf{Z}_1 and R_2 are independent conditional on R_1 . Therefore, we obtain $p(\mathbf{x}_1, \mathbf{y}_1 | r_1, r_2) = p(\mathbf{x}_1, \mathbf{y}_1 | r_1)$. Since $(\mathbf{X}_1, R_1, \mathbf{Y}_1)$ and $(\mathbf{X}_2, R_2, \mathbf{Y}_2)$ are identically distributed by the stationarity assumption for \mathbf{T}_1^N , it follows that $p(\mathbf{x}_2, \mathbf{y}_2 | r_1, r_2) = p(\mathbf{x}_2, \mathbf{y}_2 | r_2)$. Thus, Gaussian distributions $p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2 | r_1, r_2)$ are given by the variance matrices $\mathbf{\Gamma}_{\mathbf{Z}_1}(r_1), \mathbf{\Gamma}_{\mathbf{Z}_2}(r_2)$ (note that the mappings $\mathbf{\Gamma}_{\mathbf{Z}_1}(\cdot)$ and $\mathbf{\Gamma}_{\mathbf{Z}_2}(\cdot)$ are equal), and the cross-covariance matrices $\mathbf{\Sigma}_{\mathbf{Z}_1 \mathbf{Z}_2}(r_1^2)$.

Let us set

$$\mathbf{\Gamma}(r_1^2) = \begin{bmatrix} \mathbf{\Gamma}_{\mathbf{Z}_1}(r_1) & \mathbf{\Sigma}_{\mathbf{Z}_1 \mathbf{Z}_2}(r_1^2) \\ \mathbf{\Sigma}_{\mathbf{Z}_1 \mathbf{Z}_2}^T(r_1^2) & \mathbf{\Gamma}_{\mathbf{Z}_2}(r_2) \end{bmatrix} \quad (7)$$

then we obtain

$$\mathbf{Z}_2 = \mathbf{a}(r_1^2) \mathbf{Z}_1 + \mathbf{b}(r_1^2) \mathbf{W}_1 + \mathbf{c}(r_1^2) \quad (8)$$

where \mathbf{W}_1 is a standard Gaussian vector, independent of \mathbf{Z}_1 , and

$$\mathbf{a}(\mathbf{r}_1^2) = \Sigma_{\mathbf{Z}_1 \mathbf{Z}_2}^\top(\mathbf{r}_1^2) \Gamma_{\mathbf{Z}_1}^{-1}(r_1) \quad (9a)$$

$$\mathbf{b}(\mathbf{r}_n^{n+1}) \mathbf{b}^\top(\mathbf{r}_n^{n+1}) = \Gamma_{\mathbf{Z}_2}(r_2) - \mathbf{a}(\mathbf{r}_1^2) \Sigma_{\mathbf{Z}_1 \mathbf{Z}_2}(\mathbf{r}_1^2) \quad (9b)$$

$$\mathbf{c}(\mathbf{r}_1^2) = \mathbb{E}[\mathbf{Z}_2 | r_2] - \mathbf{a}(\mathbf{r}_1^2) \mathbb{E}[\mathbf{Z}_1 | r_1]. \quad (9c)$$

The matrices $\Gamma_{\mathbf{Z}_1}(r_1)$, $\Sigma_{\mathbf{Z}_1 \mathbf{Z}_2}(\mathbf{r}_1^2)$, $\mathbf{a}(\mathbf{r}_1^2)$, $\mathbf{b}(\mathbf{r}_1^2)$ and the vector $\mathbf{c}(\mathbf{r}_1^2)$ may be written in the following block-form:

$$\begin{aligned} \Gamma_{\mathbf{Z}_1}(r_1) &= \begin{bmatrix} \Gamma_{\mathbf{X}_1}(r_1) & \Sigma_{\mathbf{X}_1 \mathbf{Y}_1}(r_1) \\ \Sigma_{\mathbf{Y}_1 \mathbf{X}_1}(r_1) & \Gamma_{\mathbf{Y}_1}(r_1) \end{bmatrix} \\ \Sigma_{\mathbf{Z}_1 \mathbf{Z}_2}(\mathbf{r}_1^2) &= \begin{bmatrix} \Sigma_{\mathbf{X}_1 \mathbf{X}_2}(\mathbf{r}_1^2) & \Sigma_{\mathbf{X}_1 \mathbf{Y}_2}(\mathbf{r}_1^2) \\ \Sigma_{\mathbf{Y}_1 \mathbf{X}_2}(\mathbf{r}_1^2) & \Sigma_{\mathbf{Y}_1 \mathbf{Y}_2}(\mathbf{r}_1^2) \end{bmatrix} \\ \mathbf{a}(\mathbf{r}_1^2) &= \begin{bmatrix} \mathbf{a}_1(\mathbf{r}_1^2) & \mathbf{a}_2(\mathbf{r}_1^2) \\ \mathbf{a}_3(\mathbf{r}_1^2) & \mathbf{a}_4(\mathbf{r}_1^2) \end{bmatrix} \\ \mathbf{b}(\mathbf{r}_1^2) &= \begin{bmatrix} \mathbf{b}_1(\mathbf{r}_1^2) & \mathbf{b}_2(\mathbf{r}_1^2) \\ \mathbf{b}_3(\mathbf{r}_1^2) & \mathbf{b}_4(\mathbf{r}_1^2) \end{bmatrix} \\ \mathbf{c}(\mathbf{r}_1^2) &= \begin{bmatrix} \mathbf{c}_1(\mathbf{r}_1^2) \\ \mathbf{c}_2(\mathbf{r}_1^2) \end{bmatrix}. \end{aligned}$$

Then, by stationarity, we have for $n = 1, \dots, N-1$

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_{n+1} \\ \mathbf{Y}_{n+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{a}_1(\mathbf{r}_n^{n+1}) & \mathbf{a}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{a}_3(\mathbf{r}_n^{n+1}) & \mathbf{a}_4(\mathbf{r}_n^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{b}_1(\mathbf{r}_n^{n+1}) & \mathbf{b}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{b}_3(\mathbf{r}_n^{n+1}) & \mathbf{b}_4(\mathbf{r}_n^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{U}_{n+1} \\ \mathbf{V}_{n+1} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_1(\mathbf{r}_n^{n+1}) \\ \mathbf{c}_2(\mathbf{r}_n^{n+1}) \end{bmatrix} \quad (10) \end{aligned}$$

where $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$ are independent standard Gaussian vectors.

Definition II.1: The mixture (4) will be said ‘‘Conditionally Gaussian Observed Markov Switching Model mixture’’ (CGOMSM mixture) if it verifies $p(r_2 | \mathbf{x}_1, r_1, \mathbf{y}_1) = p(r_2 | r_1)$ and if each Gaussian pdf p_{ij} satisfies one of the two following equivalent properties:

- i) $\mathbf{a}_3(\mathbf{r}_1^2) = 0$ for each $\mathbf{r}_1^2 \in \Omega^2$;
- ii) $\Sigma_{\mathbf{X}_1 \mathbf{Y}_2}(\mathbf{r}_1^2) = \Sigma_{\mathbf{X}_1 \mathbf{Y}_1}(\mathbf{r}_1^2) \Gamma_{\mathbf{Y}_1}^{-1}(r_1) \Sigma_{\mathbf{Y}_1 \mathbf{Y}_2}(\mathbf{r}_1^2)$ for each $\mathbf{r}_1^2 \in \Omega^2$.

As explained previously, the above definition also includes the distribution $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ of CGOMSM for any n in $\{1, \dots, N-1\}$. Our filtering algorithm, which runs within the framework of the CGOMSM, is explained in next section.

Definition II.2: We call ‘‘CGOMSM Approximation Based Filter’’ (CGOMSM-ABF) the following algorithm:

- i) generate a realization $\mathbf{z}_1^{M*} = (\mathbf{z}_1^*, \dots, \mathbf{z}_M^*)$ from the prior model (2);
- ii) infer the CGOMSM parameters based on the observation \mathbf{z}_1^{M*} assuming that the chain of switches \mathbf{R}_1^{N*} is missing;
- iii) use these parameters to recover the hidden signal \mathbf{x}_1^N from \mathbf{y}_1^N .

For the step ii), we propose to use an Expectation–Maximization-based method, which we detail in Section IV. However, there are different alternative approaches that may apply as well.

Remark II.1: In practice, the initial pdf $p(\mathbf{x}_1, \mathbf{y}_1)$ may be unspecified. However, if the system has the rapid mixing property,

the residual effect of the starting distribution on $p(\mathbf{x}_n, \mathbf{y}_n)$ would be negligible for moderate and large values of n [31]–[33]. Thus, in order to generate a realization \mathbf{z}_1^{M*} , it might be possible to begin with some auxiliary pdf $p'(\mathbf{x}_1, \mathbf{y}_1)$. It is also common in practice to drop the burn-in sample $\mathbf{z}_1^*, \dots, \mathbf{z}_n^*$.

