Parameter estimation in switching Markov systems and unsupervised smoothing

Fei Zheng, Stéphane Derrode, and Wojciech Pieczynski

***Abstract*—Stationary Jump Markov Linear Systems (JMLSs) model linear systems whose parameters evolve with time according to a hidden finite state Markov chain. We propose an algorithm for parameter estimation of a recent class of JMLSs called Conditionally Gaussian Pairwise Markov Switching Models (CGPMSMs). Our algorithm, named Double-EM (DEM), is based on the Expectation-Maximization (EM) principle applied twice sequentially. The first EM is applied to the couple (switches, observations) temporarily assumed to be a Pairwise Markov Chain (PMC). The second one is used to estimate the remaining conditional transitions and conditional noise matrices of the CGPMSM. The efficiency of the proposed algorithm is studied via unsupervised smoothing on simulated data. In particular, smoothing results, produced with CGPMSM in an unsupervised manner using DEM, can be more efficient than the ones obtenied with the classic "Conditionally Gaussian Linear State-Space Model" (CGLSSM) based on true parameters and true switches.**

***Index Terms*—Markov switching linear systems, Expectation-Maximization, CGPMSM, DEM-CGPMSM, Parameter estimation, Unsupervised smoothing.**

I. Introduction

Let us consider three random sequences $\mathbf{X}_1^N = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, $\mathbf{R}_1^N = \{R_1, \ldots, R_N\}$, and $\mathbf{Y}_1^N = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$, with each $\mathbf{X}_n$, R_n, $\mathbf{Y}_n$ take their values in $\mathbb{R}^m$, $\mathbf{\Omega} = \{1, \ldots, K\}$, and $\mathbb{R}^q$ respectively. $\mathbf{Y}_1^N$ are observed while $\mathbf{R}_1^N$ and $\mathbf{X}_1^N$ are not. The task that we are handling is to estimate the hidden states $\mathbf{X}_1^N$ from only the observed $\mathbf{Y}_1^N$. We will use the Bayesian smoothing method, which consists of estimating $\mathbf{x}_1^N = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$ by $\hat{\mathbf{x}}_1^N = (\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_N)$, where each $\hat{\mathbf{x}}_n$ is given by the conditional expectation:

$$\hat{\mathbf{x}}_n = \mathbb{E}\left[\mathbf{X}_n \middle| \mathbf{y}_1^N\right]. \tag{1}$$

Such fixed-interval smoothing of switching Markov models is of interest in different situations. For instance interference suppression of spread spectrum CDMA systems [1], detection of Bernoulli–Gaussian processes with applications to seismic signals processing [2], target tracking and trajectory reconstruction [3], or still stochastic volatility estimation [4].

To model the probabilistic dependences among $\mathbf{X}_1^N$, $\mathbf{R}_1^N$ and $\mathbf{Y}_1^N$, we adopt the "Conditionally Gaussian Pairwise Markov Switching Models" (CGPMSMs [5]). The aim of this paper is to propose a parameter estimation method for CGPMSMs from only observations $\mathbf{Y}_1^N$ and to study unsupervised smoothing under the model with the estimated parameters. As the widely used classic "Conditionally Gaussian Linear State-Space Models" (CGLSSMs [2], [6]–[10]) are actually particular CGPMSMs, the proposed method can be applied to the CGLSSMs based unsupervised smoothing as well.

Parameter estimation problem is crucial in real applications. Early work [8] developed a pseudo-EM for estimating the parameters of dynamic linear models with switches, in which the process $\mathbf{R}_1^N$ is not a Markov chain but a non-stationary independent process defined only by time varying probabilities. In [11] author proposed a parameter identification method through matching the empirical statistics computed from data with corresponding statistics predicted by the model. In [12] author worked only on the EM estimation of transition probabilities of $\mathbf{R}_1^N$ for CGLSSMs. And later, when it comes to non-linear case, In [13] EM has been combined with particle filter to perform the parameter estimation. In this paper, particle filtering methods are not used, which removes the inherent problems like particle degeneracy or large deal of particles needed when the state space dimension is high [14]. Our contribution is twofold:

1) we propose a stationary CGPMSM parameters estimation method, called "Double-EM" (DEM) from only the observations $\mathbf{Y}_1^N = \mathbf{y}_1^N$;
2) parameters estimated with DEM are used to perform unsupervised smoothing by two fast approaches: one new defined directly from the DEM-CGPMSM, and another one based on "Conditionally Gaussian Observed Markov Switching Model" (CGOMSM [4], [5], [15]–[17]) associated with the estimated CGPMSM.

The proposed parameter estimation method opens ways to unsupervised processing (smoothing, filtering, prediction, *etc.*) in a very general framework. Indeed, CGPMSM can approximate any stationary (or even asymptotically stationary) Markov non-Gaussian non-linear systems [4], [15].

The remaining of the paper is organized as follows. Stationary CGPMSMs are recalled, and the new DEM algorithm is proposed in Section II. Two related unsupervised smoothing approaches are described in Section III, and two series of experiment on DEM based unsupervised smoothing are reported in Section IV regarding various factors of influence. Finally, the last section V draws the conclusion and presents some perspectives.

II. Stationary CGPMSM

The proposed parameter estimation method is valid under the stationary CGPMSMs model, defined by:

(i) the triplet $\mathbf{T}_1^N$, with $\mathbf{T}_n := (\mathbf{X}_n, R_n, \mathbf{Y}_n)$ for each $n = 1, \cdots, \mathbf{N}$, is Markovian;
(ii) for each $n = 1, \cdots, \mathbf{N} - 1$,

$$p\left(r_{n+1} \left| \mathbf{x}_n, r_n, \mathbf{y}_n \right.\right) = p\left(r_{n+1} \left| r_n \right.\right) \tag{2}$$

(which implies that $\mathbf{R}_1^N$ is Markovian);
(iii) $p\left(\mathbf{x}_1, \mathbf{y}_1 \left| r_1 \right.\right)$ is Gaussian, and for each $n = 1, \cdots, \mathbf{N} - 1$,

$$\begin{bmatrix} \mathbf{X}_{n+1} - \mathbf{M}^{\mathbf{x}}(R_{n+1}) \\ \mathbf{Y}_{n+1} - \mathbf{M}^{\mathbf{y}}(R_{n+1}) \end{bmatrix} := \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{F}}^{\mathbf{xx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}(\mathbf{R}_n^{n+1}) \\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(\mathbf{R}_n^{n+1}) \end{bmatrix}}_{\boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})} \begin{bmatrix} \mathbf{X}_n - \mathbf{M}^{\mathbf{x}}(R_n) \\ \mathbf{Y}_n - \mathbf{M}^{\mathbf{y}}(R_n) \end{bmatrix} + \begin{bmatrix} \mathbf{U}_{n+1} \\ \mathbf{V}_{n+1} \end{bmatrix}, \tag{3}$$

