EXACT FAST SMOOTHING IN SWITCHING MODELS WITH APPLICATION TO STOCHASTIC VOLATILITY

*Ivan Gorynin**, *Stéphane Derrode*†, *Emmanuel Monfrini** *and Wojciech Pieczynski**

* Telecom SudParis,
SAMOVAR, CNRS UMR 5157,
Évry, France.

† École Centrale de Lyon,
LIRIS, CNRS UMR 5205,
Écully, France.

ABSTRACT

We consider the problem of statistical smoothing in nonlinear non-Gaussian systems. Our novel method relies on a Markov-switching model to operate recursively on series of noisy input data to produce an estimate of the underlying system state. We show through a set of experiments that our technique is efficient within the framework of the stochastic volatility model.

***Index Terms*—** Nonlinear systems, Stochastic volatility, Bayesian smoother, Conditionally Gaussian linear state-space model, Smoothing in switching systems.

1. INTRODUCTION

Let us consider two random sequences $\mathbf{X}_1^N = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ and $\mathbf{Y}_1^N = (\mathbf{Y}_1, \ldots, \mathbf{Y}_N)$, $\mathbf{X}_n$ and $\mathbf{Y}_n$ respectively taking their values in $\mathbb{R}^d$ and $\mathbb{R}^p$. $\mathbf{X}_1^N$ is hidden, while $\mathbf{Y}_1^N$ is observed. We explore the problem of "Bayesian smoothing" which aims to recover, for $n = 1, \ldots, N$, $\mathbf{X}_n$ from $\mathbf{Y}_1^N$. With the usual notations for conditional expectation, one gets the solution by computing $\mathrm{E}\left[\mathbf{X}_n \left| \mathbf{y}_1^N \right.\right]$. We provide an original technique that appears as an alternative to the widely used particle filter based methods, for example in finance [1–3]. Our method, inspired from [4] to perform a fast filtering, applies when $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is a stationnary hidden Markov model (HMM), non necessarily Gaussian nor linear. Such a stationary HMM can be defined by an invariant probability density function $p(\mathbf{x}_1, \mathbf{y}_1)$ and the following recursion equations:

$$\mathbf{X}_{n+1} = \Phi(\mathbf{X}_n, \mathbf{U}_{n+1}); \quad (1)$$

$$\mathbf{Y}_n = \Psi(\mathbf{X}_n, \mathbf{V}_{n+1}), \quad (2)$$

where $\mathbf{U}_1, \mathbf{V}_1, \ldots, \mathbf{U}_N, \mathbf{V}_N$ are convenient independent variables, and functions Φ and Ψ specify the state equation and the measurement equation. There is no known Bayesian smoothing algorithm for this general framework, and one uses various approximations [5, 6]. Among them, the Monte Carlo methods are very attractive, but they can meet difficulties in some situations. Our algorithm makes use of the smoothing in the "Conditionally Markov switching hidden linear model" (CMSHLM) [7], which can accurately fit the given HMM. In this way, we state a new general method for smoothing and show its interest within the framework of the classic stochastic volatility model [8–10].

The paper is organized as follows. In the next section, we remind the CMSHLM model and show its relevance for solving the problem. In Section 3, we describe the "Stationary conditionally Gaussian observed Markov switching model" (SCGOMSM) which is a particular Gaussian CMSHLM. Our smoothing method, which is based on the Expectation-Maximization (EM) algorithm and Gaussian CMSHLM, is given in Section 4. The fifth section presents experimental results, and the sixth one draws conclusions and perspectives.

2. EXACT SMOOTHING IN CONDITIONALLY MARKOV SWITCHING HIDDEN LINEAR MODELS

Let $\mathbf{R}_1^N = (R_1, \ldots, R_N)$ be a random sequence taking its values in $\Omega = \{1, \ldots, K\}$.

Definition 1 We say that a discrete time Markov process $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a "Conditionally Markov switching hidden linear model" (CMSHLM) if it verifies

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

and the following recursion equation:

$$\begin{aligned}\mathbf{X}_{n+1} = \mathbf{F}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})\mathbf{X}_n + \\ \mathbf{G}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})\mathbf{W}_{n+1} + \mathbf{H}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1}),\end{aligned} \quad (4)$$

with suitable matrices $\mathbf{F}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$, $\mathbf{G}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$, $\mathbf{H}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$ and Gaussian unit-variance white noise vector $\mathbf{W}_{n+1}$.