Remark II.2: The approximation (4) can be obtained with an arbitrary precision. If we denote by p^* the pdf of the approximation, then $p^*(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ approximates $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ for any n in $\{1, \dots, N-1\}$ with exactly the same precision. The readers interested by further investigations of the switching model may consult [34] to see that the marginal distribution $p(\mathbf{x}_1^N, \mathbf{y}_1^N)$ of a CGOMSM is not necessarily Markovian (it is always not for the reversible switched linear models in [34]). However, in our use of CGOMSM its marginal distribution mimics closely the Markov property.

III. EXACT FAST OPTIMAL FILTERING IN CGOMSM MODELS

We recall in this section how the fast optimal filter runs in a stationary CGOMSM. Let $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ be a CGOMSM. The aim is to search $p(r_{n+1} | \mathbf{y}_1^{n+1})$, $\mathbb{E}[\mathbf{X}_{n+1} | r_{n+1}, \mathbf{y}_1^{n+1}]$ and $\mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | r_{n+1}, \mathbf{y}_1^{n+1}]$ from $p(r_n | \mathbf{y}_1^n)$, $\mathbb{E}[\mathbf{X}_n | r_n, \mathbf{y}_1^n]$, $\mathbb{E}[\mathbf{X}_n \mathbf{X}_n^\top | r_n, \mathbf{y}_1^n]$ and \mathbf{y}_{n+1} . Thus, the estimate of \mathbf{X}_{n+1} is then given by

$$\mathbb{E}[\mathbf{X}_{n+1} | \mathbf{y}_1^{n+1}] = \sum_{r_{n+1}} p(r_{n+1} | \mathbf{y}_1^{n+1}) \mathbb{E}[\mathbf{X}_{n+1} | r_{n+1}, \mathbf{y}_1^{n+1}] \quad (11)$$

and its variance is

$$\begin{aligned} \text{Var}[\mathbf{X}_{n+1} | \mathbf{y}_1^{n+1}] &= \mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | \mathbf{y}_1^{n+1}] \\ &- \mathbb{E}[\mathbf{X}_{n+1} | \mathbf{y}_1^{n+1}] \mathbb{E}[\mathbf{X}_{n+1}^\top | \mathbf{y}_1^{n+1}] \end{aligned}$$

with

$$\begin{aligned} \mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | \mathbf{y}_1^{n+1}] &= \sum_{r_{n+1}} p(r_{n+1} | \mathbf{y}_1^{n+1}) \\ &\times \mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | r_{n+1}, \mathbf{y}_1^{n+1}]. \quad (12) \end{aligned}$$

A. Some CGOMSM Properties

The main property of CGOMSM is

$$p(r_{n+1}, \mathbf{y}_{n+1} | \mathbf{x}_n, r_n, \mathbf{y}_n) = p(r_{n+1}, \mathbf{y}_{n+1} | r_n, \mathbf{y}_n) \quad (13)$$

which is straightforward from i) in Definition II.1. There are some consequences which may be drawn from the equation above and (6). First, $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ with latent \mathbf{R}_1^N is a hidden Markov chain (with *correlated noise*). Thus,

$$p(r_{n+1}, \mathbf{y}_{n+1} | r_n, \mathbf{y}_n) = p(r_{n+1} | r_n) p(\mathbf{y}_{n+1} | \mathbf{r}_n^{n+1}, \mathbf{y}_n). \quad (14)$$

Second, (13) is equivalent to

$$p(\mathbf{x}_n | \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = p(\mathbf{x}_n | r_n, \mathbf{y}_n) \quad (15)$$

which is fundamental for the conception of our fast filter. Finally, since the distribution $p(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$ is Gaussian, \mathbf{X}_{n+1} is Gaussian conditional on the pair (R_n, R_{n+1}) and on a

linear combination of \mathbf{X}_n , \mathbf{Y}_n and \mathbf{Y}_{n+1} . A similar reasoning holds for $p(\mathbf{y}_{n+1}|\mathbf{y}_n, \mathbf{r}_n^{n+1})$ and summarizing, we have

$$\mathbf{Y}_{n+1} = \mathbf{D}(\mathbf{r}_n^{n+1})\mathbf{Y}_n + \mathbf{H}(\mathbf{r}_n^{n+1}) + \mathbf{\Lambda}(\mathbf{r}_n^{n+1})\mathbf{V}_{n+1} \quad (16a)$$

$$\begin{aligned} \mathbf{X}_{n+1} = & \mathbf{A}(\mathbf{r}_n^{n+1})\mathbf{X}_n + \mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{Y}_n + \mathbf{C}(\mathbf{r}_n^{n+1})\mathbf{Y}_{n+1} \\ & + \mathbf{F}(\mathbf{r}_n^{n+1}) + \mathbf{\Pi}(\mathbf{r}_n^{n+1})\mathbf{U}_{n+1} \end{aligned} \quad (16b)$$

for some parameters $\mathbf{D}(\mathbf{r}_n^{n+1})$, $\mathbf{H}(\mathbf{r}_n^{n+1})$, $\mathbf{\Lambda}(\mathbf{r}_n^{n+1})$, $\mathbf{A}(\mathbf{r}_n^{n+1})$, $\mathbf{B}(\mathbf{r}_n^{n+1})$, $\mathbf{C}(\mathbf{r}_n^{n+1})$, $\mathbf{F}(\mathbf{r}_n^{n+1})$, $\mathbf{\Pi}(\mathbf{r}_n^{n+1})$ and standard Gaussian vectors $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$. These parameters may be derived as follows.

Let us set $M_{r_n}^{\mathbf{X}} = \mathbb{E}[\mathbf{X}_n|r_n]$ and $M_{r_n}^{\mathbf{Y}} = \mathbb{E}[\mathbf{Y}_n|r_n]$. It follows from (10) that $[\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}]^\top$ is normally distributed conditional on $[\mathbf{X}_n, \mathbf{Y}_n]^\top$ and \mathbf{R}_n^{n+1} . From (10), we find that the conditional mean of $[\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}]^\top$ is

$$\begin{aligned} & \begin{bmatrix} M_{r_{n+1}}^{\mathbf{X}} \\ M_{r_{n+1}}^{\mathbf{Y}} \end{bmatrix} + \begin{bmatrix} \mathbf{a}_1(\mathbf{r}_n^{n+1}) & \mathbf{a}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{a}_3(\mathbf{r}_n^{n+1}) & \mathbf{a}_4(\mathbf{r}_n^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{x}_n - M_{r_n}^{\mathbf{X}} \\ \mathbf{y}_n - M_{r_n}^{\mathbf{Y}} \end{bmatrix} \\ & = \begin{bmatrix} M_{r_{n+1}}^{\mathbf{X}} + \mathbf{a}_1(\mathbf{r}_n^{n+1})(\mathbf{x}_n - M_{r_n}^{\mathbf{X}}) + \mathbf{a}_2(\mathbf{r}_n^{n+1})(\mathbf{y}_n - M_{r_n}^{\mathbf{Y}}) \\ M_{r_{n+1}}^{\mathbf{Y}} + \mathbf{a}_3(\mathbf{r}_n^{n+1})(\mathbf{x}_n - M_{r_n}^{\mathbf{X}}) + \mathbf{a}_4(\mathbf{r}_n^{n+1})(\mathbf{y}_n - M_{r_n}^{\mathbf{Y}}) \end{bmatrix} \end{aligned} \quad (17)$$

and that the conditional variance matrix of $[\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}]^\top$ is $\mathbf{b}(\mathbf{r}_n^{n+1})\mathbf{b}^\top(\mathbf{r}_n^{n+1})$, written in block-form as

$$\mathbf{b}(\mathbf{r}_n^{n+1})\mathbf{b}^\top(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \gamma_1(\mathbf{r}_n^{n+1}) & \gamma_2(\mathbf{r}_n^{n+1}) \\ \gamma_2^\top(\mathbf{r}_n^{n+1}) & \gamma_4(\mathbf{r}_n^{n+1}) \end{bmatrix}. \quad (18)$$

Since $\mathbf{a}_3(\mathbf{r}_1^2) = 0$ for each \mathbf{r}_1^2 in Ω^2 , equation (16a) holds for

$$\begin{aligned} \mathbf{D}(\mathbf{r}_n^{n+1}) &= \mathbf{a}_4(\mathbf{r}_n^{n+1}), \\ \mathbf{H}(\mathbf{r}_n^{n+1}) &= -\mathbf{a}_4(\mathbf{r}_n^{n+1})M_{r_n}^{\mathbf{Y}} + M_{r_{n+1}}^{\mathbf{Y}} \end{aligned}$$

and for some matrix $\mathbf{\Lambda}(\mathbf{r}_n^{n+1})$ such that

$$\mathbf{\Lambda}(\mathbf{r}_n^{n+1})\mathbf{\Lambda}^\top(\mathbf{r}_n^{n+1}) = \gamma_4(\mathbf{r}_n^{n+1}). \quad (19)$$

