**THESE**

pour obtenir le grade de

**DOCTEUR DE L'ÉCOLE CENTRALE DE LYON**

Spécialité: Informatique

# Learning and Smoothing in Switching Markov Models with Copulas

dans le cadre de l'École Doctorale InfoMaths

présentée et soutenue publiquement par

**Fei ZHENG**

December 2017

**Directeur de thèse: Prof. Stéphane Derrode**

**Co-directeur de thèse: Prof. Wojciech Pieczynski**

**JURY**

| | | |
|---|---|---|
| Jean-Yves Tourneret | ENSEEIHT Toulouse | Examinateur |
| Séverine Dubuisson | UPMC Sorbonnes Universités | Rapporteur |
| François Roueff | Telecom ParisTech | Rapporteur |
| Stéphane Derrode | École Centrale de Lyon | Directeur de thèse |
| Wojciech Pieczynski | Telecom SudParis | Co-directeur de thèse |

# Contents

# List of Tables

# List of Figures

# Abstract

Switching Markov Models, also called Jump Markov Systems (JMS), are widely used in many fields such as target tracking, seismic signal processing and finance, since they can approach non-Gaussian non-linear systems. A considerable amount of related work studies linear JMS in which data restoration is achieved by Markov Chain Monte-Carlo (MCMC) methods. In this dissertation, we try to find alternative restoration solution for JMS to MCMC methods. The main contribution of our work includes two parts. Firstly, an algorithm of unsupervised restoration for a recent linear JMS known as Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM) is proposed. This algorithm combines a parameter estimation method named Double EM, which is based on the Expectation-Maximization (EM) principle applied twice sequentially, and an efficient approach for smoothing with estimated parameters. Secondly, we extend a specific sub-model of CGPMSM known as Conditionally Gaussian Observed Markov Switching Model (CGOMSM) to a more general one, named Generalized Conditionally Observed Markov Switching Model (GCOMSM) by introducing Copulas. Comparing to CGOMSM, the proposed GCOMSM adopts inherently more flexible distributions and non-linear structures, while optimal restoration is feasible. In addition, an identification method called GICE-LS based on the Generalized Iterative Conditional Estimation (GICE) and the Least-Square (LS) principles is proposed for GCOMSM to approximate any non-Gaussian non-linear systems from their sample data set. All proposed methods are tested by simulation. Moreover, the performance of GCOMSM is discussed by application on other generable non-Gaussian non-linear Markov models, for example, on stochastic volatility models which are of great importance in finance.

**Keywords**: Switching Markov models, non-Gaussian non-linear Markov system, triplet Markov chain, model identification, optimal time series data restoration, Expectation-Maximization.

# Résumé

Les modèles de Markov à sauts (appelés JMS pour Jump Markov System) sont utilisés dans de nombreux domaines tels que la poursuite de cibles, le traitement des signaux sismiques et la finance, étant donné leur bonne capacité à modéliser des systèmes non-linéaires et non-gaussiens. De nombreux travaux ont étudié les modèles de Markov linéaires pour lesquels bien souvent la restauration de données est réalisée grâce à des méthodes d'échantillonnage statistique de type Markov Chain Monte-Carlo (MCMC). Dans cette thèse, nous avons cherché des solutions alternatives aux méthodes MCMC et proposons deux originalités principales. La première a consisté à proposer un algorithme de restauration non supervisée d'un JMS particulier appelé « modèle de Markov couple à sauts conditionnellement gaussiens » (noté CGPMSM). Cet algorithme combine une méthode d'estimation des paramètres basée sur le principe Espérance-Maximisation (EM) et une méthode efficace pour lisser les données à partir des paramètres estimés. La deuxième originalité a consisté à étendre un CGPMSM spécifique appelé CGOMSM par l'introduction des copules. Ce modèle, appelé GCOMSM, permet de considérer des distributions plus générales que les distributions gaussiennes tout en conservant des méthodes de restauration optimales et rapides. Nous avons équipé ce modèle d'une méthode d'estimation des paramètres appelée GICE-LS, combinant le principe de la méthode d'estimation conditionnelle itérative généralisée et le principe des moindre-carrés linéaires. Toutes les méthodes sont évaluées sur des données simulées. En particulier, les performances de GCOMSM sont discutées au regard de modèles de Markov non-linéaires et non-gaussiens tels que la volatilité stochastique, très utilisée dans le domaine de la finance.

Mots-clés: Modèles de Markov à sauts, systèmes non-linéaires et non-gaussiens, chaîne de Markov triplet, identification de modèles, restauration optimale de séries temporelles de données, algorithme Espérance-Maximisation.

# Introduction

Time series data restoration is a common problem that we are facing in many fields. In this general problem, we are supposed to estimate the hidden sequence from an observed one, given or supposed there are some links between them. For example, in speech recognition, one wants to find out the uttered word from the given acoustic signal [55], [67], [120]; in motion detection, we are interested in discovering the real-time human activity from video or time sequential images [47], [104]. The Hidden Markov Model (HMM), since introduced in the late 1960s [46], [119], has become a popular statistical tool for modeling these "generative" sequences which can be characterized by an underlying process generating an observable sequence. HMM is such a class of models, assuming that the hidden states form a Markovian process, and the observations are "emitted" from the hidden states by some probability distribution. When dealing with discrete time processes, HMM are usually called Hidden Markov Chain (HMC) as the discrete time index makes the processes like chains. Thus, concerning the applications mentioned above, two related must-be-solved problems in HMC are:

1. Restoration problem: given the observation series $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, what the most likely hidden states $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ are.

2. Parameter estimation problem: under the case that the model parameters $\boldsymbol{\Theta}$ are unknown, how we figure out the suitable $\boldsymbol{\Theta}$ of the applied HMC.

For restoration problem, the most popular two methods are the forward-backward algorithm [120] and the Viterbi one [134], [122]. The forward-backward algorithm refers to $p\left(\mathbf{x}_n \left|\mathbf{y}_1^n\right.\right)$ and $p\left(\mathbf{x}_n \left|\mathbf{y}_1^N\right.\right)$, which are the posterior marginals of all hidden state variables given the observations, while the Viterbi algorithm aims to find the most likely sequence based on the maximization of

$p\left(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}_1, \ldots, \mathbf{y}_N\right)$. Two conditions can be met when dealing with the parameter estimation problem. One is to estimate the parameters from observations only, for instance, maximizing $p\left(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N \left| \boldsymbol{\Theta} \right.\right)$. Most of the time we use the Baum–Welch algorithm and applies the Expectation-Maximization (EM) principle to solve this parameter estimation problem with latent variables [12]. We may meet another parameter estimation occasion less tough in contrast, in which we have milder condition that sample data set which includes both hidden state samples and observation samples are given. In this case, we can simply maximize the complete data likelihood $p\left(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}_1, \ldots, \mathbf{y}_N \left| \boldsymbol{\Theta} \right.\right)$ to figure out the suitable parameters.

With the progress of the methods for HMC, new models which generalize the classic HMC are also developed. One extension is introducing the "switch" (also called "jump") into the HMC to characterize the time series behaviors in different regimes and permitting the change between model structures, leading to the so-called "switching Markov model" [3], [28]. The efficiency of the flexibility which benefits from this extension has been proved in targets tracking [11], [96], manufacturing control [17] and finance [41], [92]. The toughness under the switching Markov models is that most of the time, the Bayesian optimal restoration is no more feasible with unknown switches, so they are often approached by Markov Chain Monte-Carlo (MCMC) methods. This optimal restoration infeasibility also results in the hardness of parameter estimation for switching Markov models [8], [42], [90]. The other extension path of HMC enriches the dependences between the hidden states and observations. It means that the observations are no more simply "emitted" from the hidden states but have also some interactive effects on the hidden process. This extension results in the "Pairwise Markov Chain" (PMC) [114], and it shows in following works on image segmentations that the consideration of interpreting these complex dependences makes sense [109], [136]. Moreover, we are pleased to apply this more general model since either restoration or parameter estimation methods of HMC can be applied with small adjustment to the PMC structure.