We can state the following:

Proposition 1
Let $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ be a CMSHLM. Next, for each n in $\{1, \ldots, N\}$, $p\left(r_n \left| \mathbf{y}_1^N \right.\right)$, $\mathrm{E}\left[\mathbf{X}_n \left| \mathbf{y}_1^N \right.\right]$ and $\mathrm{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathbf{y}_1^N \right.\right]$ can be computed with a complexity linear in N.

Proof Firstly, let us prove that

$$\mathrm{E}\left[\mathbf{X}_n \left|\mathbf{y}_1^N\right.\right] = \sum_{r_n} p\left(r_n \left|\mathbf{y}_1^N\right.\right) \mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right], \quad (5)$$

$$\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left|\mathbf{y}_1^N\right.\right] = \sum_{r_n} p\left(r_n \left|\mathbf{y}_1^N\right.\right) \mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left|r_n, \mathbf{y}_1^n\right.\right]. \quad (6)$$

According to (3), $p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_n^{n+1}\right.\right) = p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_n\right.\right)$ and, more generally, $p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_n^N\right.\right) = p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_n\right.\right)$. Then (5) comes from $p\left(\mathbf{x}_n \left|\mathbf{y}_1^N\right.\right) = \sum_{r_n} p\left(r_n \left|\mathbf{y}_1^N\right.\right) p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_1^N\right.\right)$, and we prove (6) in a similar way.

Secondly, the posterior marginals $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ are calculable using the forward-backward algorithm. The "forward" and "backward" probabilities $\alpha_n(r_n) = p\left(r_n, \mathbf{y}_1^n\right)$, $\beta_n(r_n) = p\left(\mathbf{y}_{n+1}^N \left|r_n, \boldsymbol{y}_n\right.\right)$ are recursively computable with

$$\begin{aligned} &\alpha_1(r_1) = p\left(r_1, \boldsymbol{y}_1\right); \\ &\alpha_{n+1}(r_{n+1}) = \sum_{r_n\in\Omega} \alpha_n(r_n) p\left(r_{n+1}, \boldsymbol{y}_{n+1} \left|r_n, \boldsymbol{y}_n\right.\right); \quad (7) \\ &\beta_N(r_N) = 1; \\ &\beta_n(r_n) = \sum_{r_{n+1}\in\Omega} \beta_{n+1}(r_{n+1}) p\left(r_{n+1}, \boldsymbol{y}_{n+1} \left|r_n, \boldsymbol{y}_n\right.\right). \quad (8) \end{aligned}$$

This algorithm enables us to calculate the jump smoothed values $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ and the jump filtered values $p\left(r_n \left|\mathbf{y}_1^n\right.\right)$ using

$$p\left(r_n \left|\mathbf{y}_1^N\right.\right) = \frac{\alpha_n(r_n)\beta_n(r_n)}{\sum_{r_n^*\in\Omega} \alpha_n(r_n^*)\beta_n(r_n^*)}. \quad (9)$$

$$p\left(r_n \left|\mathbf{y}_1^n\right.\right) = \frac{\alpha_n(r_n)}{\sum_{r_n^*\in\Omega} \alpha_n(r_n^*)}. \quad (10)$$

We can also get $\mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right]$ and $\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left|r_n, \mathbf{y}_1^n\right.\right]$ inductively [7]. Indeed, the recursive formula to compute $\mathrm{E}\left[\mathbf{X}_{n+1} \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right]$ from $\mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right]$ is

$$\begin{aligned} &\mathrm{E}\left[\mathbf{X}_{n+1} \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right] = \sum_{r_n} p\left(r_n \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) \\ &\left(\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right] + \mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\right) (11) \end{aligned}$$