Fei Zheng and Stéphane Derrode are with Université de Lyon, Centre National de la Recherche Scientifique, École Centrale de Lyon, LIRIS, CNRS UMR 5205, 36, Av. Guy de Collongue, 69134 Écully, France. e-mail: *fzheng@ec-lyon.fr*, *stephane.derrode@ec-lyon.fr*. Wojciech Pieczynski is with Telecom SudParis, SAMOVAR, CNRS UMR 5157, Université Paris-Saclay, 9, Rue Charles Fourier, 91011 Evry Cedex, France. e-mail: *wojciech.pieczynski@telecom-sudparis.eu*

where $\boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})$ is the appropriate system transition matrix. $\mathbf{M}^{\mathbf{x}}(R_n)$ and $\mathbf{M}^{\mathbf{y}}(R_n)$ denote the means of $\mathbf{X}_n$ and $\mathbf{Y}_n$ conditionally on R_n, and $\left[\mathbf{U}_{n+1}^{\intercal}, \mathbf{V}_{n+1}^{\intercal}\right]^{\intercal}$ represents the Gaussian zero-mean noise with variance-covariance matrix independent from $\mathbf{T}_1^N$ that

$$cov\left(\begin{bmatrix}\mathbf{U}_{n+1}\\ \mathbf{V}_{n+1}\end{bmatrix}\right) := \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{Q}}^{\mathbf{xx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}(\mathbf{R}_n^{n+1})\\ \boldsymbol{\mathcal{Q}}^{\mathbf{yx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}(\mathbf{R}_n^{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{Q}}(\mathbf{R}_n^{n+1})}. \quad (4)$$

For simplification, we set

$$\mathbf{Z}_n := \begin{bmatrix}\mathbf{X}_n\\ \mathbf{Y}_n\end{bmatrix}, \mathbf{M}^{\mathbf{z}}(R_n) := \begin{bmatrix}\mathbf{M}^{\mathbf{x}}(R_n)\\ \mathbf{M}^{\mathbf{y}}(R_n)\end{bmatrix}, \mathbf{W}_n := \begin{bmatrix}\mathbf{U}_n\\ \mathbf{V}_n\end{bmatrix}, \quad (5)$$

so that, (3) can be concisely written as

$$\mathbf{Z}_{n+1} - \mathbf{M}^{\mathbf{z}}(R_{n+1}) = \boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})\left(\mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}(R_n)\right) + \mathbf{W}_{n+1}; \quad (6)$$

(iv) $\mathbf{R}_1^N$ is stationary ($p\left(\mathbf{r}_n^{n+1}\right)$ does not depend on n), and for each $n = 1, \cdots, N-1$ and each $\mathbf{r}_n^{n+1}$ in $\boldsymbol{\Omega}^2$:

$$\boldsymbol{\Gamma}^{\mathbf{z}}(r_{n+1}) = \boldsymbol{\mathcal{F}}(\mathbf{r}_n^{n+1})\boldsymbol{\Gamma}^{\mathbf{z}}(r_n)\boldsymbol{\mathcal{F}}^{\intercal}(\mathbf{r}_n^{n+1}) + \boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1}), \quad (7)$$

where $\boldsymbol{\Gamma}^{\mathbf{z}}(r_n) := \mathbb{E}\left[\left(\mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}(r_n)\right)\left(\mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}(r_n)\right)^{\intercal} | R_n = r_n\right]$ is the variance-covariance matrix of $\mathbf{Z}_n$ conditionally on $\mathbf{R}_n^{n+1} = \mathbf{r}_n^{n+1}$.

In stationary CGPMSM (i)-(iv), the distributions $p\left(\mathbf{t}_n, \mathbf{t}_{n+1}\right)$ do not depend on n, thus, the model distribution is defined by $p\left(\mathbf{t}_1, \mathbf{t}_2\right)$, given by $p\left(r_1, r_2\right)$ and $p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2 | r_1, r_2\right)$.

Let us recall two particular CGPMSMs (if stationary):

1) The classic "Conditionally Gaussian Linear State-Space Models" (CGLSSMs) are CGPMSMs with $\boldsymbol{\mathcal{F}}^{\mathbf{xy}}(\mathbf{r}_1^2) = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(\mathbf{r}_1^2) = \mathbf{0}$ for each $\mathbf{r}_1^2 \in \boldsymbol{\Omega}^2$;
2) The recent "Conditionally Gaussian Observed Markov Switching Models" (CGOMSMs), which are CGPMSMs with $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{r}_1^2) = \mathbf{0}$ for each $\mathbf{r}_1^2 \in \boldsymbol{\Omega}^2$.

Comparing to CGLSSMs, CGOMSMs have the advantage that fast exact restorations (filtering and smoothing) are feasible in spite of the existence of unknown switches [5] (CGOMSMs also belongs to the "Conditionally Markov Switching Hidden Linear Models" (CMCHLMs) family which allows fast exact restorations in general context, not necessarily to be Gaussian [18]).

The proposed parameter estimation method for the stationary CGPMSMs (i)-(iv) is constructed by the following two steps:

(a) Assuming that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is a stationary Pairwise Markov Chain (PMC [19]), apply an extension of the classic EM [20], [21] to estimate the parameters $\boldsymbol{\theta}^{\star}$ of its distribution $p^{\star}\left(\mathbf{r}_1^N, \mathbf{y}_1^N\right)$ from $\mathbf{Y}_1^N = \mathbf{y}_1^N$, then use $p^{\star}\left(\mathbf{r}_1^N, \mathbf{y}_1^N\right)$ to estimate $\left(\hat{\mathbf{r}}^{\star}\right)_1^N$ of $\mathbf{r}_1^N$ with Maximum Posterior Mode (MPM) criterion (see Subsection II-A);
(b) Estimate the parameters $\boldsymbol{\theta}$ of $p\left(\mathbf{T}_1^N\right)$ using $\left(\left(\hat{\mathbf{r}}^{\star}\right)_1^N, \mathbf{y}_1^N\right)$ with the proposed "switching EM" specified in Subsection II-B.

Such a method will be called "Double-EM" (DEM) from the application of EM principle twice. Let us insist on the fact that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is not Markov in general CGPMSM and thus the point (a) above is a temporal approximation. Once $\left(\hat{\mathbf{r}}^{\star}\right)_1^N$ is found, this hypothesis is no longer necessary. To be more precise, we have:

- $\mathbf{R}_1^N$ is Markovian in CGPMSMs, CGLSSMs, and CGOMSMs;
- $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is Markovian in CGOMSMs, but it is not necessarily Markovian in CGPMSMs;
- $\left(\mathbf{R}_1^N, \mathbf{X}_1^N\right)$ is Markovian in CGLSSMs, but it is not necessarily Markovian in CGPMSMs.

More generally, let $\left(\mathbf{G}_1^N, \mathbf{H}_1^N\right)$ be a stationary time-reversible Markov pairwise process. Necessary and sufficient conditions for markovianity of $\mathbf{G}_1^N$ (or $\mathbf{H}_1^N$) can be seen in [22].

Remark 2.1 As $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is not Markov in CGPMSM, considering it Markovian in the first EM could appear as a possible strong approximation. So we performed some experiments to assess how replacing the true $\mathbf{R}_1^N = \boldsymbol{r}_1^N$ with the estimated one obtained with the first EM (assuming $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ Markov). Experiments show that the approximation does not deteriorate the results significantly, and the method so obtained is the best one (except the optimal one) among all other methods studied.

Let us detail steps (a) and (b) above.