Recently, a fusion of these two extensions has been proposed and gave birth to a linear model known as "Conditionally Gaussian Pairwise Markov Switching Model" (CGPMSM) [1], which thus owns both the abilities to model the switching regimes and consider more complete variable dependences. Moreover, it has a prominent merit over the other switching Markov models that the optimal restorations can be derived with specific model setting. The CGPMSM with this special setting is taken as its sub-model named “Conditionally Gaussian Observed Markov Switching Model” (CGOMSM) [1], and has been studied in [62], [61] for approximating any stationary Markov systems. Since the supervised restoration method and solution of parameter estimation with given samples are already considered in these previous works, in this dissertation, we are interested in developing the unsupervised restoration methods for CGPMSM. It means to find solutions for learning its parameters from only observations and conducting restorations with the learned parameters. This is one main part of our work. Also, we notice that the feasibility of optimal restoration is no need to be constrained under the Gaussian linear model structure. In fact, we can form the conditional joint distributions in switching Markov models with the introduction of Copulas, which has been widely applied in the field of finance and insurance [18], [132], [49]. The Copula can be considered as a “tie” between margins, with which a joint distribution becomes easily be written in terms of univariate marginal distribution functions. It has been successfully introduced into Markov models such as the HMC and PMC [24], [37], [38], but from our best knowledge, so far, there is no work that considers the incorporation of Copulas in a switching Markov model. Inspired by this, the second main part of our work focuses on extending the CGOMSM into a more general switching Markov model by making use of Copulas. Thus, the new model can incorporate varied conditional distribution, while still allowing optimal restorations. We also consider an iterative method to solve the parameter estimation problem for the new model using sample data set, so that the model can be applicable on approaching any Markov stationary systems and perform their time series data restorations.

## Outline of the thesis

This dissertation is divided into four Chapters, organized as follows:

Chapter 1 describes the PMC model, which is the basic structure of the switching

Markov models that we are going to study. Discrete and continuous state-space PMC are introduced separately with their matched methods of restoration and parameter estimation.

Chapter 2 focuses on the restoration methods of Gaussian linear Markov models (the CGPMSM family). Optimal restoration is derived for the special sub-model of CGPMSM known as CGOMSM. Then, for the unsupervised restoration of the CGPMSM, an EM principle based parameter estimation method from only observations is proposed and described with details. Meanwhile, two restoration approaches are presented for restoration under the general CGPMSM. The fusion of the proposed parameter estimation method and the restoration approaches leads to an unsupervised strategy whose efficiency is proved by simulations. Finally, several series of experiments are conducted to analyze the performance of the proposed unsupervised restoration method with comparison to supervised optimal and sub-optimal methods considering different impacting factors.

Chapter 3 contributes to build the general non-Gaussian model which allows optimal restorations inspired by the CGOMSM model. Firstly, we give the definition of the proposed model (briefly denoted by GCOMSM), and the way for its simulation. Then the optimal restorations (filtering and smoothing) are derived, with two simulation examples to verify their efficiency and show the generality of the proposed model. Moreover, an identification method based on "Generalized Iterative Conditional Estimation" (GICE) and Least-Square (LS) called GICE-LS is proposed for estimating the distributions and parameters of the proposed model. The efficiency of GICE-LS on identification of GCOMSM is proved by simulation. Finally, we apply the restoration for GCOMSM with GICE-LS for identification to some generable non-Gaussian non-linear systems to objectively show the merits of our algorithm comparing to the restoration for CGOMSM with its identification methods and Particle Filter.

In the end, Chapter 4 summarizes the main contributions of this dissertation, presents some limitations in the proposed methods which can be improved, and draws an outlook for possible future work.

Chapter 1

# Pairwise Markov chain and basic methods

Since proposed in [114], Pairwise Markov Chain (PMC) arouses more and more attention as a generalization of Hidden Markov Chain (HMC). Playing the same role, replacing the classic HMC, the PMC has been applied to signal and image processing fields, such as speech recognition [87], image segmentation or classification [34], [35], [109], [136]. All these works show that the PMC brings improvements on result thanks to its consideration of more complex dependence between stochastic variables.

We will introduce and detail the properties of PMC in this Chapter. In section 1.1, we explain its sub-cases of different dependences between variables. Focus on the restoration of the hidden states in PMC, we consider both discrete finite space case and continuous case in Section 1.2 and Section 1.3. Supervised and unsupervised restoration solutions for these PMC models are given. Meanwhile, the frequently used Gaussian PMCs are discussed, and some results of different restoration solutions are illustrated for discrete finite space case.

Let us consider two sequences of random variables. $\mathbf{R}_1^N = (\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_N)$, each $\boldsymbol{R}_n$ takes its value in a set $\mathscr{R}$; and $\mathbf{Y}_1^N = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N)$, each $\mathbf{Y}_n$ takes its value in a set $\mathscr{Y}$. Both the spaces $\mathscr{R}$ and $\mathscr{Y}$ can be discrete or continuous. We note further $\mathbf{H}_1^N = (\mathbf{H}_1, \mathbf{H}_2 \ldots, \mathbf{H}_N)$, where $\mathbf{H}_n = (\boldsymbol{R}_n, \mathbf{Y}_n)$, and $\mathbf{r}_1^N$, $\mathbf{y}_1^N$, $\mathbf{h}_1^N$ for the realization of $\mathbf{R}_1^N$, $\mathbf{Y}_1^N$ and $\mathbf{H}_1^N$ respectively. Then, the process $\mathbf{H}_1^N$ is a PMC if it holds the Markov property that

$$p\left(\mathbf{h}_1^N\right) = p\left(\mathbf{h}_1\right) p\left(\mathbf{h}_2 \left| \mathbf{h}_1 \right.\right) \ldots p\left(\mathbf{h}_N \left| \mathbf{h}_{N-1} \right.\right). \tag{1.1}$$

## 1.1 Different dependences in PMC

There could be varying dependences inside a PMC structure, as we can decompose the transition probability of PMC into

$$\begin{aligned} p\left(\mathbf{h}_{n+1}\left|\mathbf{h}_n\right.\right) &= p\left(\boldsymbol{r}_{n+1}, \mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \mathbf{y}_n\right.\right) \\ &= p\left(\boldsymbol{r}_{n+1}\left|\boldsymbol{r}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \boldsymbol{r}_{n+1}, \mathbf{y}_n\right.\right). \end{aligned} \tag{1.2}$$

Considering the different cases in equation (1.2), we have four sub-models of PMC, each of them holds their special dependence of noise. The dependence graphs of all these sub-cases of PMC are displayed in Figure 1.1.

(a) When $p\left(\boldsymbol{r}_{n+1}\left|\boldsymbol{r}_n, \mathbf{y}_n\right.\right) = p\left(\boldsymbol{r}_{n+1}\left|\boldsymbol{r}_n\right.\right)$ and $p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \boldsymbol{r}_{n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_{n+1}\right.\right)$, the process $\mathbf{H}_1^N$ is the well recognized HMC. More precisely, we call it "Hidden Markov Chain with Independent Noise" (HMC-IN). The transition probability in (1.2) thus becomes

$$p\left(\boldsymbol{r}_{n+1}, \mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \mathbf{y}_n\right.\right) = p\left(\boldsymbol{r}_{n+1}\left|\boldsymbol{r}_n\right.\right) p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_{n+1}\right.\right). \tag{1.3}$$

In this classic case, $\boldsymbol{R}_1^N$ is a Markov chain and $\mathbf{Y}_1, \cdots, \mathbf{Y}_N$ are independent from each other knowing $\mathbf{R}_1^N$.