Likewise, \mathbf{X}_{n+1} is also normally distributed conditional on \mathbf{X}_n , \mathbf{R}_n^{n+1} and \mathbf{Y}_n^{n+1} . The conditional variance of \mathbf{X}_{n+1} is $\gamma_1(\mathbf{r}_n^{n+1}) - \gamma_2(\mathbf{r}_n^{n+1})\gamma_4^{-1}(\mathbf{r}_n^{n+1})\gamma_2^\top(\mathbf{r}_n^{n+1})$, and its conditional mean is

$$\begin{aligned} & M_{r_{n+1}}^{\mathbf{X}} + \mathbf{a}_1(\mathbf{r}_n^{n+1})(\mathbf{x}_n - M_{r_n}^{\mathbf{X}}) + \mathbf{a}_2(\mathbf{r}_n^{n+1})(\mathbf{y}_n - M_{r_n}^{\mathbf{Y}}) \\ & + \gamma_2(\mathbf{r}_n^{n+1})\gamma_4^{-1}(\mathbf{r}_n^{n+1})\left\{ \mathbf{y}_{n+1} - \left(M_{r_{n+1}}^{\mathbf{Y}} + \mathbf{a}_3(\mathbf{r}_n^{n+1})(\mathbf{x}_n - M_{r_n}^{\mathbf{X}}) \right. \right. \\ & \left. \left. + \mathbf{a}_4(\mathbf{r}_n^{n+1})(\mathbf{y}_n - M_{r_n}^{\mathbf{Y}}) \right) \right\}. \end{aligned}$$

Term-by-term identification of (16b) with the equation above gives

$$\begin{aligned} \mathbf{C}(\mathbf{r}_n^{n+1}) &= \gamma_2(\mathbf{r}_n^{n+1})\gamma_4^{-1}(\mathbf{r}_n^{n+1}) \\ \mathbf{A}(\mathbf{r}_n^{n+1}) &= \mathbf{a}_1(\mathbf{r}_n^{n+1}) - \mathbf{C}(\mathbf{r}_n^{n+1})\mathbf{a}_3(\mathbf{r}_n^{n+1}) \\ \mathbf{B}(\mathbf{r}_n^{n+1}) &= \mathbf{a}_2(\mathbf{r}_n^{n+1}) - \mathbf{C}(\mathbf{r}_n^{n+1})\mathbf{a}_4(\mathbf{r}_n^{n+1}) \\ \mathbf{F}(\mathbf{r}_n^{n+1}) &= M_{r_{n+1}}^{\mathbf{X}} - \mathbf{A}(\mathbf{r}_n^{n+1})M_{r_n}^{\mathbf{X}} - \mathbf{B}(\mathbf{r}_n^{n+1})M_{r_n}^{\mathbf{Y}} \\ & \quad - \mathbf{C}(\mathbf{r}_n^{n+1})M_{r_{n+1}}^{\mathbf{Y}} \end{aligned}$$

and $\mathbf{\Pi}(\mathbf{r}_n^{n+1})$ is a matrix such that

$$\mathbf{\Pi}(\mathbf{r}_n^{n+1})\mathbf{\Pi}^\top(\mathbf{r}_n^{n+1}) = \gamma_1(\mathbf{r}_n^{n+1}) - \mathbf{C}(\mathbf{r}_n^{n+1})\gamma_2^\top(\mathbf{r}_n^{n+1}). \quad (20)$$

B. Filtering in the CGOMSM

Let us recall how fast filter runs in a stationary CGOMSM. We sequentially compute $p(r_{n+1}|\mathbf{y}_1^{n+1})$, $\mathbb{E}[\mathbf{X}_{n+1}|r_{n+1}, \mathbf{y}_1^{n+1}]$ and $\mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}]$ using $p(r_n|\mathbf{y}_1^n)$, $\mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n]$, $\mathbb{E}[\mathbf{X}_n\mathbf{X}_n^\top|r_n, \mathbf{y}_1^n]$ and parameters $\mathbf{D}(\mathbf{r}_n^{n+1})$, $\mathbf{H}(\mathbf{r}_n^{n+1})$, $\mathbf{\Lambda}(\mathbf{r}_n^{n+1})$, $\mathbf{A}(\mathbf{r}_n^{n+1})$, $\mathbf{B}(\mathbf{r}_n^{n+1})$, $\mathbf{C}(\mathbf{r}_n^{n+1})$, $\mathbf{F}(\mathbf{r}_n^{n+1})$, $\mathbf{\Pi}(\mathbf{r}_n^{n+1})$ as follows.

Since $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a pairwise Markov model

$$p(r_{n+1}|\mathbf{y}_1^{n+1}) = \frac{\sum_{r_n} p(r_{n+1}, \mathbf{y}_{n+1}|r_n, \mathbf{y}_n) p(r_n|\mathbf{y}_1^n)}{\sum_{r_n, r_{n+1}^*} p(r_{n+1}, \mathbf{y}_{n+1}|r_n, \mathbf{y}_n) p(r_n|\mathbf{y}_1^n)} \quad (21)$$

$$p(r_n|r_{n+1}, \mathbf{y}_1^{n+1}) = \frac{p(r_{n+1}, \mathbf{y}_{n+1}|r_n, \mathbf{y}_n) p(r_n|\mathbf{y}_1^n)}{\sum_{r_n^*} p(r_{n+1}, \mathbf{y}_{n+1}|r_n^*, \mathbf{y}_n) p(r_n^*|\mathbf{y}_1^n)}. \quad (22)$$

Then,

$$\begin{aligned} \mathbb{E}[\mathbf{X}_{n+1}|r_{n+1}, \mathbf{y}_1^{n+1}] &= \sum_{r_n} p(r_n|r_{n+1}, \mathbf{y}_1^{n+1}) \\ & (\mathbf{A}(\mathbf{r}_n^{n+1})\mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n] + \mathbf{L}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})) \end{aligned} \quad (23)$$

with $\mathbf{L}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{C}(\mathbf{r}_n^{n+1})\mathbf{y}_{n+1} + \mathbf{F}(\mathbf{r}_n^{n+1})$. We obtain the equation above by taking the conditional expectation of (16b) and this result is mainly due to (15) and, consequently, to the fact that

$$\mathbb{E}[\mathbf{X}_n|\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}] = \mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n]. \quad (24)$$

A similar approach applies to the computation of $\mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}]$. It may be shown that

$$\begin{aligned} & \mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}] \\ & = \mathbf{A}(\mathbf{r}_n^{n+1})\mathbb{E}[\mathbf{X}_n\mathbf{X}_n^\top|r_n, \mathbf{y}_1^n]\mathbf{A}^\top(\mathbf{r}_n^{n+1}) \\ & \quad + \mathbf{A}(\mathbf{r}_n^{n+1})\mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n]\mathbf{L}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \\ & \quad + \mathbf{L}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\mathbb{E}[\mathbf{X}_n^\top|r_n, \mathbf{y}_1^n]\mathbf{A}^\top(\mathbf{r}_n^{n+1}) \\ & \quad + \mathbf{L}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\mathbf{L}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \mathbf{\Pi}(\mathbf{r}_n^{n+1})\mathbf{\Pi}^\top(\mathbf{r}_n^{n+1}). \end{aligned} \quad (25)$$

Then

$$\begin{aligned} & \mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}] \\ & = \sum_{r_n} p(r_n|r_{n+1}, \mathbf{y}_1^{n+1}) \mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_n^{n+1}, \mathbf{y}_1^{n+1}]. \end{aligned} \quad (26)$$

Finally, our fast optimal filter is:

Algorithm III.1

- i) Use $p(r_n|\mathbf{y}_1^n)$ and \mathbf{y}_{n+1} to obtain $p(r_{n+1}|\mathbf{y}_1^{n+1})$ and $p(r_n|r_{n+1}, \mathbf{y}_1^{n+1})$ by (21), (22);
 - ii) Use $\mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n]$, $p(r_n|r_{n+1}, \mathbf{y}_1^{n+1})$ and \mathbf{y}_{n+1} to obtain $\mathbb{E}[\mathbf{X}_{n+1}|r_{n+1}, \mathbf{y}_1^{n+1}]$ by (23);
 - iii) Use $\mathbb{E}[\mathbf{X}_n\mathbf{X}_n^\top|r_n, \mathbf{y}_1^n]$, $\mathbb{E}[\mathbf{X}_n|r_n, \mathbf{y}_1^n]$ and \mathbf{y}_{n+1} to compute $\mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}]$ by (25). Then use $\mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}]$ and $p(r_n|r_{n+1}, \mathbf{y}_1^{n+1})$ to obtain $\mathbb{E}[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top|r_{n+1}, \mathbf{y}_1^{n+1}]$ by (26).
-

IV. APPROXIMATING A NONLINEAR NON-GAUSSIAN SYSTEM WITH THE CGOMSM

We recall that according to our CGOMSM-ABF principle, we contemplate estimation of the CGOMSM parameters through a signal $(\mathbf{x}_1^{M*}, \mathbf{y}_1^{M*})$ which we sample by using (1), (2) or any other time-invariant system cf. Remark IV.1.