The recursion to compute $\mathrm{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right]$ from $\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left|r_n, \mathbf{y}_1^n\right.\right]$ is similar to the previous one:

$$\begin{aligned} &\mathrm{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right] = \sum_{r_n} p\left(r_n \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) \\ &\Big(\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left|r_n, \mathbf{y}_1^n\right.\right] \mathbf{F}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \\ &\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right] \mathbf{H}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \\ &\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathrm{E}^\top\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right] \mathbf{F}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \\ &\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbf{G}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \\ &\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbf{H}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\Big). \quad (12) \end{aligned}$$

Both expressions require $p\left(r_n \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right)$, which is obtainable using (10) and

$$p\left(r_n \left|r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) = \frac{p\left(r_{n+1}, \mathbf{y}_{n+1} \left|r_n, \mathbf{y}_n\right.\right) p\left(r_n \left|\mathbf{y}_1^n\right.\right)}{\sum_{r_n^*} p\left(r_{n+1}, \mathbf{y}_{n+1} \left|r_n^*, \mathbf{y}_n\right.\right) p\left(r_n^* \left|\mathbf{y}_1^n\right.\right)}. \quad (13)$$

3. STATIONARY CONDITIONALLY GAUSSIAN OBSERVED MARKOV SWITCHING MODELS

Let us consider a stationary Markov triplet $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ such that $\forall n \in \{1, ..., N\}$, $p(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1} | \mathbf{r}_n^{n+1})$ is a Gaussian probability density function. Let $\mathbf{Z}_n$ be $(\mathbf{X}_n^\top, \mathbf{Y}_n^\top)^\top$ and z_n be its realisation. We thus intend to specify the mean vector

$$\boldsymbol{\Upsilon}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \mathbf{M}(r_n) \\ \mathbf{M}(r_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathrm{E}\left[\mathbf{Z}_n \left|r_n\right.\right] \\ \mathrm{E}\left[\mathbf{Z}_{n+1} \left|r_{n+1}\right.\right] \end{bmatrix} \quad (14)$$

and the covariance matrix

$$\boldsymbol{\Xi}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \boldsymbol{S}(r_n) & \boldsymbol{\Sigma}(\mathbf{r}_n^{n+1}) \\ \boldsymbol{\Sigma}^\top(\mathbf{r}_n^{n+1}) & \boldsymbol{S}(r_{n+1}) \end{bmatrix} \quad (15)$$

of the multivariate normal distribution:

$$\begin{aligned} &p(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1} | \mathbf{r}_n^{n+1}) = \\ &\qquad \mathcal{N}\left(\left(\mathbf{z}_n^\top, \mathbf{z}_{n+1}^\top\right), \boldsymbol{\Upsilon}(\mathbf{r}_n^{n+1}), \boldsymbol{\Xi}(\mathbf{r}_n^{n+1})\right), \quad (16) \end{aligned}$$

Definition 2 We say that a discrete time stationary Markov process $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a "Stationary conditionally Gaussian observed Markov switching model" (SCGOMSM) if it verifies (14)-(16) and has the following property:

$$p\left(\mathbf{y}_{n+1} \left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left|\mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right) \quad (17)$$

Let us remember that SCGOMSMs can be very close to the classic "conditionally Gaussian linear state-space model" (CGLSSM) [11, 12], which does not offer the possibility of fast smoothing [5].

Proposition 2
If a discrete time stationary Markov process $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a SCGOMSM, it is also a CMSHLM with $\mathbf{F}_{n+1}$, $\mathbf{G}_{n+1}$ and $\mathbf{H}_{n+1}$ in (4) given by (26)-(28).
Proof According to (14) - (16), we have $p\left(r_{n+1} \left|\mathbf{x}_n, r_n, \mathbf{y}_n\right.\right) = p\left(r_{n+1} \left|r_n\right.\right)$. We then use (17) to prove that a SCGOMSM has property (3) of the CMSHLM.