(b) When $p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \boldsymbol{r}_{n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, r_{n+1}\right.\right)$, the process $\mathbf{H}_1^N$ is called "Hidden Markov Chain with Independent Noise of order 2" (HMC-IN2) with transition probability

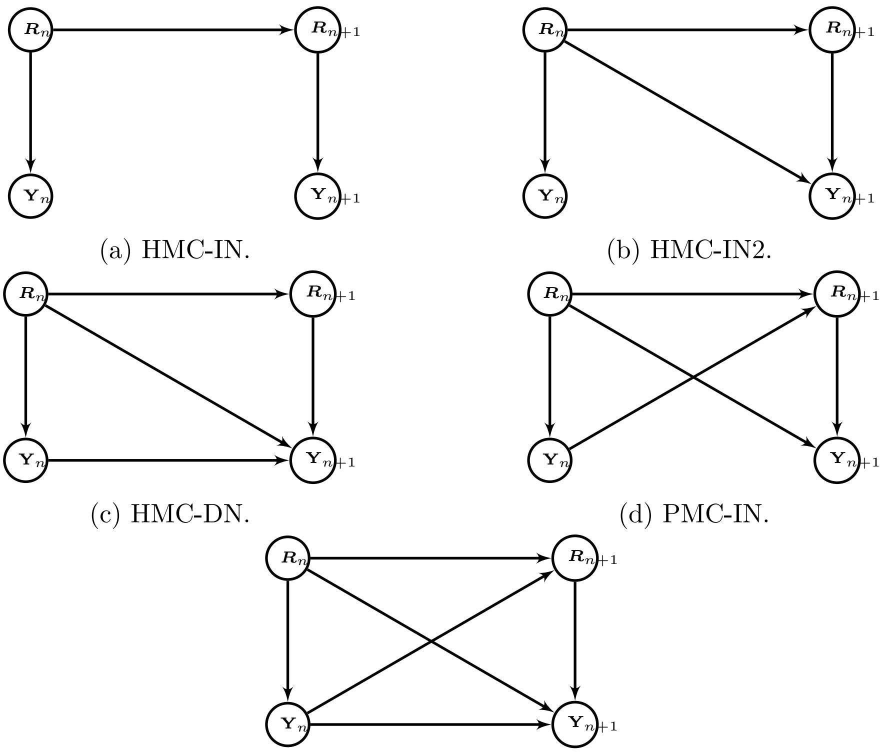
$$p\left(\boldsymbol{r}_{n+1}, \mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \mathbf{y}_n\right.\right) = p\left(\boldsymbol{r}_{n+1}\left|\boldsymbol{r}_n\right.\right) p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, r_{n+1}\right.\right). \tag{1.4}$$

$\boldsymbol{R}_1^N$ is still a Markov chain, and $\mathbf{Y}_1, \cdots, \mathbf{Y}_N$ are independent conditionally on $\boldsymbol{R}_1^N$, but the dependence on $\boldsymbol{R}_1^N$ is more complicated than in HMC-IN. HMC-IN can be seen as a particular case of this HMC-IN2.

(c) If only $\boldsymbol{R}_1^N$ is assumed Markovian, the process $\mathbf{H}_1^N$ is called "Hidden Markov Chain with Dependent Noise" (HMC-DN), with the transition probability writes

$$p\left(\boldsymbol{r}_{n+1}, \mathbf{y}_{n+1} \left| \boldsymbol{r}_n, \mathbf{y}_n \right.\right) = p\left(\boldsymbol{r}_{n+1} \left| \boldsymbol{r}_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \boldsymbol{r}_n, \boldsymbol{r}_{n+1}, \mathbf{y}_n \right.\right). \tag{1.5}$$

Under this case, $\mathbf{Y}_1, \cdots, \mathbf{Y}_N$ become dependent from each other conditionally on $\boldsymbol{R}_1^N$, and obviously, this is a more general case than the last two cases.



(a) HMC-IN.

(b) HMC-IN2.

(c) HMC-DN.

(d) PMC-IN.

(e) PMC.

Figure 1.1: Dependence graphs of particular sub-models of PMC.

(d) Here we consider $\boldsymbol{R}_1^N$ no more Markovian, and $\mathbf{Y}_1, \cdots, \mathbf{Y}_N$ independent conditionally on $\boldsymbol{R}_1^N$, which means that

$$p\left(\boldsymbol{r}_{n+1}, \mathbf{y}_{n+1} \left| \boldsymbol{r}_n, \mathbf{y}_n \right.\right) = p\left(\boldsymbol{r}_{n+1} \left| \boldsymbol{r}_n, \mathbf{y}_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \boldsymbol{r}_n, \boldsymbol{r}_{n+1} \right.\right) \tag{1.6}$$

This special case is called "Pairwise Markov Chain with Independent Noise" (PMC-IN), and if we call the most general PMC the "Pairwise Markov Chain with Dependent Noise" (PMC-DN), in which all dependences are conserved, the PMC-IN is its sub-case. Later if no confusion will be introduced, "PMC" will refer to the PMC-DN instead.

Let us notice that, there are a lot of works on Markov models among which some also inspired by the "pairwise" idea. In [50], [51], the PMC is called Bivariate Markov Chain; similarly, it is called the Coupled Markov Chain or Models in [21], [20], [118]; moreover, the Double Chain Markov Model discussed in [13] and [14] which is actually the HMC-DN but with $p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_n, \boldsymbol{r}_{n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1}\left|\boldsymbol{r}_{n+1}, \mathbf{y}_n\right.\right)$. The novelty of PMC is the fact that $\mathbf{R}_1^N$ is not necessarily Markovian, and it gives necessary and sufficient conditions for stationary time-reversible model to exist [84]. Besides, one should pay attention that these works have different emphasis. Some of them assumes $\boldsymbol{R}_1^N$ are hidden, $\mathbf{Y}_1^N$ are observed; while the others consider the total pair $\mathbf{H}_1^N$ are hidden states. Also, considering the state-space, it can be discrete classes or continues real values. One can decide where to apply the PMC and which state-space to choose according to the practical issue. In this chapter, we only discuss the case that $\boldsymbol{R}_1^N$ are hidden states and $\mathbf{Y}_1^N$ are observations.

## 1.2 PMC with discrete finite state-space

Let us consider the PMC with discrete finite state-space, like in classic HMC, hidden states $\mathbf{R}_1^N$ is a discrete process, each $R_n$ takes its values in discrete finite state-space $\boldsymbol{\Omega} = \{1, 2, \ldots, K\}$; and $\mathbf{Y}_1^N = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N)$ is a continuous observation with each $\mathbf{Y}_n$ taking its values in $\mathbb{R}^q$, $q$ represents the dimension of $\mathbf{Y}_n$. Benefiting from the Markovianity of $p\left(\mathbf{R}_1^N\left|\mathbf{Y}_1^N\right.\right)$, optimal restoration exists in PMC in spite weather $\mathbf{R}_1^N$ being Markovian or not [114], [84], [52].