To this end, we use a common algorithm known as ‘‘Expectation–Maximization’’ [35] (EM). Let us then parametrize the CGOMSM by $\Theta = \{\boldsymbol{\mu}_i, \boldsymbol{\Gamma}_i, \mathbf{p}_{ij}, \mathbf{A}_{ij}, \mathbf{B}_{ij}, \mathbf{C}_{ij}, \mathbf{D}_{ij}, \mathbf{F}_{ij}, \mathbf{H}_{ij}, \boldsymbol{\Pi}_{ij}, \boldsymbol{\Lambda}_{ij} | (i, j) \in \Omega^2\}$, where:

- for each i in Ω , $\boldsymbol{\mu}_i$ and $\boldsymbol{\Gamma}_i$ define the pdf $p(\mathbf{x}_1, \mathbf{y}_1 | r_1)$;
- for each i, j in Ω , $\mathbf{p}_{ij} = P(r_1 = i, r_2 = j)$ and $\mathbf{A}_{ij}, \mathbf{B}_{ij}, \mathbf{C}_{ij}, \mathbf{D}_{ij}, \mathbf{F}_{ij}, \mathbf{H}_{ij}, \boldsymbol{\Pi}_{ij}, \boldsymbol{\Lambda}_{ij}$ are described in (16).

To make the text easier to read, we will denote the artificial signal by $(\mathbf{x}_1^M, \mathbf{y}_1^M)$ instead of $(\mathbf{x}_1^{M*}, \mathbf{y}_1^{M*})$. Thus, \mathbf{y}_1^M stands for the pre-sampled data and should not be confused with \mathbf{y}_1^N , which is the noisy input signal.

The crux point is that \mathbf{X}_{n+1} and \mathbf{Y}_{n+1} are Gaussian conditional on the pair (R_n, R_{n+1}) and on a linear combination of $\mathbf{X}_n, \mathbf{Y}_n$ so an appropriate linear regression per each pair (R_n, R_{n+1}) enables to determine the coefficients of the combination and to solve the M-step of the EM algorithm. We detail below the EM algorithm suited to our context (cf. Section II).

Algorithm IV.1. EM Estimation

- a) Make an initial guess $\Theta^0 = \{\boldsymbol{\mu}_i^0, \boldsymbol{\Gamma}_i^0, \mathbf{p}_{ij}^0, \mathbf{A}_{ij}^0, \mathbf{B}_{ij}^0, \mathbf{C}_{ij}^0, \mathbf{D}_{ij}^0, \mathbf{F}_{ij}^0, \mathbf{H}_{ij}^0, \boldsymbol{\Pi}_{ij}^0, \boldsymbol{\Lambda}_{ij}^0 | (i, j) \in \Omega^2\}$ as follows:

- 1) Apply the K-means clustering method to \mathbf{x}_1^M . We will denote by $\kappa_n(i)$ the function which assigns 1 if \mathbf{x}_n is within the i th cluster, and 0 otherwise. We also note $\delta_n(i, j) = \kappa_n(i)\kappa_{n+1}(j)$;
- 2) For each i in Ω , $\boldsymbol{\mu}_i^0$ and $\boldsymbol{\Gamma}_i^0$ are given by

$$\boldsymbol{\mu}_i^0 = \frac{\sum_{n=1}^M \mathbf{z}_n \kappa_n(i)}{\sum_{n=1}^M \kappa_n(i)} \quad (27a)$$

$$\boldsymbol{\Gamma}_i^0 = \frac{\sum_{n=1}^M (\mathbf{z}_n - \boldsymbol{\mu}_i^0) (\mathbf{z}_n - \boldsymbol{\mu}_i^0)^\top \kappa_n(i)}{\sum_{n=1}^M \kappa_n(i)} \quad (27b)$$

where $\mathbf{z}_n^\top = [\mathbf{x}_n^\top, \mathbf{y}_n^\top]$, and for each (i, j) in Ω^2 , \mathbf{p}_{ij}^0 is given by

$$\mathbf{p}_{ij}^0 = \frac{1}{M-1} \sum_{n=1}^{M-1} \delta_n(i, j). \quad (28)$$

- 3) Compute intermediate matrices $\mathbf{E}_{ij}^0, \mathbf{S}_{ij}^0, \boldsymbol{\chi}_{ij}^0, \boldsymbol{\Phi}_{ij}^0, \mathbf{G}_{ij}^0, \mathbf{P}_{ij}^0, \boldsymbol{\xi}_{ij}^0$, and \mathbf{T}_{ij}^0 as follows:

$$\mathbf{E}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \begin{bmatrix} \mathbf{z}_n \\ \mathbf{y}_{n+1} \end{bmatrix} \delta_n(i, j) \quad (29a)$$

$$\mathbf{S}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \begin{bmatrix} \mathbf{z}_n \mathbf{z}_n^\top & \mathbf{z}_n \mathbf{y}_{n+1}^\top \\ \mathbf{y}_{n+1} \mathbf{z}_n^\top & \mathbf{y}_{n+1} \mathbf{y}_{n+1}^\top \end{bmatrix} \delta_n(i, j) \quad (29b)$$

$$\boldsymbol{\chi}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \begin{bmatrix} \mathbf{x}_{n+1} \mathbf{z}_n^\top & \mathbf{x}_{n+1} \mathbf{y}_{n+1}^\top \end{bmatrix} \delta_n(i, j) \quad (29c)$$

$$\boldsymbol{\Phi}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{x}_{n+1} \delta_n(i, j) \quad (29d)$$

$$\mathbf{G}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{y}_n \delta_n(i, j) \quad (29e)$$

$$\mathbf{P}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{y}_n \mathbf{y}_n^\top \delta_n(i, j) \quad (29f)$$

$$\boldsymbol{\xi}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{y}_{n+1} \mathbf{y}_n^\top \delta_n(i, j) \quad (29g)$$

$$\mathbf{T}_{ij}^0 = \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{y}_{n+1} \delta_n(i, j). \quad (29h)$$

- 4) For each i, j in Ω , $\mathbf{A}_{ij}^0, \mathbf{B}_{ij}^0, \mathbf{C}_{ij}^0, \mathbf{D}_{ij}^0, \mathbf{F}_{ij}^0, \mathbf{H}_{ij}^0, \boldsymbol{\Pi}_{ij}^0$, and $\boldsymbol{\Lambda}_{ij}^0$ are given by

$$\begin{bmatrix} \mathbf{F}_{ij}^0 & \mathbf{A}_{ij}^0 & \mathbf{B}_{ij}^0 & \mathbf{C}_{ij}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{ij}^0 & \boldsymbol{\chi}_{ij}^0 \end{bmatrix} \begin{bmatrix} M-1 & (\mathbf{E}_{ij}^0)^\top \\ \mathbf{E}_{ij}^0 & \mathbf{S}_{ij}^0 \end{bmatrix}^{-1} \quad (30a)$$

$$\begin{bmatrix} \mathbf{H}_{ij}^0 & \mathbf{D}_{ij}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{ij}^0 & \boldsymbol{\xi}_{ij}^0 \end{bmatrix} \begin{bmatrix} M-1 & (\mathbf{G}_{ij}^0)^\top \\ \mathbf{G}_{ij}^0 & \mathbf{P}_{ij}^0 \end{bmatrix}^{-1} \quad (30b)$$

$$\begin{aligned} (M-1) \boldsymbol{\Lambda}_{ij}^0 (\boldsymbol{\Lambda}_{ij}^0)^\top &= \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{y}_{n+1} \mathbf{y}_{n+1}^\top \delta_n(i, j) - \mathbf{H}_{ij}^0 (\mathbf{T}_{ij}^0)^\top \\ &\quad - \mathbf{D}_{ij}^0 (\boldsymbol{\xi}_{ij}^0)^\top \end{aligned} \quad (30c)$$

$$\begin{aligned} (M-1) \boldsymbol{\Pi}_{ij}^0 (\boldsymbol{\Pi}_{ij}^0)^\top &= \frac{1}{\mathbf{p}_{ij}^0} \sum_{n=1}^{M-1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^\top \delta_n(i, j) - \mathbf{F}_{ij}^0 (\boldsymbol{\Phi}_{ij}^0)^\top \\ &\quad - \begin{bmatrix} \mathbf{A}_{ij}^0 & \mathbf{B}_{ij}^0 & \mathbf{C}_{ij}^0 \end{bmatrix} (\boldsymbol{\chi}_{ij}^0)^\top. \end{aligned} \quad (30d)$$

- b) Find the new set of parameters Θ^{q+1} as follows:

- 1) For each i in Ω , compute the posterior probabilities $\phi_n^q(i) = p(r_n = i | \mathbf{x}_1^M, \mathbf{y}_1^M, \Theta^q)$, and for each i, j in Ω compute $\psi_n^q(i, j) = p(r_n = i, r_{n+1} = j | \mathbf{x}_1^M, \mathbf{y}_1^M, \Theta^q)$ (computational details are provided below);
- 2) For each i in Ω , compute $\boldsymbol{\mu}_i^{q+1}$ and $\boldsymbol{\Gamma}_i^{q+1}$ by substitution $\phi_n^q(i)$ for $\kappa_n(i)$ in (27);
- 3) For each i, j in Ω , \mathbf{p}_{ij}^{q+1} is given by

$$\mathbf{p}_{ij}^{q+1} = \frac{1}{M-1} \sum_{n=1}^{M-1} \psi_n^q(i, j). \quad (31)$$