A. EM for Pairwise Markov Chain

The parameters $\boldsymbol{\theta}^{\star}$ to be estimated are, for each $j, k \in \boldsymbol{\Omega}$, $p_{j,k} := p^{\star}\left(r_1 = j, r_2 = k\right)$; mean $\mathbf{M}_{j,k}^{\mathbf{y}_1^2}$ and variance $\boldsymbol{\Gamma}_{j,k}^{\mathbf{y}_1^2}$ of Gaussian density $f_{j,k}\left(\mathbf{y}_1^2\right) := p^{\star}\left(\mathbf{y}_1^2 | r_1 = j, r_2 = k\right)$. According to the EM principle, to iteratively get the next parameters $\left(\boldsymbol{\theta}^{\star}\right)^{(i+1)}$ from the current ones $\left(\boldsymbol{\theta}^{\star}\right)^{(i)}$, we need to compute $\psi_n\left(j,k\right) := p^{\star}\left(r_n = j, r_{n+1} = k \left| \mathbf{y}_1^N\right.\right)$ by

$$\psi_n\left(j,k\right) = \frac{\alpha_n\left(j\right) p\left(r_{n+1} = k, \mathbf{y}_{n+1} \left| r_n = j, \mathbf{y}_n\right.\right) \beta_{n+1}\left(k\right)}{\sum\limits_{(l_1,l_2)\in\boldsymbol{\Omega}^2} \alpha_n\left(l_1\right) p\left(r_{n+1} = l_2, \mathbf{y}_{n+1} \left| r_n = l_1, \mathbf{y}_n\right.\right) \beta_{n+1}\left(l_2\right)} \quad (8)$$

with $\alpha_n\left(j\right) := p\left(r_n = j \left| \mathbf{y}_1^n\right.\right)$, $\beta_n\left(j\right) := \frac{p\left(\mathbf{y}_{n+1}^N | r_n = j, \mathbf{y}_n\right)}{p\left(\mathbf{y}_{n+1}^N | \mathbf{y}_1^n\right)}$ represent the normalized "forward", "backward" probabilities [23]. They are calculated from the following recursions:

$$\alpha_1\left(j\right) = p\left(r_1 = j \left| \mathbf{y}_1\right.\right);$$
$$\alpha_{n+1}\left(j\right) = \frac{\sum\limits_{l\in\boldsymbol{\Omega}} \alpha_n\left(l\right) p\left(r_{n+1} = j, \mathbf{y}_{n+1} \left| r_n = l, \mathbf{y}_n\right.\right)}{\sum\limits_{(l_1,l_2)\in\boldsymbol{\Omega}^2} \alpha_n\left(l_1\right) p\left(r_{n+1} = l_2, \mathbf{y}_{n+1} \left| r_n = l_1, \mathbf{y}_n\right.\right)}, \quad (9)$$

$$\beta_N\left(j\right) = 1;$$
$$\beta_n\left(j\right) = \frac{\sum\limits_{l\in\boldsymbol{\Omega}} \beta_{n+1}\left(l\right) p\left(r_{n+1} = l, \mathbf{y}_{n+1} \left| r_n = j, \mathbf{y}_n\right.\right)}{\sum\limits_{(l_1,l_2)\in\boldsymbol{\Omega}^2} \alpha_n\left(l_1\right) p\left(r_{n+1} = l_2, \mathbf{y}_{n+1} \left| r_n = l_1, \mathbf{y}_n\right.\right)}, \quad (10)$$

Thus, the next parameters $\left(\boldsymbol{\theta}^{\star}\right)^{(i+1)}$ is obtained by

$$\left(p_{j,k}\right)^{(i+1)} = \frac{1}{N-1}\sum_{n=1}^{N-1}\psi_n\left(j,k\right);$$
$$\left(\mathbf{M}_{j,k}^{\mathbf{y}_1^2}\right)^{(i+1)} = \frac{\sum\limits_{n=1}^{N-1}\psi_n\left(j,k\right)\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix}}{\sum\limits_{n=1}^{N-1}\psi_n\left(j,k\right)};$$
$$\left(\boldsymbol{\Gamma}_{j,k}^{\mathbf{y}_1^2}\right)^{(i+1)} = \frac{\sum\limits_{n=1}^{N-1}\psi_n\left(j,k\right)\left(\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix} - \left(\mathbf{M}_{j,k}^{\mathbf{y}_1^2}\right)^{(i+1)}\right)\left(\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix} - \left(\mathbf{M}_{j,k}^{\mathbf{y}_1^2}\right)^{(i+1)}\right)^{\intercal}}{\sum\limits_{n=1}^{N-1}\psi_n\left(j,k\right)}. \quad (11)$$

To find initial $\left(\boldsymbol{\theta}^{\star}\right)^{(0)}$, we use simply the empirical estimation based on the classified $\mathbf{y}_1^N$ obtained through classic K-means. EM is stopped after the change of the likelihood between two iterations is considered small enough. Then, having $\phi_n\left(j\right) :=$

$p^{\star}\left(r_n = j \left|\mathbf{y}_1^N\right.\right) = \sum_{k=1}^{K} \psi_n(j,k)$ got from the last iteration, $(\hat{\mathbf{r}}^{\star})_1^N = (\hat{r}_1^{\star}, \cdots, \hat{r}_N^{\star})$ is obtained by $\hat{r}_n^{\star} = \arg\max_j \phi_n(j)$ adopting the MPM criterion.

B. Switching EM

Knowing the switches, a CGPMSM is actually a varying parameter "Pairwise Gaussian Markov Model" (PGMM [5]). In this Section, we assume that $(\hat{\mathbf{r}}^{\star})_1^N$ are the true switches, and extend the constant parameter PGMM-based EM algorithm proposed in [24] to the varying parameter case. We call this extension "Switching EM".

The function that EM updates under the assumption is

$$\boldsymbol{\theta}^{(l+1)} = \arg\max_{\boldsymbol{\theta}} \left[\mathbb{E}_{\boldsymbol{\theta}^{(l)}}\left[\ln p_{\boldsymbol{\theta}}\left(\mathbf{X}_1^N, (\hat{\mathbf{r}}^{\star})_1^N, \mathbf{y}_1^N\right) \,\middle|\, (\hat{\mathbf{r}}^{\star})_1^N, \mathbf{y}_1^N\right]\right], \tag{12}$$

with the complete data likelihood given by

$$\ln p_{\boldsymbol{\theta}}\left(\mathbf{X}_1^N, (\hat{\mathbf{r}}^{\star})_1^N, \mathbf{y}_1^N\right) = \ln\left(p_{\boldsymbol{\theta}}(\hat{r}_1^{\star})\right) + \ln\left(p_{\boldsymbol{\theta}}\left(\mathbf{X}_1, \mathbf{y}_1 \left|\hat{r}_1^{\star}\right.\right)\right) + \sum_{n=1}^{N-1} \ln\left(p_{\boldsymbol{\theta}}\left(\hat{r}_{n+1}^{\star}\left|\hat{r}_n^{\star}\right.\right)\right) + \ln\left(p_{\boldsymbol{\theta}}\left(\mathbf{X}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{X}_n, \mathbf{y}_n, \hat{r}_n^{\star}, \hat{r}_{n+1}^{\star}\right.\right)\right). \tag{13}$$

Thus we have

$$\boldsymbol{\theta}^{(l+1)} = \arg\max_{\boldsymbol{\theta}} \Big[\mathbb{E}_{\boldsymbol{\theta}^{(l)}}\Big[\ln\left(p_{\boldsymbol{\theta}}\left(\mathbf{X}_1, \mathbf{y}_1\left|\hat{r}_1^{\star}\right.\right)\right) + \sum_{n=1}^{N-1} \ln\left(p_{\boldsymbol{\theta}}\left(\mathbf{X}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{X}_n, \mathbf{y}_n, \hat{r}_n^{\star}, \hat{r}_{n+1}^{\star}\right.\right)\right) \,\Big|\, (\hat{\mathbf{r}}^{\star})_1^N, \mathbf{y}_1^N\Big]\Big]. \tag{14}$$

1) E-step: As no confusion will be introduced, we temporally remove the dependence notation related to $(\hat{\mathbf{r}}^{\star})_1^N$. The solution of (14) is function of $p\left(\mathbf{x}_n\left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|N}, \boldsymbol{P}_{n|N}\right)$. Let us recall how it is computed from the forward recursion followed by a backward one.