### 1.2.1 Optimal restoration

Here we explain how the restorations (both filtering and smoothing) of PMC with discrete finite state-space run. We define that

$$\phi_n(j) = p\left(r_n = j\left|\mathbf{y}_1^N\right.\right), \tag{1.7}$$

$$\psi_n(j,k) = p\left(r_n = j, r_{n+1} = k\left|\mathbf{y}_1^N\right.\right), \tag{1.8}$$

where $j, k \in \mathbf{\Omega}$. The restoration can be calculated through the forward and backward probabilities by Baum's algorithm [12], [86], [40] from the structure of PMC. To iteratively compute (1.7)-(1.8), we adopt the "normalized" forward and backward probabilities [35]:

$$\alpha_n(j) = p\left(r_n = j\left|\mathbf{y}_1^n\right.\right), \tag{1.9}$$

$$\beta_n(j) = \frac{p\left(\mathbf{y}_{n+1}^N\left|r_n = j, \mathbf{y}_n\right.\right)}{p\left(\mathbf{y}_{n+1}^N\left|\mathbf{y}_1^n\right.\right)}. \tag{1.10}$$

These definition avoid the numerical underflow problem comparing to the original one [12], which computes the forward $p\left(\mathbf{y}_1^N, \mathbf{x}_n = j\right)$ and backward $p\left(\mathbf{y}_{n+1}^N\left|\mathbf{y}_n, \mathbf{x}_n = j\right.\right)$ recursively instead.

With the definitions above, we get forwardly the $\alpha_n$ through

$$\begin{aligned} \alpha_1(j) &= \frac{p\left(r_1 = j, \mathbf{y}_1\right)}{\sum\limits_{l\in\mathbf{\Omega}} p\left(r_1 = l, \mathbf{y}_1\right)}; \\ \alpha_n(j) &= \frac{\sum\limits_{l\in\mathbf{\Omega}} \alpha_{n-1}(l)\, p\left(r_n = j, \mathbf{y}_n\left|r_{n-1} = l, \mathbf{y}_{n-1}\right.\right)}{\sum\limits_{(l_1,l_2)\in\mathbf{\Omega}^2} \alpha_{n-1}(l_1)\, p\left(r_n = l_2, \mathbf{y}_n\left|r_{n-1} = l_1, \mathbf{y}_{n-1}\right.\right)}, \end{aligned} \tag{1.11}$$

which is the probability of filtering. While backwardly, we get the intermediate elements for smoothing

$$\begin{aligned} \beta_N(j) &= 1; \\ \beta_n(j) &= \frac{\sum\limits_{l\in\mathbf{\Omega}} \beta_{n+1}(l)\, p\left(r_{n+1} = l, \mathbf{y}_{n+1}\left|r_n = j, \mathbf{y}_n\right.\right)}{\sum\limits_{(l_1,l_2)\in\mathbf{\Omega}^2} \alpha_n(l_1)\, p\left(r_{n+1} = l_2, \mathbf{y}_{n+1}\left|r_n = l_1, \mathbf{y}_n\right.\right)}, \end{aligned} \tag{1.12}$$

where $1 \leq n < N$. So the smoothing probability, which is noted as $\phi_n(j)$ in (1.7) is given by

$$\phi_n(j) = \alpha_n(j)\,\beta_n(j), \tag{1.13}$$

and the joint posteriori probability $\psi_n(j,k)$ is given by

$$\begin{aligned}&\psi_n(j,k)\\&=\frac{\alpha_n(j)\,p\left(r_{n+1}=k,\mathbf{y}_{n+1}\left|r_n=j,\mathbf{y}_n\right.\right)\beta_{n+1}(k)}{\sum\limits_{(l_1,l_2)\in\mathbf{\Omega}^2}\alpha_n(l_1)\,p\left(r_{n+1}=l_2,\mathbf{y}_{n+1}\left|r_n=l_1,\mathbf{y}_n\right.\right)\beta_{n+1}(l_2)}.\end{aligned}\tag{1.14}$$

Most of the time, we deal the restoration under simple but practical assumption that the PMC is Gaussian stationary. It means that the probability of $p(\mathbf{h}_n,\mathbf{h}_{n+1})$ does not depend on $n$, and therefore, the distributions $p\left(\mathbf{y}_n,\mathbf{y}_{n+1}\left|r_n,r_{n+1}\right.\right)$, which can be written as $p\left(\mathbf{y}_1,\mathbf{y}_2\left|r_1,r_2\right.\right)$ are Gaussian given by:

$$p(\mathbf{h}_1,\mathbf{h}_2)=p(r_1=j,\mathbf{y}_1,r_2=k,\mathbf{y}_2)=p_{j,k}\;f_{j,k}(\mathbf{y}_1,\mathbf{y}_2),\tag{1.15}$$

$p_{j,k}$ is the abbreviation of $p(r_n=j,r_{n+1}=k)$ (we will keep using this abbreviation though out this dissertation), and

$$\begin{aligned}f_{j,k}(\mathbf{y}_1,\mathbf{y}_2)&=p\left(\mathbf{y}_1,\mathbf{y}_2\left|r_1=j,r_2=k\right.\right)\\&=\mathcal{N}\left(\mathbf{M}_{j,k}^{\mathbf{y}_1^2},\mathbf{\Gamma}_{j,k}^{\mathbf{y}_1^2}\right).\end{aligned}\tag{1.16}$$

$\mathbf{M}_{j,k}^{\mathbf{y}_1^2}$ and $\mathbf{\Gamma}_{j,k}^{\mathbf{y}_1^2}$ denote the mean and variance of the joint Gaussian distribution of $(\mathbf{y}_1,\mathbf{y}_2)$ conditionally on $(r_n=j,r_{n+1}=k)$ respectively.

In practice, sometimes, Gaussian distribution may be not always suitable, and a flexible shape of $f_{j,k}(\mathbf{y}_1,\mathbf{y}_2)$ can be defined by two marginal distributions and a dependence item known as copula [29], [98], [100], [124]. So the form of $f_{j,k}(\mathbf{y}_1,\mathbf{y}_2)$ writes according to this construction as

$$f_{j,k}(\mathbf{y}_1,\mathbf{y}_2)=f_{j,k}^{(l)}(\mathbf{y}_1)\,f_{k,j}^{(r)}(\mathbf{y}_2)\,c_{j,k}\left(F_{j,k}^{(l)}(\mathbf{y}_1),F_{k,j}^{(r)}(\mathbf{y}_2)\right),\tag{1.17}$$

in which $f_{j,k}^{(l)}(\mathbf{y}_1)=f^{(l)}(\mathbf{y}_1|r_n=j,r_{n+1}=k)$, $f_{k,j}^{(r)}(\mathbf{y}_2)=f^{(r)}(\mathbf{y}_2|r_{n+1}=k,r_n=j)$ are the two marginal densities, with $(l)$, $(r)$ specify the left or right margin respectively. The dependent structure $c_{j,k}(\cdot,\cdot)$ is the so called “copula”, and $F_{j,k}^{(l)}(\mathbf{y}_1)$ denotes the associated Cumulative Distribution Function (CDF) of $f_{j,k}^{(l)}(\mathbf{y}_1)$, and

$F_{k,j}^{(r)}(\mathbf{y}_2)$, the associated CDF of $f_{j,k}^{(r)}(\mathbf{y}_2)$. More details of the copula in $f_{j,k}(\mathbf{y}_1, \mathbf{y}_2)$ of PMC will be discussed later embedded in the Markov switching model which we are going to deal with in Chapter 3.

Of course, for any sub-case of PMC that has special dependence structure as described in previous Section, the restoration of the general PMC are suitable.

### 1.2.2 Unsupervised restoration

When applying the PMC to a real system, we have no idea what are the parameters of a suitable PMC. In this case, we often turn to the well known Expectation-Maximization (EM) principle for solution.