To find out the corresponding $\mathbf{F}_{n+1}$, $\mathbf{G}_{n+1}$ and $\mathbf{H}_{n+1}$ in (4), we set

$$\mathbf{A}(\mathbf{r}_n^{n+1}) = \boldsymbol{\Sigma}^\top(\mathbf{r}_n^{n+1})\, \boldsymbol{S}^{-1}(r_n), \quad (18)$$

and consider $\mathbf{B}(\mathbf{r}_n^{n+1})$ and $\mathbf{Q}(\mathbf{r}_n^{n+1})$ such that

$$\begin{aligned} &\mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{B}^\top(\mathbf{r}_n^{n+1}) = \\ &\qquad \boldsymbol{S}(r_{n+1}) - \boldsymbol{\Sigma}^\top(\mathbf{r}_n^{n+1})\boldsymbol{S}^{-1}(r_n)\boldsymbol{\Sigma}(\mathbf{r}_n^{n+1}), \quad (19) \end{aligned}$$

$$\mathbf{Q}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \boldsymbol{Q}_1(\mathbf{r}_n^{n+1}) & \boldsymbol{Q}_2(\mathbf{r}_n^{n+1}) \\ \boldsymbol{Q}_3(\mathbf{r}_n^{n+1}) & \boldsymbol{Q}_4(\mathbf{r}_n^{n+1}). \end{bmatrix} = \mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{B}^\top(\mathbf{r}_n^{n+1}). \tag{20}$$

Equation (17) induces that the matrix $\mathbf{A}(\mathbf{r}_n^{n+1})$ has the following form:

$$\mathbf{A}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \boldsymbol{A}_1(\mathbf{r}_n^{n+1}) & \boldsymbol{A}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{0} & \boldsymbol{A}_4(\mathbf{r}_n^{n+1}) \end{bmatrix}. \tag{21}$$

Hence, we can state that the discrete time process $(\mathbf{Z}_1^N)$ satisfies the following recursion equation:

$$\mathbf{Z}_{n+1} = \mathbf{A}(\mathbf{R}_n^{n+1})(\mathbf{Z}_n - \mathbf{M}(R_n)) \\ + \mathbf{B}(\mathbf{R}_n^{n+1})\mathbf{W}_{n+1} + \mathbf{M}(R_{n+1}), \tag{22}$$

where $\mathbf{W}_1, \ldots, \mathbf{W}_N$ are Gaussian unit-variance white noise vectors. Let us set $\mathbf{M}(r_n) = \left(\mathbf{M}_1(r_n)^\top, \mathbf{M}_2(r_n)^\top\right)^\top$

$p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right)$ is a multivariate normal probability density function. Its covariance matrix is $\mathbf{Q}(\mathbf{r}_n^{n+1})$ and its mean vector is

$$\mathbf{A}(\mathbf{r}_n^{n+1}) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix} + \begin{bmatrix} \mathbf{N}_1(\mathbf{r}_n^{n+1}) \\ \mathbf{N}_2(\mathbf{r}_n^{n+1}) \end{bmatrix} \\ = \begin{bmatrix} \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{x}_n + \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_1(\mathbf{r}_n^{n+1}) \\ \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_2(\mathbf{r}_n^{n+1}) \end{bmatrix}, \tag{23}$$

where we set

$$\mathbf{N}_1(\mathbf{r}_n^{n+1}) = \mathbf{M}_1(r_{n+1}) - \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{M}_1(r_n) \\ - \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{M}_2(r_n), \\ \mathbf{N}_2(\mathbf{r}_n^{n+1}) = \mathbf{M}_2(r_{n+1}) - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{M}_2(r_n).$$

$p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1} \right.\right)$ is also a multivariate normal probability density function with mean vector is

$$\boldsymbol{Q}_2(\mathbf{r}_n^{n+1})\boldsymbol{Q}_4^{-1}(\mathbf{r}_n^{n+1})(\mathbf{y}_{n+1} - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n - \mathbf{N}_2(\mathbf{r}_n^{n+1})) \\ + \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{x}_n + \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_1(\mathbf{r}_n^{n+1}), \tag{24}$$

and covariance matrix

$$\boldsymbol{Q}_1(\mathbf{r}_n^{n+1}) - \boldsymbol{Q}_2(\mathbf{r}_n^{n+1})\boldsymbol{Q}_4^{-1}(\mathbf{r}_n^{n+1})\boldsymbol{Q}_3(\mathbf{r}_n^{n+1}). \tag{25}$$