The forward recursion gives $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n+1|n+1}, \boldsymbol{P}_{n+1|n+1}\right)$ from $p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|n}, \boldsymbol{P}_{n|n}\right)$ and ends at $p\left(\mathbf{x}_N\left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{N|N}, \boldsymbol{P}_{N|N}\right)$. Then the backward recursion gives $p\left(\mathbf{x}_n\left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|N}, \boldsymbol{P}_{n|N}\right)$ from $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n+1|N}, \boldsymbol{P}_{n+1|N}\right)$ and ends with $p\left(\mathbf{x}_1\left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{1|N}, \boldsymbol{P}_{1|N}\right)$. More precisely, they run as follows. Define $\mathbf{N}_{n+1}^{\mathbf{x}}$ and $\mathbf{N}_{n+1}^{\mathbf{y}}$ the items linked to the means that

$$\begin{bmatrix}\mathbf{N}_{n+1}^{\mathbf{x}}(\mathbf{R}_n^{n+1}) \\ \mathbf{N}_{n+1}^{\mathbf{y}}(\mathbf{R}_n^{n+1})\end{bmatrix} = \begin{bmatrix}\mathbf{M}^{\mathbf{x}}(R_{n+1}) \\ \mathbf{M}^{\mathbf{y}}(R_{n+1})\end{bmatrix} - \boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})\begin{bmatrix}\mathbf{M}^{\mathbf{x}}(R_n) \\ \mathbf{M}^{\mathbf{y}}(R_n)\end{bmatrix}.$$

Let

$$\begin{aligned} \boldsymbol{S}_{n|n+1} &:= \boldsymbol{\mathcal{Q}}^{\mathbf{yy}} + \boldsymbol{\mathcal{F}}^{\mathbf{yx}}\boldsymbol{P}_{n|n}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yx}}\right)^{\intercal}; \\ \boldsymbol{K}_{n|n+1} &:= \boldsymbol{P}_{n|n}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yx}}\right)^{\intercal}\left(\boldsymbol{S}_{n|n+1}\right)^{-1}; \\ \hat{\mathbf{y}}_{n+1|n} &:= \boldsymbol{\mathcal{F}}^{\mathbf{yx}}\hat{\mathbf{x}}_{n|n} + \boldsymbol{\mathcal{F}}^{\mathbf{yy}}\mathbf{y}_n + \mathbf{N}_{n+1}^{\mathbf{y}}; \\ \tilde{\mathbf{y}}_{n+1|n} &:= \mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1|n}. \end{aligned} \tag{15}$$

$$\begin{aligned} \hat{\mathbf{x}}_{n|n+1} &= \hat{\mathbf{x}}_{n|n} + \boldsymbol{K}_{n|n+1}\tilde{\mathbf{y}}_{n+1|n}; \\ \boldsymbol{P}_{n|n+1} &= \boldsymbol{P}_{n|n} - \boldsymbol{K}_{n|n+1}\boldsymbol{S}_{n|n+1}\left(\boldsymbol{K}_{n|n+1}\right)^{\intercal}, \end{aligned} \tag{16}$$

Then

$$\begin{aligned} \hat{\mathbf{x}}_{n+1|n+1} &= \boldsymbol{\mathcal{A}}\hat{\mathbf{x}}_{n|n+1} + \boldsymbol{\mathcal{B}}_n; \\ \boldsymbol{P}_{n+1|n+1} &= \boldsymbol{\mathcal{Q}}_2 + \boldsymbol{\mathcal{A}}\boldsymbol{P}_{n|n+1}\boldsymbol{\mathcal{A}}^{\intercal}, \end{aligned} \tag{17}$$

in which

$$\begin{aligned} \boldsymbol{\mathcal{A}} &:= \boldsymbol{\mathcal{F}}^{\mathbf{xx}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{F}}^{\mathbf{yx}}; \\ \boldsymbol{\mathcal{B}}_n &:= \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\mathbf{y}_{n+1} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\mathbf{N}_{n+1}^{\mathbf{y}} \\ &\quad + \left(\boldsymbol{\mathcal{F}}^{\mathbf{xy}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{F}}^{\mathbf{yy}}\right)\mathbf{y}_n + \mathbf{N}_{n+1}^{\mathbf{x}}; \\ \boldsymbol{\mathcal{Q}}_2 &:= \boldsymbol{\mathcal{Q}}^{\mathbf{xx}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{Q}}^{\mathbf{yx}}. \end{aligned} \tag{18}$$

Regarding the backward recursion, $\mathcal{N}\left(\hat{\mathbf{x}}_{n|N}, \boldsymbol{P}_{n|N}\right)$ is computed from $\mathcal{N}\left(\hat{\mathbf{x}}_{n+1|N}, \boldsymbol{P}_{n+1|N}\right)$ with

$$\begin{aligned} \hat{\mathbf{x}}_{n|N} &= \hat{\mathbf{x}}_{n|n+1} + \boldsymbol{K}_{n|N}(\hat{\mathbf{x}}_{n+1|N} - \hat{\mathbf{x}}_{n+1|n+1}); \\ \boldsymbol{P}_{n|N} &= \boldsymbol{P}_{n|n+1} + \boldsymbol{K}_{n|N}(\boldsymbol{P}_{n+1|N} - \boldsymbol{P}_{n+1|n+1})\left(\boldsymbol{K}_{n|N}\right)^{\intercal}, \end{aligned} \tag{19}$$

where $\boldsymbol{K}_{n|N} = \boldsymbol{P}_{n|n+1}\boldsymbol{\mathcal{A}}^{\intercal}\left(\boldsymbol{P}_{n+1|n+1}\right)^{-1}$.

For later use, we also calculate the covariance $\boldsymbol{C}_{n+1,n|N}$ of $\mathbf{X}_{n+1}$ and $\mathbf{X}_n$ conditional on $\mathbf{y}_1^N$, given by:

$$\boldsymbol{C}_{n+1,n|N} = \boldsymbol{P}_{n+1|N}\left(\boldsymbol{K}_{n|N}\right)^{\intercal}. \tag{20}$$

We should notice that this computation is of difference from the ones in [24], [25], because there is a "shift" of the pair from $(\mathbf{X}_n, \mathbf{Y}_{n-1})$ in the model handled in these two articles to $(\mathbf{X}_n, \mathbf{Y}_n)$ in our model (3). Moreover, our model considers that the means of $(\mathbf{X}_n, \mathbf{Y}_n)$ change with the switches.

2) M-step: For maximization, let us set $(\hat{\mathbf{r}}^{\star})_1^N = \mathbf{r}_1^N$ for simplification. Besides, when dealing with the optimization problem of (12), one may replace the quantities $(\mathbf{X}_n, \mathbf{y}_n)$ with the centered ones, for which we set $\mathbf{Z}_n' = \mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}(R_n)$. Defining:

$$\begin{aligned} \boldsymbol{C}^{n,n,(l)} &:= \mathbb{E}_{\boldsymbol{\theta}^{(l)}}\left[\mathbf{Z}_n'\left(\mathbf{Z}_n'\right)^{\intercal}\left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right] = \\ &\begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n)\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n)\end{bmatrix}^t + \begin{bmatrix}\boldsymbol{P}_{n|N} & 0 \\ 0 & 0\end{bmatrix}, \end{aligned} \tag{21}$$

$$\begin{aligned} \boldsymbol{C}^{n+1,n,(l)} &:= \mathbb{E}_{\boldsymbol{\theta}^{(l)}}\left[\mathbf{Z}_{n+1}'\left(\mathbf{Z}_n'\right)^{\intercal}\left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right] \\ &= \begin{bmatrix}\hat{\mathbf{x}}_{n+1|N} - \mathbf{M}^{\mathbf{x}}(r_{n+1}) \\ \mathbf{y}_{n+1} - \mathbf{M}^{\mathbf{y}}(r_{n+1})\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n)\end{bmatrix}^t \\ &\quad + \begin{bmatrix}\boldsymbol{C}_{n+1,n|N} & 0 \\ 0 & 0\end{bmatrix}. \end{aligned} \tag{22}$$