EM is an iterative method for searching maximum likelihood (ML) estimates of parameters in statistical models, when parts of the variables are missing (latent). The definition of EM principle was explained in [33], [97], but there are earlier works on this iterative method for exponential families [129], [128], [127], published as pointed out in [33]. The convergence of EM in [33] is revised by [135] later.

Back to the PMC that we are dealing with, $\left(\mathbf{r}_1^N, \mathbf{y}_1^N\right)$ is considered as the complete data for likelihood calculation, while $\mathbf{r}_1^N$ is latent, so the unsupervised Bayesian restoration based on ML can be handled with EM as already dealt in [82], [121] extended from the solution of HMC discussed in [123], [25]. It is well to be mention that, EM algorithm works well when the PMC is stationary. If not, the unsupervised restoration results can be quite poor, due to the bad match between the real and estimated models.

#### 1.2.2.1 EM for Gaussian stationary case

As proposed in [82], the EM method estimates the parameters of stationary Gaussian PMC by maximizing the likelihood function of incomplete data $\mathbf{Y}_1^N$ iteratively according to

$$\mathbf{\Theta}^{\mathbf{h}(i+1)} = \underset{\mathbf{\Theta}^{\mathbf{h}}}{\arg\max}\, \mathbb{E}_{\mathbf{\Theta}^{\mathbf{h}(i)}} \left[ \ln p_{\mathbf{\Theta}^{\mathbf{h}}} \left(\mathbf{H}_1^N\right) \left| \mathbf{y}_1^N \right. \right], \tag{1.18}$$

with $\mathbf{\Theta}^{\mathbf{h}} = \left(p_{j,k}, \mathbf{M}_{j,k}^{\mathbf{y}_1^2}, \mathbf{\Gamma}_{j,k}^{\mathbf{y}_1^2}\right)$, $1 \leq j, k \leq K$ and the index $i$ denotes the EM iteration. The EM algorithm constituted by the Expectation (E-step) and Maximization (M-

Step) iteratively run as follows:

***1) E-step:***

E-step calculates the expectation of the likelihood with current parameters $\mathbf{\Theta}^{\mathbf{h}(i)}$ (estimated from last M-step) which is actually simplified to get the update of $\psi_n(j,k)$ in (1.8). The computation is just the same as the optimal smoothing of this discrete state-space PMC which has been specified from equations (1.7) to (1.14).

***2) M-step:***

Then, the M-step searches to maximize (1.18) by taking derivative with respect to each parameter, which gives the following update equations for all the parameters:

$$\begin{aligned}
\hat{p}_{j,k} &= \frac{1}{N-1}\sum_{n=1}^{N-1}\psi_n(j,k);\\
\hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2} &= \frac{\sum\limits_{n=1}^{N-1}\psi_n(j,k)\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix}}{\sum\limits_{n=1}^{N-1}\psi_n(j,k)};\\
\hat{\mathbf{\Gamma}}_{j,k}^{\mathbf{y}_1^2} &= \frac{\sum\limits_{n=1}^{N-1}\psi_n(j,k)\left(\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix}-\hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2}\right)\left(\begin{bmatrix}\mathbf{y}_n\\ \mathbf{y}_{n+1}\end{bmatrix}-\hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2}\right)^{\mathsf{T}}}{\sum\limits_{n=1}^{N-1}\psi_n(j,k)}.
\end{aligned} \tag{1.19}$$

To initialize the iterations, one simple way is to use K-means clustering method to find the initial switches $\mathbf{R}_1^N = \mathbf{r}_1^N$, and calculate the initial values for parameters

$\mathbf{\Theta}^{\mathbf{h}(0)}$ by empirical estimations:

$$\begin{aligned}
\hat{p}_{j,k} =& \frac{\boldsymbol{Card}\left(j,k\right)}{N-1};\\
\hat{\mathbf{M}}^{\mathbf{y}_1^2}_{j,k} =& \frac{1}{\boldsymbol{Card}\left(j,k\right)} \sum_{n=1}^{N-1} \delta_n\left(j,k\right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} \right);\\
\hat{\mathbf{\Gamma}}^{\mathbf{y}_1^2}_{j,k} =& \frac{1}{\boldsymbol{Card}\left(j,k\right)}\\
& \sum_{n=1}^{N-1} \delta_n\left(j,k\right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} - \hat{\mathbf{M}}^{\mathbf{y}_1^2}_{j,k} \right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} - \hat{\mathbf{M}}^{\mathbf{y}_1^2}_{j,k} \right)^{\intercal}.
\end{aligned} \tag{1.20}$$

in which $\delta_n\left(j,k\right)$ denotes the function $\mathbb{1}\left(r_n = j, r_{n+1} = k\right)$, and $\boldsymbol{Card}\left(j,k\right) = \sum_{n=1}^{N-1} \delta_n\left(j,k\right)$. There are also some other initialization methods which could be applied as discussed and compared in [15].

Finally, EM is stopped after the change of the likelihood between two iterations is considered small enough (one can set a threshold to specify the convergence).

#### 1.2.2.2 ICE for stationary case

When it comes to non-Gaussian case, direct derivative of parameters from the form of ML may be complex or not possible. As an alternative method, “iterative conditional estimation” (ICE) was proposed by [113] for solving the fundamental limitation of EM, since it uses also the complete data $\mathbf{H}_1^N$, but the computation of its likelihood is not necessary. The efficiency and convergence of ICE have been verified with application in statistical image segmentation by [83], [111], [19] and [31].

ICE assumes that there exists an estimator for $\mathbf{H}_1^N$ noted by $\hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{H}_1^N\right) = \hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ with hidden data $\mathbf{R}_1^N$ that one wants to recover. The natural best estimator, which considers the minimum mean square error noted by $\mathbb{E}_{\mathbf{\Theta}^{\mathbf{h}}}\left[\hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{H}_1^N\right)|\mathbf{y}_1^N\right]$ is a conditional expectation on $\mathbf{\Theta}^{\mathbf{h}}$, while $\mathbf{\Theta}^{\mathbf{h}}$ is unknown. However, we have iterative method to approach it

$$\mathbf{\Theta}^{\mathbf{h}(i+1)} = \mathbb{E}_{\mathbf{\Theta}^{\mathbf{h}(i)}}\left[\hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{H}_1^N\right)|\mathbf{y}_1^N\right]. \tag{1.21}$$

We see that, when $\hat{\mathbf{\Theta}}^{\mathbf{h}}$ is chosen to be an ML estimator, equation (1.21) becomes

$$\mathbf{\Theta}^{\mathbf{h}(i+1)} = \mathbb{E}_{\mathbf{\Theta}^{\mathbf{h}(i)}}\left[\arg\max_{\mathbf{\Theta}^{\mathbf{h}}} \ln p_{\mathbf{\Theta}^{\mathbf{h}}}\left(\mathbf{H}_1^N\right)\left|\mathbf{y}_1^N\right.\right], \tag{1.22}$$

which is identical to EM if the expectation and the log-likelihood maximization can be exchanged, and this occurs when the distribution of complete data belongs to an exponential family. Therefore, EM algorithm can be taken as a particular case of ICE for this kind of canonical parameterization structures. More discussion about the equivalence of ICE and EM can be found in [32].