Then compute intermediate matrices E_{ij}^{q+1} , S_{ij}^{q+1} , χ_{ij}^{q+1} , Φ_{ij}^{q+1} , G_{ij}^{q+1} , P_{ij}^{q+1} , ξ_{ij}^{q+1} , and T_{ij}^{q+1} by substituting $\psi_n^q(i, j)$, \mathbf{p}_{ij}^{q+1} with $\delta_n(i, j)$, \mathbf{p}_{ij}^0 in (29). Finally, compute A_{ij}^{q+1} , B_{ij}^{q+1} , C_{ij}^{q+1} , D_{ij}^{q+1} , F_{ij}^{q+1} , H_{ij}^{q+1} , Π_{ij}^{q+1} , and Λ_{ij}^{q+1} by substituting E_{ij}^{q+1} , S_{ij}^{q+1} , χ_{ij}^{q+1} , Φ_{ij}^{q+1} , G_{ij}^{q+1} , P_{ij}^{q+1} , ξ_{ij}^{q+1} , T_{ij}^{q+1} with E_{ij}^0 , S_{ij}^0 , χ_{ij}^0 , Φ_{ij}^0 , G_{ij}^0 , P_{ij}^0 , ξ_{ij}^0 , T_{ij}^0 in (30).

Details on the derivation of Algorithm IV.1 will be made available upon request to the corresponding author.

Let us recall the formulas for $\phi_n^q(i)$ and $\psi_n^q(i, j)$. Let us set $\mathbf{t}_n = (\mathbf{x}_n, r_n, \mathbf{y}_n)$, $\alpha_n(r_n) = p(r_n, \mathbf{z}_n^1)$ and $\beta_n(r_n) = p(\mathbf{z}_{n+1}^N | \mathbf{t}_n)$. Then, the forward-backward algorithm computes recursively $\alpha_n(r_n)$ and $\beta_n(r_n)$ as follows:

- $\alpha_1(r_1) = p(\mathbf{t}_1)$, $\alpha_{n+1}(r_{n+1}) = \sum_{r_n \in \Omega} \alpha_n(r_n) p(\mathbf{t}_{n+1} | \mathbf{t}_n)$ for $1 \leq n \leq M-1$;
- $\beta_M(r_M) = 1$, $\beta_n(r_n) = \sum_{r_{n+1} \in \Omega} \beta_{n+1}(r_{n+1}) p(\mathbf{t}_{n+1} | \mathbf{t}_n)$ for $1 \leq n \leq M-1$,

where

$$\begin{aligned} p(\mathbf{t}_1) &= p(r_1) p(\mathbf{z}_1 | r_1) \\ p(\mathbf{t}_{n+1} | \mathbf{t}_n) &= p(r_{n+1} | r_n) p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} | \mathbf{x}_n, \mathbf{y}_n, \mathbf{r}_n^{n+1}) \\ p(r_1 = i | \Theta^q) &= \sum_{j \in \Omega} \mathbf{p}_{ij}^q \\ p(\mathbf{z}_1 | r_1 = i, \Theta^q) &= \mathcal{N}(\mathbf{z}_1; \boldsymbol{\mu}_i^q, \boldsymbol{\Gamma}_i^q) \\ p(r_{n+1} = j | r_n = i, \Theta^q) &= \frac{\mathbf{p}_{ij}^q}{p(r_1 = i, \Theta^q)} \\ p(\mathbf{y}_{n+1} | \mathbf{y}_n, \mathbf{r}_n^{n+1} = (i, j), \Theta^q) &= \mathcal{N}(\mathbf{y}_{n+1}; \mathbf{D}_{ij}^q \mathbf{y}_n + \mathbf{H}_{ij}^q, \boldsymbol{\Lambda}_{ij}^q (\boldsymbol{\Lambda}_{ij}^q)^\top) \\ p(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1} = (i, j), \Theta^q) &= \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}_{ij}^q \mathbf{x}_n + \mathbf{B}_{ij}^q \mathbf{y}_n + \mathbf{C}_{ij}^q \mathbf{y}_{n+1} + \mathbf{F}_{ij}^q, \boldsymbol{\Pi}_{ij}^q (\boldsymbol{\Pi}_{ij}^q)^\top). \end{aligned}$$

Thus,

$$\psi_n^q(\mathbf{r}_n^{n+1}) = \frac{\alpha_n(r_n) p(\mathbf{t}_{n+1} | \mathbf{t}_n) \beta_{n+1}(r_{n+1})}{\sum_{r_n^*, r_{n+1}^*} \alpha_n(r_n^*) p(\mathbf{t}_{n+1}^* | \mathbf{t}_n^*) \beta_{n+1}(r_{n+1}^*)} \quad (32)$$

with $\mathbf{t}_n^* = (\mathbf{x}_n, r_n^*, \mathbf{y}_n)$.

Finally, the CGOMSM-ABF implementation, which we use in experiments, runs as follows:

Algorithm IV.2

- a) Generate an artificial sample $(\mathbf{x}_1^{M*}, \mathbf{y}_1^{M*})$ according to the model considered;
 - b) Since $(\mathbf{x}_1^{M*}, \mathbf{y}_1^{M*})$ may be seen as a realization of some CGOMSM $(\mathbf{X}_1^{M*}, \mathbf{R}_1^{M*}, \mathbf{Y}_1^{M*})$ with missing \mathbf{R}_1^{M*} , estimate Θ with Algorithm IV.1;
 - c) Filtering: when a new measurement \mathbf{y}_{n+1} is received, compute $p(r_{n+1} | \mathbf{y}_1^{n+1})$, $\mathbb{E}[\mathbf{X}_{n+1} | r_{n+1}, \mathbf{y}_1^{n+1}]$ and $\mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | r_{n+1}, \mathbf{y}_1^{n+1}]$ using Algorithm III.1, then $\mathbb{E}[\mathbf{X}_{n+1} | \mathbf{y}_1^{n+1}]$ and $\mathbb{E}[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top | \mathbf{y}_1^{n+1}]$ are given by (11) and (12).
-

Remark IV.1: A given switching model can have different meanings. Here we use one to approximate a nonlinear non-Gaussian system. Another possible situation is that the system is linear Gaussian, and there are random switches. Although quite different, these contexts can be dealt with using the same model, as the CGOMSM used in this paper. Of course, there also exists non-Gaussian switching systems [36], [37].

V. EXPERIMENTS ON STOCHASTIC VOLATILITY MODELS

A stochastic volatility model describes the variance of a stochastic process. Examples of stochastic volatility models include the Heston model, the GARCH model and many more [15], [38], [39]. In this Section, we provide two series of Monte-Carlo experiments and an example of estimating the volatility of a series of stock returns. In the first series, we focus on a standard stochastic volatility (SV) model [24]–[27], that is usually presented as follows:

$$X_1 = \mu + U_1 \quad (33a)$$

$$X_{n+1} = \mu + \phi(X_n - \mu) + \sigma U_{n+1} \quad (33b)$$

$$Y_n = \beta \exp\left(\frac{X_n}{2}\right) V_n \quad (33c)$$

where $U_1, V_1, \dots, U_N, V_N$ are independent standard Gaussian vectors and μ, ϕ, β, σ are fixed. Since this model is of the form (1), it is an HMM. The second series of experiments is devoted to a recent extension, known as the asymmetric SV (ASV) model [28]–[30], which may be presented as follows:

$$X_1 = \mu + U_1 \quad (34a)$$

$$X_{n+1} = \mu + \phi(X_n - \mu) + \sigma \left(\frac{\rho Y_n}{\beta \exp\left(\frac{X_n}{2}\right)} + \lambda U_{n+1} \right) \quad (34b)$$

$$Y_n = \beta \exp\left(\frac{X_n}{2}\right) V_n. \quad (34c)$$

Since both SV and ASV are generative models, they fit well to our CGOMSM-ABF approach.

If $\hat{\mathbf{x}}(y_1^n)$ is a filtered signal obtained from \mathbf{y}_1^N , and \mathbf{x}_1^N is the “ground-truth,” then the mean squared error (MSE) of the filter can be estimated by

$$\text{MSE} = \sum_{n=1}^N (x_n - \hat{x}_n(y_1^n))^2 \quad (35)$$

which is a useful performance criterion for comparing filtering effectiveness. All the results presented below are means of 100 equivalent independent experiments, each of them being computed using $N = 1000$ simulated data points.