Solutions $\boldsymbol{\mathcal{F}}_{j,k}^{(l+1)} = \boldsymbol{\mathcal{F}}^{(l+1)}(r_n = j, r_{n+1} = k)$ and $\boldsymbol{\mathcal{Q}}_{j,k}^{(l+1)} = \boldsymbol{\mathcal{Q}}^{(l+1)}(r_n = j, r_{n+1} = k)$ of (12) for each $j, k \in \boldsymbol{\Omega}$ verify that

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\mathcal{F}}_{j,k}} \sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(l)}}\left[p_{\boldsymbol{\theta}}\left(\mathbf{z}_{n+1}'\left|\mathbf{z}_n'\right.\right)\left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right] &= \mathbf{0}, \\ \frac{\partial}{\partial \boldsymbol{\mathcal{Q}}_{j,k}} \sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(l)}}\left[p_{\boldsymbol{\theta}}\left(\mathbf{z}_{n+1}'\left|\mathbf{z}_n'\right.\right)\left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right] &= \mathbf{0}, \end{aligned} \tag{23}$$

with $p_{\boldsymbol{\theta}}\left(\mathbf{z}_{n+1}'\left|\mathbf{z}_n'\right.\right) = \mathcal{N}\left(\boldsymbol{\mathcal{F}}(\mathbf{r}_n^{n+1})\mathbf{z}_n', \boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1})\right)$. Setting $\delta_n(j,k) := \mathbb{1}(r_n = j, r_{n+1} = k)$ and $\boldsymbol{Card}(j,k) := \sum_{n=1}^{N-1} \delta_n(j,k)$, we get the expressions of $\boldsymbol{\mathcal{F}}_{j,k}^{(l+1)}$ and $\boldsymbol{\mathcal{Q}}_{j,k}^{(l+1)}$ that

$$\begin{aligned} \boldsymbol{\mathcal{F}}_{j,k}^{(l+1)} &= \widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}_{n+1}', \mathbf{z}_n'}\left(\widetilde{\boldsymbol{C}}_{j,k}^{n,n,(l)}\right)^{-1}; \\ \boldsymbol{\mathcal{Q}}_{j,k}^{(l+1)} &= \frac{1}{\boldsymbol{Card}(j,k)}\left(\widetilde{\boldsymbol{C}}_{j,k}^{n+1,n+1,(l)} - \boldsymbol{\mathcal{F}}_{j,k}^{(l+1)}\left(\widetilde{\boldsymbol{C}}_{j,k}^{n+1,n,(l)}\right)^{\intercal}\right), \end{aligned} \tag{24}$$

in which

$$\begin{aligned} \widetilde{\boldsymbol{C}}_{j,k}^{n,n,(l)} &:= \sum_{n=1}^{N-1} \delta_n(j,k)\, \boldsymbol{C}^{n,n,(l)}; \\ \widetilde{\boldsymbol{C}}_{j,k}^{n+1,n,(l)} &:= \sum_{n=1}^{N-1} \delta_n(j,k)\, \boldsymbol{C}^{n+1,n,(l)}; \\ \widetilde{\boldsymbol{C}}_{j,k}^{n+1,n+1,(l)} &:= \sum_{n=1}^{N-1} \delta_n(j,k)\, \boldsymbol{C}^{n+1,n+1,(l)}. \end{aligned} \tag{25}$$

Remark 2.2 Although CGOMSM seems naturally fit for the assumption that we need to estimate $(\hat{\mathbf{r}}^{\star})_1^N$, it is unidentifiable through EM. So DEM can work for general CGPMSMs except the very CGOMSM case.

III. Approximated unsupervised smoothing

As fast exact smoothing is not feasible in CGPMSMs, we propose to study the performance of DEM with two smoothing approaches.

The first one, called DEM-CGOMSM, uses the parameters estimated through DEM and modified into CGOMSM to have $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{r}_n^{n+1}) = \mathbf{0}$ for all $\mathbf{r}_n^{n+1} \in \boldsymbol{\Omega}^2$, so that fast smoothing is workable under the approached CGOMSM model [5].

The second one, called DEM-CGPMSM, uses the classic approximation, replacing Gaussian mixtures $p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^n \right.\right)$ and $p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^N \right.\right)$ by Gaussian distributions with the same means and variances in recursive computations. However, the originality of DEM-CGPMSM is that the distributions $p\left(r_{n+1} \left| r_n, \mathbf{y}_1^n \right.\right)$ and $p\left(r_{n+1} \left| r_n, \mathbf{y}_1^N \right.\right)$ needed are those estimated by EM for PMC in DEM. In detail, assuming that $p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^n \right.\right)$ is Gaussian, mean and variance of Gaussian $p\left(\mathbf{x}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right)$ can be computed by (17). Then,

$$p\left(\mathbf{x}_{n+1} \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right) = \sum_{r_n} p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right) p\left(\mathbf{x}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right). \tag{26}$$

Similarly, $p\left(\mathbf{x}_n \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^N \right.\right)$ is computed from $p\left(\mathbf{x}_{n+1} \left| r_{n+1}, \mathbf{y}_1^N \right.\right)$ (assumed Gaussian) using the backward recursion (19), and then

$$p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^N \right.\right) = \sum_{r_{n+1}} p\left(r_{n+1} \left| r_n, \mathbf{y}_1^N \right.\right) p\left(\mathbf{x}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^N \right.\right). \tag{27}$$

Probabilities $p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right)$ and $p\left(r_{n+1} \left| r_n, \mathbf{y}_1^N \right.\right)$ in these two equations above are computable because of the assumed Markovianity of $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ in DEM. Finally, the smoothing result is given by $\hat{\mathbf{x}}_n = \sum_{r_n} \mathbb{E}\left[\mathbf{x}_n \left| r_n, \mathbf{y}_1^N \right.\right] p\left(r_n \left| \mathbf{y}_1^N \right.\right)$ for each $n = 1, \cdots, N$.

IV. Experiments

Two experiments are conducted to test the performance of the DEM algorithm on simulated data under CGPMSMs, and to study the two proposed unsupervised smoothing approaches based on DEM parameter estimation. The first experimental series regards different noise levels, and the second one considers the impact of the conditional mean value $\mathbf{M}^{\mathbf{y}}(R_n)$.