This allows to complete the proof and to specify $\mathbf{F}_{n+1}$, $\mathbf{G}_{n+1}$ and $\mathbf{H}_{n+1}$:

$$\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{A}_1(\mathbf{r}_n^{n+1}), \tag{26}$$

$$\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_1(\mathbf{r}_n^{n+1}) + \tag{27}$$

$$\boldsymbol{Q}_2(\mathbf{r}_n^{n+1})\boldsymbol{Q}_4^{-1}(\mathbf{r}_n^{n+1})(\mathbf{y}_{n+1} - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n - \mathbf{N}_2(\mathbf{r}_n^{n+1})),$$

$$\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\mathbf{G}_{n+1}^T(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = (25). \tag{28}$$

To sum up, the smoothing procedure in the SCGOMSM at first recursively computes $p\left(r_n \left| \mathbf{y}_1^n \right.\right)$, $\mathrm{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n \right.\right]$, $\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left| r_n, \mathbf{y}_1^n \right.\right]$ using Algorithm 1 below, then computes the state smoothed values using (8) and (9) and finally produces the smoothed output using (5) and (6).

Algorithm 1 Given $\mathrm{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n \right.\right]$, $\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^\top \left| r_n, \mathbf{y}_1^n \right.\right]$, $p\left(r_n \left| \mathbf{y}_1^n \right.\right)$, and $\mathbf{y}_{n+1}$:

- Calculate $\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, $\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$ and $\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\mathbf{G}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$ with (26)-(28);
- Calculate $p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) = p\left(r_{n+1} \left| r_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right)$. Let us remind that the probability density function $p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right)$ is multivariate normal with mean vector $\mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_2(\mathbf{r}_n^{n+1})$ and covariance matrix $\boldsymbol{Q}_4(\mathbf{r}_n^{n+1})$;
- Use (7) and (10) to calculate $p\left(r_{n+1} \left| \mathbf{y}_1^{n+1} \right.\right)$, then use (13) to calculate $p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right)$;
- Use (11) and (12) to calculate $\mathrm{E}\left[\mathbf{X}_{n+1} \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right]$ and $\mathrm{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right]$.

4. APPROXIMATING NON-LINEAR NON-GAUSSIAN MODELS

Let us consider a stationary HMM that we described in the introduction. The probability density functions $p\left(\mathbf{x}_n^{n+1}\right)$ and $p\left(\mathbf{y}_n \left| \mathbf{x}_n \right.\right)$ do not depend on n, so $p\left(\mathbf{x}_1^2, \mathbf{y}_1^2\right)$ defines the entire distribution of $p\left(\mathbf{x}_1^N, \mathbf{y}_1^N\right)$. The idea is to approach it by a Gaussian mixture of K^2 components in such a way that this mixture derives from a SCGOMSM [13].

More precisely, it is possible to find an approximation

$$p\left(\mathbf{x}_1^2, \mathbf{y}_1^2\right) \approx \sum_{1 \le i,j, \le K} c_{ij}\, p_{ij}(\mathbf{x}_1^2, \mathbf{y}_1^2), \tag{29}$$

with an arbitrary precision and under mild conditions. In order to find a SCGOMSM close to the given HMM, we assume that c_{ij} is the distribution of the random variable (R_1, R_2) taking its values in $\Omega^2 = \{1, ..., K\}^2$, which means that we set $c_{ij} = P(R_1 = i, R_2 = j)$. With respect to (16), we can then parametrize the conditional distribution $p_{ij}(\mathbf{x}_1^2, \mathbf{y}_1^2) = p(\mathbf{x}_1^2, \mathbf{y}_1^2 | r_1 = i, r_2 = j)$ by the mean vector $\boldsymbol{\Upsilon}_{ij}$ and the covariance matrix $\boldsymbol{\Xi}_{ij}$. The suitable values for c_{ij}, $\boldsymbol{\Upsilon}_{ij}$ and $\boldsymbol{\Xi}_{ij}$ can be found with a training sample $(\mathbf{x}_1'^M, \mathbf{y}_1'^M)$ simulated within the HMM framework, and the Expectation-Maximization algorithm (EM) that we describe below.