The advantage of ICE is that, if we can compute the conditional distribution of $\left(\mathbf{R}_1^N|\mathbf{y}_1^N\right)$ at step $i$ but not the expectation in (1.21) analytically, we can simulate the realization $\mathbf{r}_1^N$ of $\mathbf{R}_1^N$ according to $p\left(\mathbf{R}_1^N\left|\mathbf{y}_1^N\right.\right)$, with the current parameter $\mathbf{\Theta}^{\mathbf{h}(i)}$ (it is called the Random Imputation Principle (RIP) in [27]), and then $\theta_{i+1}$ can be approximated empirically, thanks to the law of large numbers as

$$\mathbf{\Theta}^{\mathbf{h}(i+1)} = \frac{1}{\mathcal{M}}\left[\hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{r}_1^N\right)_1 + \hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{r}_1^N\right)_2 + \cdots + \hat{\mathbf{\Theta}}^{\mathbf{h}}\left(\mathbf{r}_1^N\right)_{\mathcal{M}}\right], \tag{1.23}$$

where $\left(\mathbf{r}_1^N\right)_1, \cdots, \left(\mathbf{r}_1^N\right)_{\mathcal{M}}$ are $\mathcal{M}$ realizations of $\mathbf{R}_1^N$.

Let us pay attention that, there is another similar simulation based alternative method for EM, which is called stochastic EM (SEM) [99], [85], [95], [27]. SEM takes realization (stochastic) step after E-step only once, and M-step which defining $\mathbf{\Theta}^{\mathbf{h}(i+1)}$ is given by solve the ML function with the realized complete data. We can see that, SEM is also a special case of ICE when $\mathcal{M} = 1$ and ML is chosen to be the $\hat{\mathbf{\Theta}}^{\mathbf{h}}$.

#### 1.2.2.3 Principles for infering hidden states

There are several criterions to infer the hidden $\mathbf{R}_1^N$ from the filtering probabilities $p\left(r_n\left|\mathbf{y}_1^n\right.\right)$ and smoothing ones $p\left(r_n\left|\mathbf{y}_1^N\right.\right)$. The Bayesian MPM (Maximum Posterior Mode) criterion, which maximizes the posteriors is commonly used according to the computation of

$$\hat{r}_n = \arg\max_j \phi_n(j), \tag{1.24}$$

with $j \in \mathbf{\Omega}$ for $n$ from $\{1, \cdots, N\}$. And another Bayesian criterion is the MAP (Maximum A Posteriori estimation) defined as a regularization of ML estimation that

$$\hat{\mathbf{r}}_1^N = \arg\max_{r_n \in \mathbf{\Omega}} p\left(\mathbf{r}_1^N \left| \mathbf{y}_1^N \right.\right) = \arg\max_{r_n \in \mathbf{\Omega}} p\left(\mathbf{r}_1^N, \mathbf{y}_1^N\right). \tag{1.25}$$

In this dissertation, we always use MPM to obtain the restoration of $\mathbf{R}_1^N$ due to its simplicity.

We address here an experiment to show the performance of unsupervised restoration methods on HMC-DN as a groundwork, since it is a partial structure of the switching Markov models we deal with later.

As defined in HMC-DN, $\mathbf{R}_1^N$ is Markov, and we set each $R_n$ adopts simply two possible values, which means that $\mathbf{\Omega} = \{1, 2\}$. The probabilities of $\mathbf{R}_1^N$, which has already appeared in (1.15) are defined by $p_{1,2} = p_{2,1} = 0.05$ and $p_{1,1} = p_{2,2} = 0.45$. The dependence of $p\left(\mathbf{y}_1^N \left| \mathbf{r}_1^N \right.\right)$ is set to be Gaussian with the auto-regressive relation

$$\mathbf{Y}_{n+1} = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}\left(\mathbf{R}_n^{n+1}\right)\mathbf{Y}_n + \boldsymbol{\mathcal{B}}^{\mathbf{yy}}\left(\mathbf{R}_n^{n+1}\right)\mathbf{V}_{n+1}, \tag{1.26}$$

in which $\mathbf{V}_{n+1}$ is a standard normal white noise written as $\mathbf{V}_{n+1} \sim \mathcal{N}(0,1)$, and initially, the $\mathbf{Y}_1 \sim \mathcal{N}(0,1)$ also. The parameters $\boldsymbol{\mathcal{F}}^{\mathbf{yy}}\left(\mathbf{R}_n^{n+1}\right)$ and $\boldsymbol{\mathcal{B}}^{\mathbf{yy}}\left(\mathbf{R}_n^{n+1}\right)$ are assigned as $\boldsymbol{\mathcal{F}}^{\mathbf{yy}}(R_n = 0, R_{n+1} = 0) = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(R_n = 1, R_{n+1} = 0) = 0.4$, $\boldsymbol{\mathcal{F}}^{\mathbf{yy}}(R_n = 0, R_{n+1} = 1) = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(R_n = 1, R_{n+1} = 1) = 0.9$ and $\boldsymbol{\mathcal{B}}^{\mathbf{yy}}(R_n, R_{n+1}) = \sqrt{1 - \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(R_n, R_{n+1})^2}$. Under this setting, we have the conditional means of $\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n, r_{n+1}\right)$ all 0, variance all 1, and the covariance $cov\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n, r_{n+1}\right) = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(r_n, r_{n+1})$. 2000 samples are simulated according to the model setting, then, the supervised filtering, smoothing, and unsupervised smoothing through EM and ICE are applied on the observations for restoration. In particular, the ICE applied here adopts the classic empirical estimation of the moments as $\hat{\mathbf{\Theta}}^{\mathbf{h}}$, based on hidden state realizations $\mathbf{r}_1^N$, and $\mathcal{M}$ set to one in (1.23), which runs

$$\begin{aligned}
\hat{p}_{j,k} =& \frac{1}{N-1} \sum_{n=1}^{N-1} \psi_n\left(j,k\right);\\
\hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2} =& \frac{1}{\boldsymbol{Card}\left(j,k\right)} \sum_{n=1}^{N-1} \delta_n\left(j,k\right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} \right);\\
\hat{\mathbf{\Gamma}}_{j,k}^{\mathbf{y}_1^2} =& \frac{1}{\boldsymbol{Card}\left(j,k\right)}\\
& \sum_{n=1}^{N-1} \delta_n\left(j,k\right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} - \hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2} \right) \left( \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} - \hat{\mathbf{M}}_{j,k}^{\mathbf{y}_1^2} \right)^{\intercal},
\end{aligned} \tag{1.27}$$

where $\delta_n\left(j,k\right)$ and $\boldsymbol{Card}\left(j,k\right)$ are defined the same as in (1.20).

The average result of 100 Monte-Carlo experiments are reported in Table 1.1 As EM and ICE make use of the entire $\mathbf{Y}_1^N$, we only report their smoothing result in the Table. The error ratio tendencies of EM and ICE of both one instance and average of 100 independent experiments are displayed in figure 1.2. We find that with estimator based on realization, ICE is more fluctuating than EM, but the two algorithms perform nearly the same under the setting of this experiment. The fluctuation may be smoothen with the increasing value of $\mathcal{M}$ which is only set to 1 in this example.

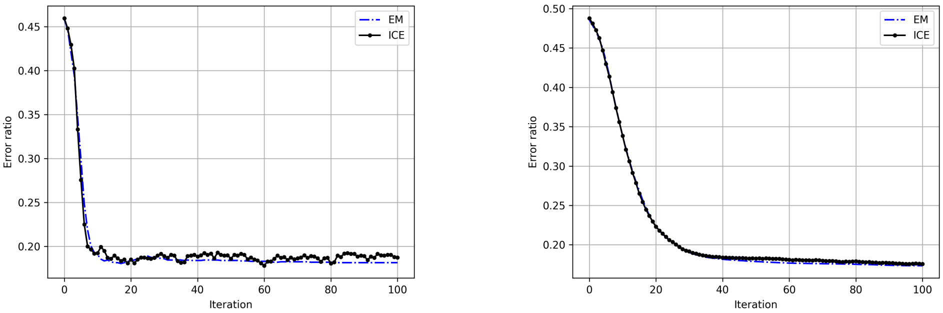
Table 1.1: Restoration error ratio of all methods (average of 100 independent experiments) .

| | Optimal filtering | Optimal smoothing | EM | ICE |
|---|---|---|---|---|
| Error Ratio | 0.196 | 0.173 | 0.189 | 0.180 |

## 1.3 PMC with continuous state-space

The continuous state-space PMC has the hidden state takes its value in a continuous real space. To distinguish it from the discrete state-space PMC, we take $\mathbf{X}_1^N = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N)$ instead of $\mathbf{R}_1^N$ to denote the continuous value hidden state, where each $\mathbf{X}_n$ takes its value in $\mathbb{R}^s$, and "$s$" being the dimension of $\mathbf{X}_n$.