A. SV Model

Here we compare the performance of our method with respect to the particle filtering (PF) and the Gaussian sum filtering (GSF), using data simulated by different stationary SV models. We set $\mu = 0.5$, $\beta = 0.5$, and consider four different cases for ϕ and σ such that $\phi^2 + \sigma^2 = 1$ (that is to ensure that the common

TABLE I
MEAN MSE RESULTS FOR DIFFERENT SV MODELS DEFINED BY ϕ AND σ ($\mu = 0.5, \beta = 0.5$)

Cases	ϕ	σ^2	CGOMSM-ABF				PF	GSF
			$K = 2$	$K = 3$	$K = 5$	$K = 7$		
1	0.99	0.0199	0.41	0.27	0.20	0.19	0.18	0.21
2	0.90	0.1900	0.55	0.49	0.47	0.46	0.46	0.50
3	0.80	0.3600	0.63	0.59	0.58	0.57	0.57	0.60
4	0.50	0.7500	0.72	0.71	0.70	0.70	0.70	0.72

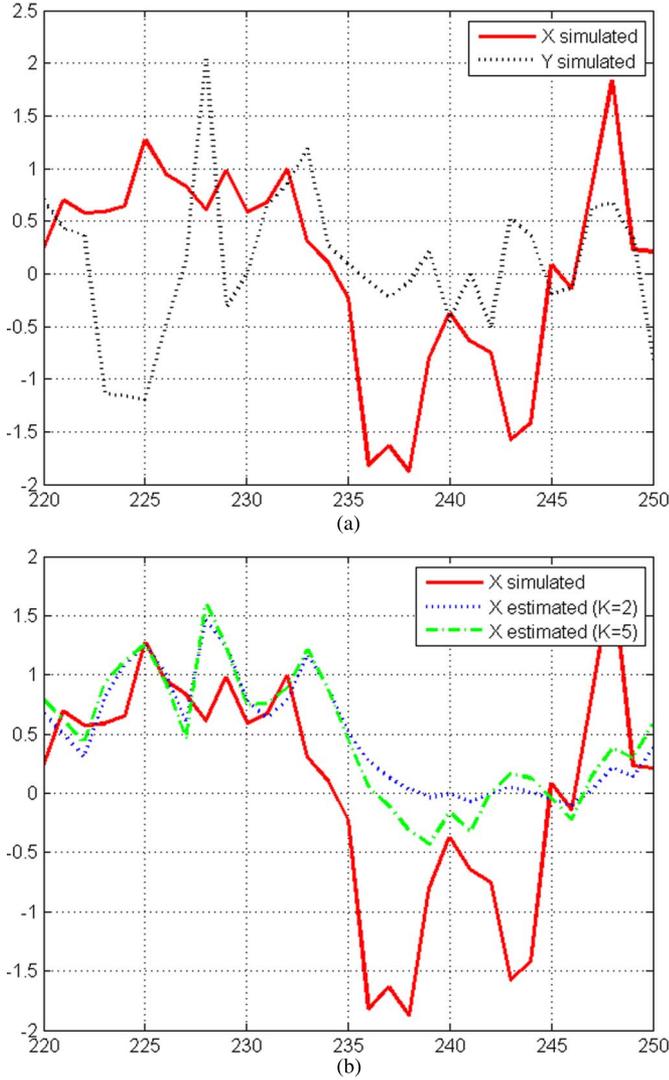


Fig. 1. Up: simulated log-volatility trajectory with a SV model (red, plain), simulated log-returns (black, dotted). Down: log-volatility estimates computed using $K = 2$ classes (blue, dotted), and $K = 5$ classes (green, dashed).

variance of the variables X_n is unitary). The results are reported in Table I.

Here are the details of each filtering method used in the experiments

- For our method, we try out different values of K (cf. Remark V.1) and we infer the CGOMSM from some independently generated sample (x_1^{M*}, y_1^{M*}) of size $M = 20\,000$, performing 100 EM iterations. See Fig. 1 for an example of trajectories.
- The PF implementation uses 1500 particles. We found out empirically that PF behaves asymptotically for this particle number.

- In order to use the GSF, let us linearise the SV model by taking the log of both sides of (33c), to get

$$X_1 = \mu + U_1 \tag{36a}$$

$$X_{n+1} = \mu + \phi(X_n - \mu) + \sigma U_{n+1} \tag{36b}$$

$$Y'_n = X_n + V'_n \tag{36c}$$

where $Y'_n = \log(Y_n^2) - 2 \log \beta$ and V'_1, \dots, V'_n are independent, non-Gaussian variables, such that $\exp(V'_1/2), \dots, \exp(V'_n/2)$ are standard Gaussians. Then, the common pdf of V'_n is $p(v) = \exp(v/2)\mathcal{N}(\exp(v/2); 0, 1)$, which we approximate by a Gaussian mixture of r components using the EM algorithm: $p(v'_n) \approx \sum_{m=1}^r \gamma_m \mathcal{N}(v'_n; \hat{v}'_m, R_m)$. We found that when $r \geq 5$, the approximation is accurate enough to achieve a negligible residual effect. Since the number ξ_n of mixands in the filtering pdf $p(x_n|y_n^n) = \sum_{j=1}^{\xi_n} \alpha_{nj} \mathcal{N}(x_n; \hat{x}_{nj}, P_{nj})$ exponentially grows with time, a reduction technique is implemented to keep computational demands of the algorithm within reasonable bounds [40].

For our experiments, we decided to reduce the number of terms as follows: when ξ_n becomes greater than r , we keep the r mixands in $\sum_{j=1}^{\xi_n} \alpha_{nj} \mathcal{N}(x_n; \hat{x}_{nj}, P_{nj})$ which have the greatest weight coefficients α_{nj} , and we discard the remaining. Therefore, we impose the constraint that $\xi_n = r$. We found out empirically that GSF behaves asymptotically for $r \geq 3$, but does not attain the optimal MSE.

We note that since the model (36) is linear, there is no reason for considering the extensions of the GSF for non-linear systems, such as Gaussian sum unscented Kalman filter [20] or Gaussian sum unscented Kalman filter with adaptive scaling parameters [21].

Contrary to our filtering method, which makes use of a single global approximation, the GSUKF relies on multiple approximations:

- an approximation of the noise terms with a Gaussian mixture;
- some reduction technique to keep the number of mixands of the filtering pdf within reasonable bounds.

Additionally, when the model is nonlinear, the GSUKF uses the unscented transform (UT) for computing the approximate means and covariances. And the UT relies, in turn, on its scaling parameters. Our experiments show that computing a single global approximation may be advantageous and helps to avoid the residual cumulative effects. However, unlike our method, the GSUKF may be used for non-stationary systems.

B. Asymmetric SV Model

Regarding the ASV model, we compare the performance between our method and PF only, since the GSF and GSUKF

TABLE II
MEAN MSE RESULTS FOR DIFFERENT ASV MODELS DEFINED BY ρ AND λ ($\mu = 0.5, \beta = 0.5,$ AND $\sigma^2 = 0.75$), FOR $\phi = 0.5$

Cases	ρ	λ^2	CGOMSM-ABF			PF
			$K = 2$	$K = 3$	$K = 5$	
1	-0.9	0.19	0.23	0.22	0.20	0.20
2	-0.8	0.36	0.36	0.35	0.34	0.33
3	-0.5	0.75	0.59	0.58	0.58	0.57
4	-0.3	0.91	0.68	0.67	0.66	0.65
5	0.0	1.00	0.72	0.71	0.70	0.70

TABLE III
MEAN MSE RESULTS FOR DIFFERENT ASV MODELS DEFINED BY ρ AND λ ($\mu = 0.5, \beta = 0.5,$ AND $\sigma^2 = 0.75$), FOR $\phi = 0.8$

Cases	ρ	λ^2	CGOMSM-ABF			PF
			$K = 2$	$K = 3$	$K = 5$	
1	-0.9	0.19	0.22	0.21	0.19	0.18
2	-0.8	0.36	0.33	0.31	0.29	0.29
3	-0.5	0.75	0.52	0.49	0.48	0.47
4	-0.3	0.91	0.59	0.55	0.54	0.54
5	0.0	1.00	0.63	0.59	0.58	0.57

ignore the impact of the volatility asymmetry coefficient ρ and therefore are not suitable for this model (recalling that the ASV is not an HMM). The experimental configuration is identical to the previous one. For the sake of consistency with the asymmetric volatility phenomenon [41], ρ should be assumed negative.

We set $\mu = 0.5, \beta = 0.5, \sigma^2 = 0.75$, and consider five different cases for ρ and λ such that $\rho^2 + \lambda^2 = 1$ (to ensure that the common variance of the variables X_n is unitary). The results are reported in Table II for $\phi = 0.5$ and in Table III for $\phi = 0.8$. Fig. 2 shows an ASV trajectory, and its restoration with the CGOMSM-ABF algorithm for $K = 2$ and $K = 5$ classes.

According to our results, the approach we propose is efficient for both SV and ASV models, and attains the same asymptotic performances as the PF. Regarding the processing time, we find that:

- after having it adjusted to the SV model, our filter is nearly five times faster than the PF;
- the model approximation algorithm may take time depending on the system complexity.