Algorithm 2 inputs the simulated sample to produce the sequence $\left(c_{ij}^{(q)}, \boldsymbol{\Upsilon}_{ij}^{(q)}, \boldsymbol{\Xi}_{ij}^{(q)}\right)_{q \in \mathbb{N}, 1 \le i,j \le K}$, assumed to tend to $(c_{ij}, \boldsymbol{\Upsilon}_{ij}, \boldsymbol{\Xi}_{ij})_{1 \le i,j \le K}$.

Algorithm 2

1. Let $q \leftarrow 0$. Use the k-means clustering algorithm to assign each couple $(\mathbf{x}'_m, \mathbf{y}'_m)$ to one of the K clusters and consider, for each $1 \le i \le K$ and $1 \le m \le M$, $\phi_m^{(0)}(i)$ that equals 1 if $(\mathbf{x}'_m, \mathbf{y}'_m)$ belongs to the i-th cluster and 0 otherwise. Furthermore, consider, for each $1 \le i, j \le K$ and $1 \le m < M$, $\psi_m^{(0)}(i, j) = \phi_m^{(0)}(i)\phi_{m+1}^{(0)}(j)$;
2. Use the formulas below to calculate, for each $1 \le i, j \le$

K, the new estimates $\left(c_{ij}^{(q+1)}, \boldsymbol{\Upsilon}_{ij}^{(q+1)}, \boldsymbol{\Xi}_{ij}^{(q+1)}\right)$.

$$c_{ij}^{(q+1)} = \frac{1}{M-1}\sum_{m=1}^{M-1} \psi_m^{(q)}(i,j), \tag{30}$$

$$\boldsymbol{\Upsilon}_{ij}^{(q+1)} = \frac{\sum_{m=1}^{M-1} \mathbf{z}'^{m+1}_m \psi_m^{(q)}(i,j)}{\sum_{m=1}^{M-1} \psi_m^{(q)}(i,j)}, \tag{31}$$

$$\boldsymbol{\Xi}_{ij}^{(q+1)} = \frac{\sum_{m=1}^{M-1} \left(\mathbf{z}'^{m+1}_m - \boldsymbol{\Upsilon}_{ij}^{(q+1)}\right)\left(\mathbf{z}'^{m+1}_m - \boldsymbol{\Upsilon}_{ij}^{(q+1)}\right)^T \psi_m^{(q)}(i,j)}{\sum_{m=1}^{M-1} \psi_m^{(q)}(i,j)}, \tag{32}$$

where $\mathbf{z}'^{m+1}_m = (\mathbf{x}'^T_m, \mathbf{y}'^T_m, \mathbf{x}'^T_{m+1}, \mathbf{y}'^T_{m+1})^T$.

3. To compute $\psi_m^{(q+1)}(i,j)$, use (33) to compute the "forward" probabilities and use (34) to compute the "backward" ones, then use (35) to calculate $\psi_m^{(q+1)}(i,j)$ for each $1 \le i,j \le K$ and $1 \le m < M$.

$$\begin{aligned} \alpha_1(i) &= a_i p\left(\mathbf{x}'_1, \mathbf{y}'_1 \left| r_1 = i\right.\right); \\ \alpha_{m+1}(j) &= \sum_{i\in\Omega} \alpha_m(i)\gamma_m^{(q+1)}(i,j)p_{j|i}; \end{aligned} \tag{33}$$

$$\begin{aligned} \beta_M(j) &= 1; \\ \beta_m(i) &= \sum_{j\in\Omega} \beta_{m+1}(j)\gamma_m^{(q+1)}(i,j)p_{j|i}. \end{aligned} \tag{34}$$

$$\psi_m^{(q+1)}(i,j) = \frac{\alpha_m(i)\gamma_m^{(q+1)}(i,j)p_{j|i}\beta_{m+1}(j)}{\sum_{i^*,j^*\in\Omega} \alpha_m(i^*)\gamma_m^{(q+1)}(i^*,j^*)p_{j^*|i^*}\beta_{m+1}(j^*)}, \tag{35}$$