A commonly used example is when this continuous state-space PMC meets the



(a) One instance.

(b) Average of 100 experiments.

Figure 1.2: Error ratio tendency with iterations.

particular Gaussian linear case, which is called "Linear Gaussian Pairwise Markov Model" [116] or "Gaussian Pairwise Markov Model" (GPMM) [1], written as

$$\begin{bmatrix}\mathbf{X}_{n+1}\\ \mathbf{Y}_{n+1}\end{bmatrix}=\underbrace{\begin{bmatrix}\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1} & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}_{n+1}\\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1} & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}_{n+1}\end{bmatrix}}_{\boldsymbol{\mathcal{F}}_{n+1}}\begin{bmatrix}\mathbf{X}_{n}\\ \mathbf{Y}_{n}\end{bmatrix}+\underbrace{\begin{bmatrix}\boldsymbol{\mathcal{B}}^{\mathbf{xx}}_{n+1} & \boldsymbol{\mathcal{B}}^{\mathbf{xy}}_{n+1}\\ \boldsymbol{\mathcal{B}}^{\mathbf{yx}}_{n+1} & \boldsymbol{\mathcal{B}}^{\mathbf{yy}}_{n+1}\end{bmatrix}}_{\boldsymbol{\mathcal{B}}_{n+1}}\begin{bmatrix}\mathbf{U}_{n+1}\\ \mathbf{V}_{n+1}\end{bmatrix}, \tag{1.28}$$

in which $\boldsymbol{\mathcal{F}}_{n+1}$ and $\boldsymbol{\mathcal{B}}_{n+1}$ are parameters of the regime. $\mathbf{W}_{n+1}=\left[\mathbf{U}^{\intercal}_{n+1},\mathbf{V}^{\intercal}_{n+1}\right]^{\intercal}$ are noises which follow independently the standard normal distribution, and are independent of $\mathbf{X}_1$, $\mathbf{Y}_1$. Under Gaussian assumption that $\mathbf{X}_1$, $\mathbf{Y}_1$ and $\mathbf{W}_1$ are all Gaussian, the pair $\left(\mathbf{X}_1^N,\mathbf{Y}_1^N\right)$ is then a Gaussian process. Consequently, $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^n\right.\right)$, $p\left(\mathbf{x}_{n}\left|\mathbf{y}_1^n\right.\right)$, $p\left(\mathbf{x}_{n}\left|\mathbf{y}_1^N\right.\right)$ are all Gaussian.

The continuous state-space Gaussian linear HMC known as Hidden Gaussian Markov Model (HGMM) [4], [9], [77] often written in the form

$$\begin{aligned}\mathbf{X}_{n+1}&=\mathbf{A}_{n+1}\mathbf{X}_n+\mathbf{B}_{n+1}\mathbf{U}_{n+1};\\ \mathbf{Y}_{n+1}&=\mathbf{C}_{n+1}\mathbf{X}_{n+1}+\mathbf{D}_{n+1}\mathbf{V}_{n+1},\end{aligned} \tag{1.29}$$

with matrices $\mathbf{A}_{n+1}$, $\mathbf{B}_{n+1}$, $\mathbf{C}_{n+1}$, $\mathbf{D}_{n+1}$ defining the linear functions. $\mathbf{U}_{n+1}$, $\mathbf{V}_{n+1}$ are standard normal white noises which independent from each other and independent from $\mathbf{X}_1$ and $\mathbf{Y}_1$. However, as HMC is a sub-model of PMC, spontaneously, HGMM is a sub-model of GPMM. Just with some parameters set to be 0, HGMM can be rewritten as

$$\begin{bmatrix} \mathbf{X}_{n+1} \\ \mathbf{Y}_{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xx}} & \mathbf{0} \\ \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{yx}} & \mathbf{0} \end{bmatrix}}_{\boldsymbol{\mathcal{F}}_{n+1}} \begin{bmatrix} \mathbf{X}_{n} \\ \mathbf{Y}_{n} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{xx}} & \mathbf{0} \\ \boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{yx}} & \boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{yy}} \end{bmatrix}}_{\boldsymbol{\mathcal{B}}_{n+1}} \begin{bmatrix} \mathbf{U}_{n+1} \\ \mathbf{V}_{n+1} \end{bmatrix}, \tag{1.30}$$

where $\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xx}} = \mathbf{A}_{n+1}$, $\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{yx}} = \mathbf{C}_{n+1}\mathbf{A}_{n+1}$, $\boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{xx}} = \mathbf{B}_{n+1}$, $\boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{yx}} = \mathbf{C}_{n+1}\mathbf{B}_{n+1}$, and $\boldsymbol{\mathcal{B}}_{n+1}^{\mathbf{yy}} = \mathbf{D}_{n+1}$. As in PMC, the hidden $\mathbf{X}_1^N$ in GPMM can be Markov or not [84].

### 1.3.1 Restoration of continuous state-space PMC

The optimal restoration for the continuous state-space PMC in general way can be derived by the following steps.

One step ahead prediction:

$$p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^n\right.\right) = \int p\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) d\mathbf{x}_n, \tag{1.31}$$

so the forward filtering is

$$\begin{aligned} p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right) &= \frac{p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{y}_1^n\right.\right)}{p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_1^n\right.\right)} \\ &= \frac{\int p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) d\mathbf{x}_n}{\int p\left(\mathbf{y}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) d\mathbf{x}_n} \end{aligned}. \tag{1.32}$$

Benefit from the pairwise structure, we have

$$p\left(\mathbf{x}_n\left|\mathbf{x}_{n+1}, \mathbf{y}_1^{n+1}\right.\right) = p\left(\mathbf{x}_n\left|\mathbf{x}_{n+1}, \mathbf{y}_1^N\right.\right), \tag{1.33}$$

and the backward smoothing can be reached by

$$\begin{aligned} p\left(\mathbf{x}_n\left|\mathbf{y}_1^N\right.\right) &= \int \frac{p\left(\mathbf{x}_n, \mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right)}{p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right)} p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^N\right.\right) d\mathbf{x}_{n+1} \\ &= \int \frac{p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_1^n\right.\right)}{p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n\right.\right) p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right)} p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^N\right.\right) d\mathbf{x}_{n+1}, \\ &= p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) \int \frac{p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_1^n\right.\right) p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^N\right.\right)}{p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{y}_1^n\right.\right)} d\mathbf{x}_{n+1} \end{aligned} \tag{1.34}$$

with

$$p\left(\mathbf{x}_{n+1},\mathbf{y}_{n+1}\left|\mathbf{y}_1^n\right.\right)=\int p\left(\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right)p\left(\mathbf{x}_{n+1},\mathbf{y}_{n+1}\left|\mathbf{x}_n,\mathbf{y}_n\right.\right)d\mathbf{x}_n. \tag{1.35}$$

Then (1.31)-(1.34) are computable in Gaussian case.