Remark V.1: At the moment, we have no computational technique to select the minimum number of classes allowing to obtain asymptotic performances. We note only the tradeoff between the computational cost and the variance of the resulting estimates. Indeed, with a greater number of classes the former increases, while the latter decreases. In practice, five classes would be enough for most situations.

C. Recovering Volatility From Stock Market Data

Let us remind that if P_{n-1} denotes the stock price at the beginning of the previous trading day and if P_n denotes the stock price at the beginning of the current trading day, then:

- $R_n = (P_n - P_{n-1})/P_{n-1}$ is the current daily return on the stock investment;
- $u_n = \log(1 + R_n) = \log(P_n/P_{n-1})$ is the *continuously compounded daily return*. It is also often called the *log-return*.

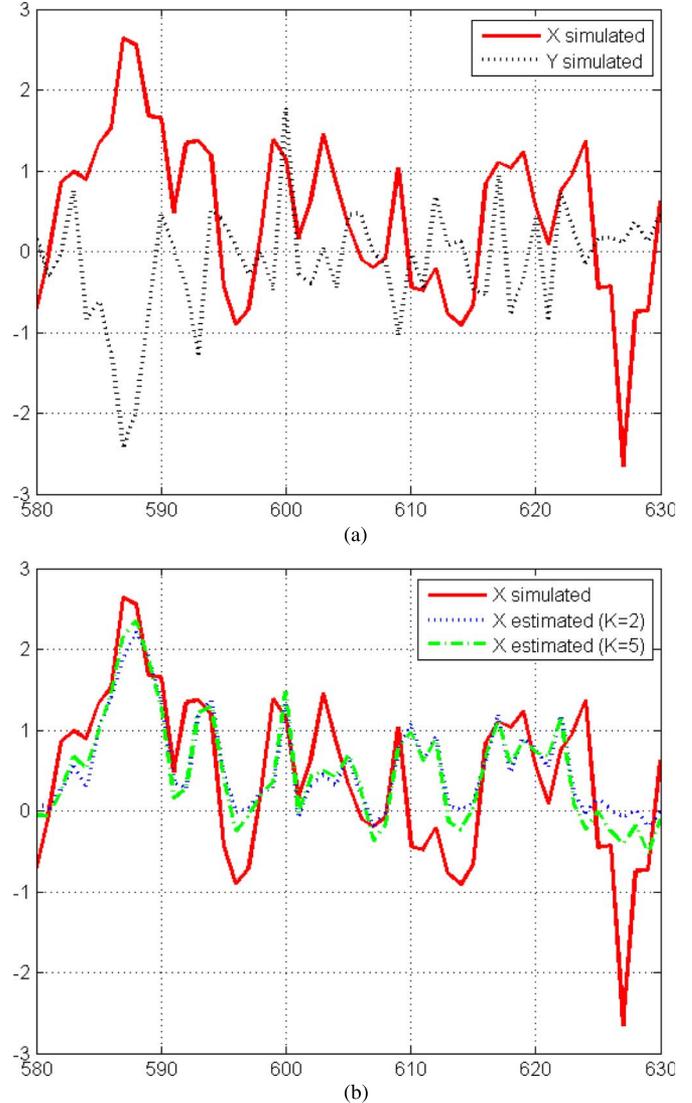


Fig. 2. Up: simulated log-volatility trajectory with an ASV model (red, plain), simulated log-returns (black, dotted). Down: log-volatility estimates computed using $K = 2$ classes (blue, dotted), and $K = 5$ classes (green, dashed).

To see why u_n is called the continuously compounded return, take the exponential of both sides to get $\exp(u_n) = P_n/P_{n-1}$. Rearranging, we get $P_n = P_{n-1} \exp(u_n)$ so that u_n is the continuously compounded growth rate in prices between the beginning of the previous and the current trading days. This has to be contrasted with R_n , which is the simple growth rate in prices P_{n-1} and P_n without any compounding.

Following [42]–[45] to examine the performance of our method on the stock market data, we compute the log-returns u_n over daily SPX index data from Jun. 23, 1980 to Aug. 30, 2002 ($N = 5604$), then we calculate $y_n = u_n^* - \mu_r$, where μ_r is given in [45] and u_n^* denotes pre-processed log-return with an ARMA(2,1) model [44]–[46]. Next, we use our algorithm to compute the filtered volatility estimates within the ASV model, whose parameters are given in [45] and reported in Table IV. Our result is shown in Fig. 3.

We find that the volatility estimates produced by our algorithm are consistent with the log-return process: as we can see in

TABLE IV
PARAMETERS OF THE ASV MODEL FOR THE STOCK MARKET DATA

Param.	μ_r	μ	ϕ	σ	ρ	β
Value	$7 \cdot 10^{-5}$	-9.54	0.98	0.17	-0.43	1.00

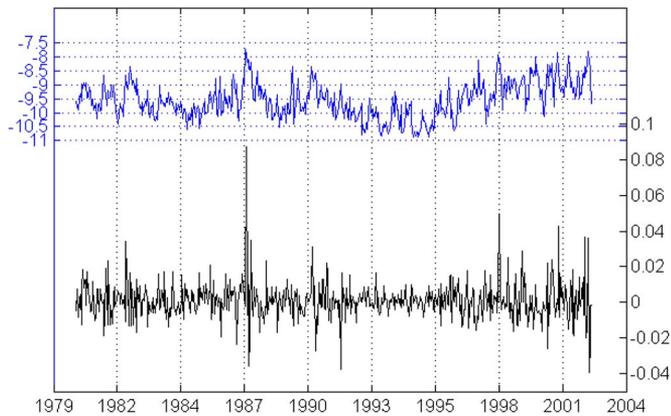


Fig. 3. Trajectories of the SPX index log-returns (down) and log-volatility estimates (up). The x -axis represents the dates for both trajectories, the y -axis labeling on the left concerns the log-volatility values, and the y -axis labeling on the right is related to the log-return values.

TABLE V
MEAN SQUARE DISTANCES BETWEEN OUR VOLATILITY ESTIMATES AND THOSE FROM THE PF, WITH DIFFERENT NUMBER OF CLASSES

$K = 5$	$K = 7$	$K = 9$	$Var[\mathbf{X}]$
0.0022	0.0017	0.0015	0.6145

Fig. 3, the intervals where the fluctuation of log-returns are low (e.g., between 1991 and 1995) match the intervals where the log-volatility is low, and vice versa. Moreover, we calculated the mean squared distance between our volatility estimates and those of the particle filter, and we find that this distance is negligible compared to the variance of the log-volatility process. Furthermore, when the number K of classes in our algorithm increases, this distance decreases as shown in Table V.

VI. CONCLUSION

We proposed a method to estimate the latent variables in nonlinear and non-Gaussian systems, called CGOMSM-ABF for “conditionally Gaussian observed Markov switching model—approximation based filter.” It makes use of a single global approximation of the system. Our method is very general and has several advantages over existing techniques. Its performance has been examined on synthetic samples of the standard and asymmetric stochastic volatility, as well as on real data (SPX index returns). Indeed, we found that our method attains the asymptotic performances of the particle filter, what could not be obtained with the Gaussian sum filter [17], [18].

The filtering procedure, which is the object of the paper, is applicable in general stationary (or asymptotically stationary) Markov dynamic systems, provided that one can sample its realizations. It is as fast as the standard Kalman filter, however, one has to adjust the filter to a particular model by using, for instance, the EM-based algorithm that we have supplied.

Our future prospects include filtering for various stochastic volatility models, including those with high dimension of the

state space, i.e., multifactor volatility models. Another important topic would be the unsupervised estimation of parameters of the initial PMM model in (2) [47], [48].