where $a_i = \sum_{j\in\Omega} c_{ij}^{(q+1)}$, $p_{j|i} = \frac{c_{ij}^{(q+1)}}{a_i}$ and $\gamma_m^{(q+1)}(i,j) = p\left(\mathbf{x}'_{m+1}, \mathbf{y}'_{m+1} \left| r_m = i, r_{m+1} = j, \mathbf{x}'_m, \mathbf{y}'_m\right.\right)$. The conditional probability density functions $p\left(\mathbf{x}'_1, \mathbf{y}'_1 \left| r_1 = i\right.\right)$ and $p\left(\mathbf{x}'_{m+1}, \mathbf{y}'_{m+1} \left| r_m = i, r_{m+1} = j, \mathbf{x}'_m, \mathbf{y}'_m\right.\right)$ are multivariate normal distribution defined by $\boldsymbol{\Upsilon}_{ij}^{(q+1)}$ and $\boldsymbol{\Xi}_{ij}^{(q+1)}$ with respect to (14)-(23).

4. Stop according to a stopping criterion or increment q and repeat from Step 2.

Finally, our new smoothing method is:

Algorithm 3 Given input data $\mathbf{y}_1^N$,

1. Consider a training sample $(\mathbf{x}'^M_1, \mathbf{y}'^M_1)$ simulated within the HMM framework;
2. Apply Algorithm 2 to the training sample to compute $(c_{ij}, \boldsymbol{\Upsilon}_{ij}, \boldsymbol{\Xi}_{ij})_{1\le i,j\le K}$;
3. Compute recursively $p\left(r_n \left|\mathbf{y}_1^n\right.\right)$, $\mathrm{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right]$, $\mathrm{E}\left[\mathbf{X}_n\mathbf{X}_n^T \left|r_n, \mathbf{y}_1^n\right.\right]$ using Algorithm 1;
4. Calculate state smoothed values using (8) and (9);
5. Calculate the smoothed output using (5) and (6);

5. EXPERIMENTS

Let us consider the stochastic volatility model defined by the following recursion equations:

$$\mathbf{X}_1 = \mathbf{U}_1; \tag{36}$$

$$\mathbf{X}_{n+1} = \mu + \phi(\mathbf{X}_n - \mu) + \sigma\mathbf{U}_{n+1}; \tag{37}$$

$$\mathbf{Y}_n = \beta\exp\left(\mathbf{X}_n/2\right)\mathbf{V}_n, \tag{38}$$

with fixed μ and σ, and Gaussian unit-variance white noise vectors $\mathbf{U}_1, \mathbf{V}_1, \ldots, \mathbf{U}_N, \mathbf{V}_N$. In the non-degenerate case, this model is stationary if and only if $\phi < 1$ and $\frac{\sigma^2}{1-\phi^2} = 1$ [14]. Therefore, we consider four sets of parameters, corresponding to $\mu = 0.5$, $\beta = 0.5$, different values of σ^2 and ϕ with respect to the condition $\sigma^2 = 1 - \phi^2$.

For each experiment instance, we aim to compare the behaviour of our method for $K = 2, 3, 5$ or 7. In concrete terms, we simulate a training sample $(\mathbf{x}'^M_1, \mathbf{y}'^M_1)$ and a test sample $(\mathbf{x}_1^N, \mathbf{y}_1^N)$ within the framework of the stochastic volatility model. Next, we use Algorithm 3 to compute a smoothed output, and we use $\mathbf{x}_1^N$ to calculate the mean square error (MSE). For our study cases, the training sample size is $M = 20000$, the number of iterations of EM is $Q_{EM} = 100$ and the test sample size is $N = 1000$.

We also use a particle smoother to compute $\mathrm{E}\left[\mathbf{X}_n \left|\mathbf{Y}_1^{n+T}\right.\right]$ with $T = 5$, and we measure the MSE. We found out that using greater values of T needs more particles to cope with the degeneracy phenomenon, but does not change the MSE value. We thus consider that $\mathrm{E}\left[\mathbf{X}_n \left|\mathbf{Y}_1^{n+T}\right.\right]$ is a good approximation of $\mathrm{E}\left[\mathbf{X}_n \left|\mathbf{Y}_1^{N}\right.\right]$.

The results of our experiments are presented in Table 1. In the case of $K = 5$, the results are similar to those obtained with a particle smoother (PS). When choosing 100 particles, PS is quicker than our method because of the EM algorithm. However, when the underlying SCGOMSM fits the HMM, our method is as fast as the Kalman filter, and faster than PS. For the higher values of K, the results remain stable and analogous to the PS ones, which probably means that they are close to the theoretical ones.