A. Experiment of varying noise level

Let us consider a simple case of CGPMSM, where $m = q = 1$, $\boldsymbol{\Omega} = \{1, 2\}$, and with joint probabilities of $\mathbf{R}_1^N$ given by $p_{1,1} = p_{2,2} = 0.45$, $p_{1,2} = p_{2,1} = 0.05$. The means of $\mathbf{X}_1^N$ and $\mathbf{Y}_1^N$ of both switch classes are set to be zero (the means of $\mathbf{X}_1^N$ are assumed to be known, since it can't be recovered), while the variance–covariance matrices of $p\left(\mathbf{X}_1^2, \mathbf{Y}_1^2 \left| r_1 = j, r_2 = k \right.\right)$, $j, k \in \boldsymbol{\Omega}$ are of the form (28) (see also the dependence graph in Fig. 1).

$$\boldsymbol{\Gamma}_{j,k}^{\mathbf{z}_1^2} = \begin{bmatrix} 1 & b_j & a_{j,k} & d_{j,k} \\ b_j & 1 & e_{j,k} & c_{j,k} \\ a_{j,k} & e_{j,k} & 1 & b_k \\ d_{j,k} & c_{j,k} & b_k & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_j^{\mathbf{z}} & \boldsymbol{\Sigma}_{j,k}^{\mathbf{z}_1,\mathbf{z}_2} \\ \left(\boldsymbol{\Sigma}_{j,k}^{\mathbf{z}_1,\mathbf{z}_2}\right)^{\intercal} & \boldsymbol{\Gamma}_k^{\mathbf{z}} \end{bmatrix}. \tag{28}$$

Then the equivalent parameters $\boldsymbol{\mathcal{F}}_{j,k}$ and $\boldsymbol{\mathcal{Q}}_{j,k}$ are given by:

$$\boldsymbol{\mathcal{F}}_{j,k} = \left(\boldsymbol{\Sigma}_{j,k}^{\mathbf{z}_1,\mathbf{z}_2}\right)^{\intercal} \left(\boldsymbol{\Gamma}_j^{\mathbf{z}}\right)^{-1}; \quad \boldsymbol{\mathcal{Q}}_{j,k} = \boldsymbol{\Gamma}_k^{\mathbf{z}} - \boldsymbol{\mathcal{F}}_{j,k} \boldsymbol{\Sigma}_{j,k}^{\mathbf{z}_1,\mathbf{z}_2}. \tag{29}$$

And conversely, using Lyapunov equation [26], (29) implies

$$\begin{aligned} \boldsymbol{\Gamma}_j^{\mathbf{z}} &= argvec\left[\left(\mathbf{I} - \boldsymbol{\mathcal{F}}_{j,j} \otimes \boldsymbol{\mathcal{F}}_{j,j}\right)^{-1} vec\left(\boldsymbol{\mathcal{Q}}_{j,j}\right)\right]; \\ \boldsymbol{\Sigma}_{j,k}^{\mathbf{z}_1,\mathbf{z}_2} &= \left(\boldsymbol{\mathcal{F}}_{j,k} \boldsymbol{\Gamma}_j^{\mathbf{z}}\right)^{\intercal}, \end{aligned} \tag{30}$$

where $argvec(.)$ denotes the inverse function of the operator $vec(.)$ that stacks the columns of a matrix, and $\otimes$ represents the Kronecker product. We fix the value of $a_{j,1} = 0.1$, $a_{j,2} = 0.5$; $c_{j,1} = 0.5$, $c_{j,2} = 0.9$, while the other parameters in $\boldsymbol{\Gamma}_{j,k}^{\mathbf{z}_1^2}$ are set as given in TABLE I, regarding five cases with decreasing noise level. Then, six smoothing methods are considered for comparison, among which, three are based on true parameters and three on DEM estimated ones.

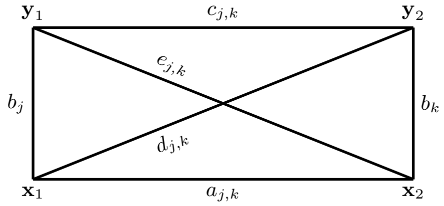

Fig. 1. Dependence graph of $(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2)$ conditional on $(r_n = j, r_{n+1} = k)$.

TABLE I
Parameters of five different noise cases.

Case	b_1	b_2	$e_{j,1}$	$e_{j,2}$	$d_{1,1}$	$d_{1,2}$	$d_{2,1}$	$d_{2,2}$
1	0.00	0.20	0.40	0.10	0.30	0.30	0.39	0.47
2	0.10	0.30	0.50	0.20	0.35	0.39	0.42	0.54
3	0.20	0.40	0.60	0.30	0.39	0.47	0.45	0.61
4	0.30	0.50	0.70	0.40	0.42	0.54	0.48	0.68
5	0.40	0.60	0.80	0.50	0.45	0.61	0.49	0.73

The three true parameter based methods denoted with "TR" are:

1. **TR-CGPMSM**: optimal reference smoothing based on CGPMSM with true parameters and true switches $\mathbf{R}_1^N = \mathbf{r}_1^N$;
2. **TR-CGOMSM**: optimal smoothing based on CGOMSM obtained from CGPMSM by replacing $d_{j,k}$ with $d_{j,k}^{\star} = c_{j,k} b_j$, for all $j, k \in \boldsymbol{\Omega}$, which gives $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{r}_1^2) = \mathbf{0}$;
3. **TR-CGLSSM**: optimal smoothing based on true switches $\mathbf{R}_1^N = \mathbf{r}_1^N$ and CGLSSM obtained from CGPMSM by replacing $d_{j,k}$, $e_{j,k}$, $c_{j,k}$ with $d_{j,k}^{\star} = a_{j,k} b_k$, $e_{j,k}^{\star} = a_{j,k} b_j$, $c_{j,k}^{\star} = b_j a_{j,k} b_k$, for all $j, k \in \boldsymbol{\Omega}$ which gives $\boldsymbol{\mathcal{F}}^{\mathbf{xy}}(\mathbf{r}_1^2) = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(\mathbf{r}_1^2) = \mathbf{0}$.

The three unsupervised DEM based smoothing methods are:

4. **DEM-CGPMSM**: smoothing based on parameters from DEM with details given in Section III;
5. **DEM-CGOMSM**: smoothing based on parameters from DEM, and modified to become a CGOMSM (with same modifications as in TR-CGOMSM);
6. **DEM-CGLSSM**: smoothing based on parameters from DEM, and modified to become a CGLSSM (with same modifications as in TR-CGLSSM).

All experiments are carried out on $N = 10000$ samples, with 100 iterations for first EM, and 500 iterations for switching EM (assuming convergence of the algorithms), the initial value $\boldsymbol{\mathcal{F}}_{j,k}^{(0)}$ and $\boldsymbol{\mathcal{Q}}_{j,k}^{(0)}$ are set as:

$$\boldsymbol{\mathcal{F}}_{j,1}^{(0)} = \begin{bmatrix} -0.5 & 1.0 \\ 0.2 & 0.5 \end{bmatrix}; \; \boldsymbol{\mathcal{F}}_{j,2}^{(0)} = \begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 0.5 \end{bmatrix};$$

$$\boldsymbol{\mathcal{Q}}_{j,k}^{(0)} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}.$$

The experiments are conducted with python 3.6 programing language on a 3.7GHz CPU. The entire DEM parameter estimation takes about 29 minutes (3 minutes for first EM for PMC, and 26 minutes for the following switching EM). It takes 11 seconds for the later smoothing using DEM-CGPMSM and 20 seconds using DEM-CGOMSM. All results are averages of 100 independent experiments. The smoothing results related to different methods are illustrated in Fig. 2. The

TABLE II
$\mathcal{F}_{j,k}$, $\mathcal{Q}_{j,k}$ AND THEIR DEM ESTIMATES (CASE 3 OF TABLE I).