REFERENCES

- [1] C. Andrieu, M. Davy, and A. Doucet, “Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions,” *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, Jul. 2003.
- [2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [3] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Berlin, Germany: Springer-Verlag, 2005.
- [4] A. Doucet and A. Johansen, *A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later*. London, U.K.: Oxford Univ. Press, 2011.
- [5] W. Pieczynski, “Exact filtering in conditionally Markov switching hidden linear models,” *Comptes Rendus Mathématique*, vol. 349, no. 9–10, pp. 587–590, 2011.
- [6] N. Abbassi, D. Benboudjema, and W. Pieczynski, “Kalman filtering approximations in triplet Markov Gaussian switching models,” in *Proc. IEEE Int. Workshop SSP’11*, Nice, France, Jun. 2011, pp. 290–294.
- [7] N. Abbassi, D. Benboudjema, S. Derrode, and W. Pieczynski, “Optimal filter approximations in conditionally Gaussian pairwise Markov switching models,” *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1104–1109, Apr. 2015.
- [8] S. Derrode and W. Pieczynski, “Exact fast computation of optimal filter in Gaussian switching linear systems,” *IEEE Signal Process. Lett.*, vol. 20, no. 7, pp. 701–704, Jul. 2013.
- [9] Y. Petetin and F. Desbouvries, “A class of fast exact Bayesian filters in dynamical models with jumps,” *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3643–3653, Jul. 15, 2014.
- [10] S. Derrode and W. Pieczynski, “Fast filter in nonlinear systems with application to stochastic volatility model,” in *Proc. EUSIPCO’14*, Lisbon, Portugal, Sep. 2014, pp. 290–294.
- [11] D. Creal, “A survey of sequential Monte Carlo methods for economics and finance,” *Econometric Rev.*, vol. 31, no. 3, pp. 245–296, 2012.
- [12] D. Duffie, J. Pan, and K. J. Singleton, “Transform analysis and asset pricing for affine jump diffusions,” *Econometrica*, vol. 68, no. 6, pp. 1343–1376, 2000.
- [13] R. F. Engle and J. G. Rangel, “The spline-GARCH model for low-frequency volatility and its global macroeconomic causes,” *Rev. Financial Studies*, vol. 21, pp. 1187–1222, 2008.
- [14] B. Eraker, “Do stock prices and volatility jump? Reconciling evidence from spot and option prices,” *J. Finance*, vol. 59, pp. 1367–1404, 2004.
- [15] C.-J. Kim and C. R. Nelson, *State-Space Models With Regime Switching*, Cambridge, MA, USA: MIT Press, 1999.
- [16] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwell, MA, USA: Artech House, 2004.
- [17] H. Sorenson and D. Alspach, “Recursive Bayesian estimation using Gaussian sums,” *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.
- [18] M. Šimandl and J. Kráľovec, “Filtering, prediction and smoothing with Gaussian sum representation,” in *System Identification (SYSID ’00)*, Oxford, U.K.: IFAC, 2001, pp. 1157–1162. [Online]. Available: http://www.kky.zcu.cz/en/publications/SimandlM_2001_Filteringprediction
- [19] E. A. Wan and R. V. D. Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proc. IEEE Adaptive Syst. Signal Process., Commun., Control Symp. (AS-SPCC’00)*, 2000, pp. 153–158.
- [20] M. Šimandl and J. Duník, “Sigma point Gaussian sum filter design using square root unscented filters,” in *Preprints of the 16th IFAC World Congress*, Prague, Czech Republic: IFAC, 2005, pp. 1–6. [Online]. Available: http://www.kky.zcu.cz/en/publications/SimandlM_2005_SigmapointGaussian
- [21] O. Straka, J. Duník, and M. Šimandl, “Gaussian sum unscented Kalman filter with adaptive scaling parameters,” in *Proc. 14th Int. Conf. Inf. Fusion (FUSION)*, Chicago, IL, USA, 2011, pp. 1–8.
- [22] B. Feng, H. Ma, M. Fu, and C. Yang, “Real-time state estimator without noise covariance matrices knowledge-fast minimum norm filtering algorithm,” *IET Control Theory Applicat.*, vol. 9, no. 9, p. 1422, 2015.
- [23] B. Feng, M. Fu, H. Ma, Y. Xia, and B. Wang, “Kalman filter with recursive covariance estimation-sequentially estimating process noise covariance,” *IEEE Trans. Industrial Electron.*, vol. 61, no. 11, pp. 6253–6263, Nov. 2014.
- [24] E. Ghysels, A. Harvey, and E. Renault, “Stochastic volatility,” *Handbook Statist.*, vol. 14, pp. 119–192, 1995.

- [25] E. Jacquier, N. G. Polson, and P. Rossi, "Bayesian analysis of stochastic volatility models," *J. Bus. Econ. Statist.*, vol. 12, no. 4, pp. 371–389, 1994.
- [26] N. Shephard, S. Kim, and S. Chib, "Stochastic volatility: Likelihood inference and comparison with ARCH models," *Rev. Econ. Studies*, vol. 65, no. 3, pp. 361–393, 1998.
- [27] S. J. Taylor, *Modeling Financial Times Series*. Singapore: World Scientific, 2007.
- [28] N. Y. Nikolaev, E. Smirnov, and L. M. de Menezes, "Nonlinear filtering of asymmetric stochastic volatility models and value-at-risk estimation," in *Proc. IEEE Conf. Comput. Intell. Financial Eng. Econ. (CIFER'14)*, London, U.K., 2014, pp. 310–317.
- [29] Y. Omori and T. Watanabe, "Block sampler and posterior mode estimation for asymmetric stochastic volatility models," *Comput. Statist. Data Anal.*, vol. 52, no. 6, pp. 2892–2910, 2008.
- [30] M. Takashi, Y. Omori, and T. Watanabe, "Estimating stochastic volatility models using daily returns and realized volatility simultaneously," *Comput. Statist. Data Anal.*, vol. 53, no. 6, pp. 2404–2426, 2009.
- [31] S. N. Ethier and T. G. Kurtz, *Markov Processes: Characterization and Convergence*. New York, NY, USA: Wiley, 2009, vol. 282.
- [32] D. Bakry, P. Cattiaux, and A. Guillin, "Rate of convergence for ergodic continuous Markov processes: Lyapunov versus Poincaré," *J. Funct. Anal.*, vol. 254, no. 3, pp. 727–759, 2008.
- [33] R. Douc, G. Fort, E. Moulines, and P. Soulier, "Practical drift conditions for subgeometric rates of convergence," *Ann. Appl. Probab.*, pp. 1353–1377, 2004.
- [34] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, "Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise," *Signal Process.*, vol. 91, no. 2, pp. 163–175, 2011.
- [35] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Ser. B (Methodol.)*, pp. 1–38, 1977.
- [36] J. D. Hamilton and R. Susmel, "Autoregressive conditional heteroskedasticity and changes in regime," *J. Econometrics*, vol. 64, pp. 307–333, 1994.
- [37] R. Shumway and D. Stoffer, "Dynamic linear models with switching," *J. Amer. Statist. Assoc.*, vol. 86, no. 415, pp. 763–769, 1991.
- [38] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.
- [39] R. F. Engle, "Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.
- [40] O. Straka and M. Šimandl, "Distance-based pruning for Gaussian sum method in non-Gaussian system state estimation," in *Proc. 8th IASTED Int. Conv. Intell. Syst. Control*, Anaheim, CA, USA, 2005, pp. 96–101.
- [41] F. Black, "Studies of stock price volatility changes," in *Proc. Meetings Amer. Statist. Assoc., Bus., Econ. Statist. Sec.*, 1976, pp. 177–181.
- [42] B. Eraker, M. Johannes, and N. Polson, "The impact of jumps in volatility and returns," *J. Finance*, vol. 58, no. 3, pp. 1269–1300, 2003.
- [43] S. Chib, F. Nardari, and N. Shephard, "Markov chain Monte Carlo methods for stochastic volatility models," *J. Econometrics*, vol. 108, no. 2, pp. 281–316, 2002.
- [44] G. B. Durham, "Monte Carlo methods for estimating, smoothing, and filtering one- and two-factor stochastic volatility models," *J. Econometrics*, vol. 133, no. 1, pp. 273–305, 2006.
- [45] G. B. Durham, "SV mixture models with application to S&P 500 index returns," *J. Financial Economics*, vol. 85, no. 3, pp. 822–856, 2007.
- [46] T. Andersen, L. Benzoni, and J. Lund, "An empirical investigation of continuous-time equity return models," *J. Finance*, vol. 57, no. 3, pp. 1239–1284, 2002.
- [47] O. Cappé, "Online EM algorithm for hidden Markov models," *J. Comput. Graph. Statist.*, vol. 20, no. 3, pp. 728–749, 2011.
- [48] E. Ozkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump Markov nonlinear system," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 754–765, Feb. 1, 2015.



Ivan Gorynin received the M.Sc. degree in statistics from Télécom SudParis, Evry, France, in 2014. He is currently pursuing the Ph.D. degree in the CITI Department, Telecom SudParis.

His research interests include term structure and stochastic volatility modeling, financial econometrics, machine learning, Big Data, non-linear time-series analysis, mathematical statistics, and hidden Markov models.



Stéphane Derrode (SM'15) was as an Associate Professor in signal and image processing first at École Nationale Supérieure de Physique de Marseille, Marseille, France, in 2001 and then at École Centrale de Marseille in 2005, France. He joined École Centrale de Lyon, Lyon, France, in 2014 where he is currently Professor in Computer Sciences within the Mathematics and IT Department. His research interests include nonlinear, non-Gaussian state-space systems for time-series analysis, with applications to UAV, object tracking in video and dynamic biometry.



Emmanuel Monfrini received the Ph.D. degree in mathematical statistics from Paris VI University, Paris, France, in 2002.

He has been an Associate Professor at Télécom SudParis, Evry, France, since 2008. His research interest is centered around statistical modeling, with a particular interest for Markovian models and their application in several fields including signal and image processing.



Wojciech Pieczynski received the Doctorat d'État degree in mathematical statistics from the Université Pierre et Marie Curie, Paris, France, in 1986.

He is currently a Professor at the Telecom SudParis, Evry, France, and Head of the Communications, Images, and Information Processing (CITI) department. His research interests include mathematical statistics, stochastic processes, and statistical image processing.