6. CONCLUSION

We put forward a new method to find the hidden signal in the framework of a nonlinear and non-Gaussian model. This method is general and works under slight conditions: we only need to be able to sample data according to the given HMM. When our model fits the nonlinear and non-Gaussian system,

Table 1. MSE of 100 separate experiments in the cases described above, for our method and for the particle smoother (1500 particles).

			Nb of mixture components				
	ϕ	σ^2	2	3	5	7	PS
1	0.99	0.02	0.39	0.23	0.14	0.13	0.12
2	0.90	0.19	0.48	0.39	0.35	0.34	0.33
3	0.80	0.36	0.56	0.50	0.47	0.47	0.46
4	0.50	0.75	0.70	0.67	0.66	0.66	0.66

the method is as fast as the classic Kalman filter in linear systems.

We tested our approach in the framework of the stochastic volatility model, and it turns out that the mean square error obtained is very close to the theoretical one, the latter estimated by a particle smoother.

As a conclusion, let us mention two perspectives. The first one is to consider different and more complex stochastic volatility models [3, 15–17]; the second one is to consider more advanced families of switching models allowing fast exact smoothing.

REFERENCES

[1] 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.

[2] B. Eraker, "Do stock prices and volatility jump? Reconciling evidence from spot and option prices," *The Journal of Finance*, vol. 59, pp. 1367–1404, 2004.

[3] C.-J. Kim and C. R. Nelson, *State-space models with regime switching*, MIT Press, 1999.

[4] 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, June 2011, pp. 290–294.

[5] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag, 2005.

[6] A. Doucet and A. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later*, Eds. London, U.K., Oxford Univ. Press, 2011.

[7] W. Pieczynski, "Exact filtering in conditionally Markov switching hidden linear models," *Comptes Rendus Mathématique*, vol. 349, no. 9-10, pp. 587–590, 2011.

[8] E. Ghysels, A. Harvey, and E. Renault, "Stochastic volatility," *Handbook of Statistics*, vol. 14, pp. 119–192, 1995.

[9] E. Jacquier, N. G. Polson, and P. Rossi, "Bayesian analysis of stochastic volatility models," *Journal of Business & Economic Statistics*, vol. 12, no. 4, pp. 371–389, 1994.

[10] N. Shephard, S. Kim, and S. Chib, "Stochastic volatility: likelihood inference and comparison with ARCH models," *Review of Economic Studies*, vol. 65, no. 3, pp. 361–393, 1998.

[11] S. Derrode and W. Pieczynski, "Exact fast computation of optimal filter in Gaussian switching linear systems," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 701–704, July 2013.

[12] Y. Petetin and F. Desbouvries, "A class of fast exact Bayesian filters in dynamical models with jumps," *IEEE Trans. on Signal Processing*, vol. 62, no. 14, pp. 3643–3653, 2014.

[13] S. Derrode and W. Pieczynski, "Fast filter in nonlinear systems with application to stochastic volatility model," in *Proc. of the EUSIPCO'14*, Lisbon, Portugal, Sept. 2014, pp. 290–294.

[14] Yu Meng, "Bayesian analysis of a stochastic volatility model," 2009.

[15] N. Y. Nikolaev, E. Smirnov, and L. M. de Menezes, "Nonlinear filtering of asymmetric stochastic volatility models and value-at-risk estimation," in *Proc. of the IEEE Conf. Computational Intelligence for Financial Engineering and Economics (CIFEr-2014)*, London, U.K., 2014, pp. 310–317.

[16] Y. Omori and T. Watanabe, "Block sampler and posterior mode estimation for asymmetric stochastic volatility models," *Computational Statistics and Data Analysis*, vol. 52, no. 6, pp. 2892–2910, 2008.

[17] M. Takashi, Y. Omori, and T. Watanabe, "Estimating stochastic volatility models using daily returns and realized volatility simultaneously," *Computational Statistics and Data Analysis*, vol. 53, no. 6, pp. 2404–2426, 2009.