	$\mathcal{F}_{1,1}$	$\mathcal{F}_{1,2}$	$\mathcal{F}_{2,1}$	$\mathcal{F}_{2,2}$
True	$\begin{bmatrix}-0.02 & 0.60\\ 0.30 & 0.44\end{bmatrix}$	$\begin{bmatrix}0.46 & 0.21\\ 0.30 & 0.84\end{bmatrix}$	$\begin{bmatrix}-0.17 & 0.67\\ 0.30 & 0.38\end{bmatrix}$	$\begin{bmatrix}0.45 & 0.12\\ 0.30 & 0.78\end{bmatrix}$
DEM	$\begin{bmatrix}-0.01 & 0.98\\ 0.17 & 0.30\end{bmatrix}$	$\begin{bmatrix}-0.04 & 0.56\\ -0.01 & 0.74\end{bmatrix}$	$\begin{bmatrix}0.61 & 0.08\\ 0.33 & 0.74\end{bmatrix}$	$\begin{bmatrix}0.58 & 0.06\\ 0.17 & 0.84\end{bmatrix}$
	$\mathcal{Q}_{1,1}$	$\mathcal{Q}_{1,2}$	$\mathcal{Q}_{2,1}$	$\mathcal{Q}_{2,2}$
True	$\begin{bmatrix}0.64 & -0.09\\ -0.09 & 0.66\end{bmatrix}$	$\begin{bmatrix}0.71 & 0.00\\ 0.00 & 0.10\end{bmatrix}$	$\begin{bmatrix}0.62 & -0.06\\ -0.06 & 0.67\end{bmatrix}$	$\begin{bmatrix}0.74 & 0.02\\ 0.02 & 0.11\end{bmatrix}$
DEM	$\begin{bmatrix}0.55 & 0.09\\ 0.09 & 0.80\end{bmatrix}$	$\begin{bmatrix}0.27 & 0.19\\ 0.19 & 0.29\end{bmatrix}$	$\begin{bmatrix}1.15 & 0.02\\ 0.02 & 0.17\end{bmatrix}$	$\begin{bmatrix}0.62 & 0.02\\ 0.02 & 0.15\end{bmatrix}$

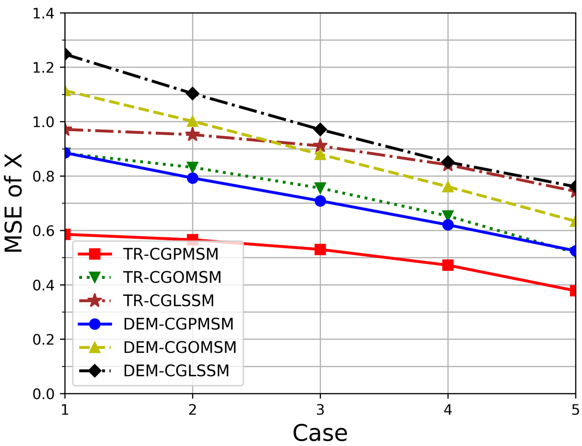

Fig. 2. Restoration MSE of Series 1 ($\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2) = 0.3$).

estimated parameters $\mathcal{F}_{j,k}$ and $\mathcal{Q}_{j,k}$ under case 3 are reported in TABLE II, while the estimated means are $\mathbf{M}^{\mathbf{y}}(r_n = 1) = 0.002$; $\mathbf{M}^{\mathbf{y}}(r_n = 2) = 0.001$, and the estimated joint probabilities of switches are $p_{1,1} = 0.40$; $p_{1,2} = p_{2,1} = 0.06$; $p_{2,2} = 0.48$.

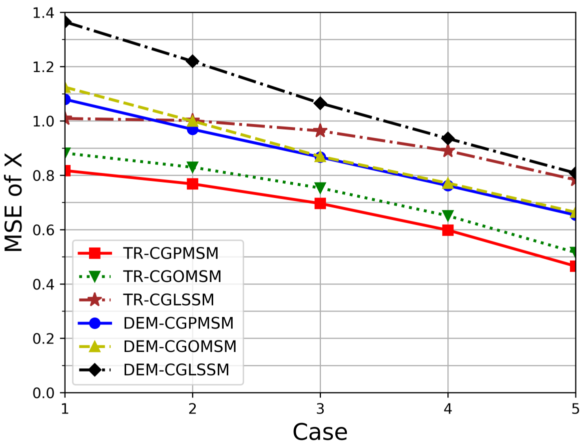

Fig. 3. Restoration MSE of Series 1 ($\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2) = 0.1$).

Moreover, a parallel experiment is conducted with $\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2)$ set to be 0.1 by changing the setting $d_{j,k}$ in Table I using (30). The smoothing results are reported in Fig. 3. Then, from this series of experiments, we can summarize that:

1) comparing TR-CGPMSM, TR-CGOMSM, and TR-CGLSSM shows how models are "far" from each other when smoothing is concerned. As expected, TR-CGOMSM can be close to TR-CGPMSM if $\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2)$ is small. TR-CGLSSM is to be avoided as MSE is often about 50% larger than that obtained by TR-CGPMSM;
2) the main conclusion is that DEM-CGPMSM is of interest when it comes to unsupervised smoothing. In general, the decrease of smoothing efficiency with respect to the optimal TR-CGPMSM is acceptable, and its performance has even the chance to surpass TR-CGOMSM and TR-CGLSSM.

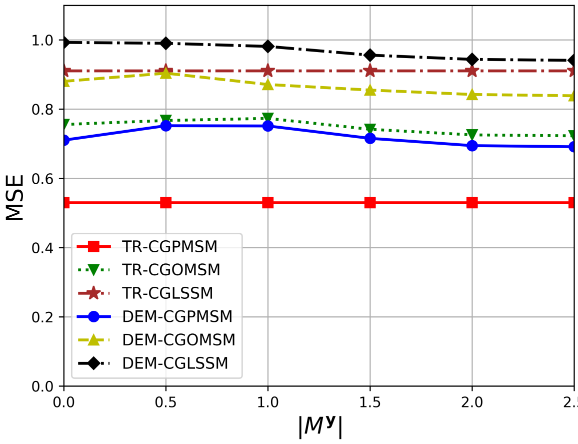

Fig. 4. Restoration MSE of Series 2 ($\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2) = 0.3$).

B. Experiment of varying mean

In this experimental series, we vary the means of observation conditionally on switches to adjust the difficulty for EM for PMC to find a suitable $(\hat{\mathbf{r}}^\star)_1^N$ instead of the true switches. All the simulation conditions and model parameters are set the same as the Case 3 with $\mathcal{F}^{\mathbf{yx}}(\mathbf{r}_1^2) = 0.3$ in previous series, except a change on $\mathbf{M}^{\mathbf{y}}(r_n)$ values. More precisely, we range $|\mathbf{M}^{\mathbf{y}}|$ from 0.0 to 2.5, where $|\mathbf{M}^{\mathbf{y}}|$ represents the absolute value of $\mathbf{M}^{\mathbf{y}}(r_n)$, and $\mathbf{M}^{\mathbf{y}}(r_n = 1) = -\mathbf{M}^{\mathbf{y}}(r_n = 2)$. For example, $|\mathbf{M}^{\mathbf{y}}| = 2.5$ indicates that $\mathbf{M}^{\mathbf{y}}(r_n = 1) = 2.5$ and $\mathbf{M}^{\mathbf{y}}(r_n = 2) = -2.5$. Restoration MSE through the six smoothing methods is illustrated in Fig. 4.

With the increase of $|\mathbf{M}^{\mathbf{y}}|$, $(\hat{\mathbf{r}}^\star)_1^N$ got from EM is closer to the true $\mathbf{r}_1^N$ (when $|\mathbf{M}^{\mathbf{y}}| = 0.0$, the error ratio of $(\hat{\mathbf{r}}^\star)_1^N$ comparing to $\mathbf{r}_1^N$ is around 20%, while when $|\mathbf{M}^{\mathbf{y}}| = 2.5$, it is nearly 0%, which means the observations are well classified for switching EM to estimate the remaining model parameters. The tendency of smoothing MSE through all methods with unknown $\mathbf{r}_1^N$ with respect to $|\mathbf{M}^{\mathbf{y}}|$ are not monotone. This phenomenon is caused by the introduced error from removing the mean of each individual wrong classification of $\mathbf{y}_1^N$ in the switching EM process. Larger $|\mathbf{M}^{\mathbf{y}}|$ introduces more error if $\mathbf{y}_1^N$ is classified in $\hat{r}_n^\star \neq r_n$. However, generally speaking, the 20% error ratio of $(\hat{\mathbf{r}}^\star)_1^N$ to r_n does not have too much influence on smoothing result as illustrated in Fig. 4, which indicates the mildness of the Markovian assumption on $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ in the overall DEM-CGPMSM algorithm.

V. CONCLUSION

We considered the problem of unsupervised smoothing in presence of unknown switches. There are three random sequences $\mathbf{X}_1^N = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, $\mathbf{R}_1^N = \{R_1, \ldots, R_N\}$, and $\mathbf{Y}_1^N = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$, with each $\mathbf{X}_n$, R_n, $\mathbf{Y}_n$ taking their values in $\mathbb{R}^m$, $\mathbf{\Omega} = \{1, \ldots, K\}$, and, $\mathbb{R}^q$ respectively. Only $\mathbf{Y}_1^N$ are observed and the problem is to estimate realizations of $\mathbf{X}_1^N$.

We proposed a general parameter estimation method - called "Double-EM" (DEM) - from $\mathbf{Y}_1^N$ for "Conditionally Gaussian Pairwise Markov Switching Models" (CGPMSMs [5], [18]), which extend simultaneously the classic "Conditionally Gaussian Linear State-Space Models" (CGLSSMs [6], [8], [9], [27]) and the recent "Conditionally Gaussian Pairwise Markov Switching Models" (CGOMSMs [4], [5], [15]). Then we studied unsupervised smoothing based on parameters estimated from DEM. Conducted simulations show the good behavior of the proposed method. In particular, considering data produced with a CGPMSM, smoothing method based on parameters estimated with DEM is competing, or even better, that smoothing based on real paramaters, real switches, and the classical CGLSSM model.

Let us mention some perspectives of extensions or applications. The Markov switching model governing switches of this paper can be extended to a semi-Markov one, as recently studied in [28]–[32]. Otherwise, Gaussian noise can probably be extended to any kind of noise by the use of copulas, as recently proposed in [33]. Finally, any stationary non-Gaussian non-linear Markov system can be approximated by a CGPMSM, as proposed in [15] and applied to smoothing in [4]. Thus "Double-EM" (DEM) and subsequent unsupervised smoothing can be used once we are faced with data produced by stationary Markov system - with or without switches - of any form.

ACKNOWLEDGMENT

The first author would like to thank the China Scholarship Council for providing a scholarship to pursue her PhD in France.

REFERENCES

[1] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2139–2156, 1999.
[2] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump Markov linear systems," *IEEE transactions on signal processing*, vol. 49, no. 6, pp. 1216–1227, 2001.
[3] H. A. P. Blom and Y. Bar-Shalom, "Time-reversion of a hybrid state stochastic difference system with a jump-linear smoothing application," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 836–847, 1990.
[4] I. Gorynin, S. Derrode, E. Monfrini, and W. Pieczynski, "Fast smoothing in switching approximations of non-linear and non-Gaussian models," *Computational Statistics & Data Analysis*, vol. 114, pp. 38–46, 2017.
[5] N. Abbassi, D. Benboudjema, S. Derrode, and W. Pieczynski, "Optimal filter approximations in conditionally Gaussian pairwise Markov switching models," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1104–1109, April 2015.
[6] O. Cappé, E. Moulines, and T. Ryden, *Inference in hidden Markov models (Springer Series in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
[7] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar 2001.
[8] R. H. Shumway and D. S. Stoffer, "Dynamic linear models with switching," *Journal of the American Statistical Association*, vol. 86, no. 415, pp. 763–769, 1991.
[9] N. S. Kim, T. G. Kang, S. J. Kang, C. W. Han, and D. H. Hong, "Speech feature mapping based on switching linear dynamic system," *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 20, no. 2, pp. 620–631, Feb 2012.
[10] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.
[11] E. Cinquemani, R. Porreca, G. Ferrari-Trecate, and J. Lygeros, "A general framework for the identification of jump Markov linear systems," in *46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5737–5742.
[12] U. Orguner and M. Demirekler, "Maximum likelihood estimation of transition probabilities of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5093–5108, 2008.
[13] E. Ozkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump Markov nonlinear system," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 754–765, 2015.
[14] K. Kamatani, "Local degeneracy of Markov chain Monte Carlo methods," *ESAIM: Probability and Statistics*, vol. 18, pp. 713–725, 2014.
[15] I. Gorynin, S. Derrode, E. Monfrini, and W. Pieczynski, "Fast filtering in switching approximations of nonlinear Markov systems with applications to stochastic volatility," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 853–862, 2017.
[16] 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.
[17] Y. Petetin and F. Desbouvries, "A class of fast exact Bayesian filters in dynamical models with jumps," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3643–3653, 2014.
[18] W. Pieczynski, "Exact filtering in conditionally Markov switching hidden linear models," *Comptes Rendus Mathématique*, vol. 349, no. 9-10, pp. 587–590, 2011.
[19] S. Derrode and W. Pieczynski, "Signal and image segmentation using pairwise Markov chains," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2477–2489, 2004.
[20] P. Lanchantin, "Chaînes de Markov triplets et segmentation non supervisée de signaux," Ph.D. dissertation, Institut National des Télécommunications, Evry, France, 2006.
[21] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, pp. 253–264, 1982.
[22] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, "Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise," *Signal Processing*, vol. 91, no. 2, pp. 163–175, 2011.
[23] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.
[24] B. A. El Fquih and F. Desbouvries, "Unsupervised signal restoration in partially observed Markov chains," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3. IEEE, 2006, pp. III13–III16.
[25] V. Némesin and S. Derrode, "Robust blind pairwise Kalman algorithms using QR decompositions," *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 5–9, 2013.
[26] J. Hamilton, *Time Series Analysis*. Princeton University Press, 1994. [Online]. Available: https://books.google.fr/books?id=B8_1UBmqVUoC
[27] C. Kim, "Dynamic linear models with Markov-switching," *Journal of Econometrics*, vol. 60, no. 1, pp. 1–22, 1994.
[28] M. Faraji-Niri and M.-R. Jahed-Motlagh, "Stochastic stability and stabilization of semi-Markov jump linear systems with uncertain transition rates," *Information Technology And Control*, vol. 46, no. 1, pp. 37–52, 2017.
[29] Z. Hou, H. Dong, and P. Shi, "Asymptotic stability in the distribution of nonlinear stochastic systems with semi-Markovian switching," *The ANZIAM Journal*, vol. 49, no. 2, pp. 231–241, 2007.
[30] J. Huang and Y. Shi, "Stochastic stability and robust stabilization of semi-Markov jump linear systems," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 18, pp. 2028–2043, 2013.
[31] L. Zhang, Y. Leng, and P. Colaneri, "Stability and stabilization of discrete-time semi-Markov jump linear systems via semi-Markov kernel approach," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 503–508, 2016.
[32] L. Zhang, T. Yang, and P. Colaneri, "Stability and stabilization of semi-Markov jump linear systems with exponentially modulated periodic distributions of sojourn time," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2870–2885, 2017.
[33] S. Derrode and W. Pieczynski, "Unsupervised classification using hidden Markov chain with unknown noise copulas and margins," *Signal Processing*, vol. 128, pp. 8–17, 2016.