The unsupervised restoration based on EM for GPMM has been developed by [5] and the robustness strengthened by [101] through QR decompositions. Moreover, a partial supervised solution is given by [102]. As we will depict the extension model of GPMM with switches in next chapter, GPMM will become its sub-case with zero switch. For not duplicating the state of method, the unsupervised restoration of GPMM can be referred in next Section removing the switch symbols.

## 1.4 Conclusion

This chapter presents the principle of Pairwise Markov Chains (PMCs) and their restoration algorithms, whatever supervised or unsupervised based on Expectation-Maximization (EM) and Iterative Conditional Estimation (ICE) principles for parameter estimation. The PMC is a generalization of the classic Hidden Markov Chain (HMC). Definition, property and advantage of PMC are described in details in the beginning of this chapter. Two cases of hidden states (discrete finite and continuous) in PMC are specified then, with the derivation of both supervised and unsupervised restorations. In addition, an example of unsupervised restoration of the discrete finite state-space PMC is reported to show the performance of all restoration methods on the commonly used Gaussian case.

PMC is the basic of the switching Markov model we handle in this thesis. Actually, the special switching Markov model we are going to deal with, is a Triplet Markov Chain (TMC) [117] developed from the GPMM (Gaussian continuous state-space PMC) with essential consideration of switching regime. In practice, this "switching" regime makes the Markov chain owns the ability to describe the dramatic change of the auto-regression and better suits for approaching the non-linear systems. This chapter paves the way for finding the solution of the main subject which we are facing to and the methods described will be the reference of the new methods we developed for switching Markov models in following chapters.

Chapter 2

# Optimal and approximated restorations in Gaussian linear Markov switching models

As described in previous Chapter, the simplest model for the distribution of $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ which allows fast Bayesian linear processing, is the classic HGMM defined by Gaussian distribution $p\left(\mathbf{x}_1\right)$ of $\mathbf{X}_1$, the Markov transitions $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n \right.\right)$ and simple dependence $p\left(\mathbf{y}_n \left| \mathbf{x}_n \right.\right)$. The HGMM has been later extended to GPMM defined by Gaussian distribution $p\left(\mathbf{x}_1, \mathbf{y}_1\right)$ of $(\mathbf{X}_1, \mathbf{Y}_1)$ and the pairwise Markov transitions $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n \right.\right)$. Optimal filtering and smoothing remains workable in GPMM, while comparing to HGMM, it incorporates more complex dependence between the stochastic variables.

Let us now extend the previous models by introducing a hidden process to model the "switches" (also called "jump"). We consider $\mathbf{X}_1^N = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, $\mathbf{R}_1^N = \{R_1, R_2, \ldots, R_N\}$, and $\mathbf{Y}_1^N = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N\}$, each $\mathbf{X}_n$, $R_n$, $\mathbf{Y}_n$ takes its value in $\mathbb{R}^s$, $\mathbf{\Omega} = \{1, 2, \ldots, K\}$, and $\mathbb{R}^q$ respectively. $\mathbf{Y}_1^N$ is observed, and the problem is to estimate the hidden realizations of $\mathbf{X}_1^N$ from only $\mathbf{Y}_1^N$. Introducing discrete switches $\mathbf{R}_1^N$ in the plain models mentioned before is of interest for at least two aspects. Firstly, this can model stochastic regime changes. It is to say that it allows random changes in the parameters which define the plain HGMM and GPMM. Secondly, when considering a Markov triplet $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ such that $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is a GPMM conditionally on $\mathbf{R}_1^N$, the distribution of $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ becomes a Gaussian mixture distribution, which is likely to approximate non-Gaussian non-linear systems. Such situation has been successfully considered in [62], where it shows that any stationary or asymptomatically stationary Markov system can be approximated by a Gaussian switching system once a method to simulate realizations of the system is available.

Introducing discrete switches $\mathbf{R}_1^N$ in the classic HGMM leads to the following Conditionally Gaussian Linear State-space Model (CGLSSM) [26]:

$$\begin{aligned} &\mathbf{R}_1^N \text{ is Markov,} \\ &\mathbf{X}_{n+1} = \mathbf{A}_{n+1}(R_{n+1})\mathbf{X}_n + \mathbf{B}_{n+1}(R_{n+1})\mathbf{U}_{n+1}, \\ &\mathbf{Y}_{n+1} = \mathbf{C}_{n+1}(R_{n+1})\mathbf{X}_{n+1} + \mathbf{D}_{n+1}(R_{n+1})\mathbf{V}_{n+1}. \end{aligned} \tag{2.1}$$

$\mathbf{A}_{n+1}(R_{n+1})$, $\mathbf{B}_{n+1}(R_{n+1})$, $\mathbf{C}_{n+1}(R_{n+1})$, $\mathbf{D}_{n+1}(R_{n+1})$ are matrices conditionally on $R_n$ of dimension $s \times s$, $s \times s$, $q \times s$ and $q \times q$ respectively. $\mathbf{U}_{n+1}$, $\mathbf{V}_{n+1}$ are random variables distributed according to standard Normal distribution. The CGLSSM is also known as "Linear system with jump parameters" [133]; "Switching Linear Dynamic Systems" [77], [125]; "Jump Markov Linear Systems" [7]; "Switching Linear State-space Models" [6]; and "Conditional Linear Gaussian Models" [93] applied in tracking, speech feature mapping and biomedical engineering, *etc.*

While $\mathbf{R}_1^N$ is hidden, in CGLSSM, computing conditional mean estimates of hidden states is infeasible as it involves a cost that grows exponentially with the number of observations [115]. The restoration is often approached by Markov Chain Monte-Carlo (MCMC) methods [42], [43]. When it comes to the unsupervised case, recent works on the parameter estimation of CGLSSM or the more general Jump Markov System (JMS), combine EM with Sequential Monte-Carlo (SMC) methods to do the parameter estimation [54], [108]. As MCMC methods can approximate properly the target distribution with large sample numbers, the restoration performance can be quite satisfactory. However, the computation consumption of MCMC based methods increases with sample numbers when high accuracy is required and can meet degeneracy problems [71], [64].

We consider the recent model extended from CGLSSM, called Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM) [1], which introduces the switches $\mathbf{R}_1^N$ in GPMM. The aim of this chapter is to propose a parameter estimation method only from the observed $\mathbf{Y}_1^N$ for stationary CGPMSM, and perform unsupervised restoration without making use of the MCMC methods. The remaining of this chapter is organized as follows. In Section 2.1, we recall the definition of the special JMLS family called CGPMSM which we are interested in. Next, we derive the optimal filtering and smoothing of the "Conditionally Gaussian Observed Markov Switching Model" (CGOMSM), which is a sub-model of CGPMSM. Then, for the restoration of general stationary CGPMSM, which will be discussed in following Sections, we detail its parameterizations. Two experimental series on simulated data are conducted in this Section to verify the supervised filtering and smoothing for CGOMSM, and their ability to restore the close CGPMSM as sub-optimal solution. Section 2.2 extends the EM algorithm for parameter estimation in GPMM to Switching EM which works on parameter estimation on CGPMSM with known switches. Then, a parameter estimation method for CGPMSM with unknown switches called Double EM is proposed, which applies twice the EM principle incorporating the Switching EM. Meanwhile, the shortcoming of the proposed Double EM is also pointed out. Section 2.3 proposes two restoration approaches in CGPMSM, one is with partial mild modification of parameters and the other is based on EM principle. Then, several unsupervised strategies are produced by fusing the Double EM parameter estimation and restoration approaches. Two Series of experiments focus on different observation means and varying noise levels are conducted to test the proposed Double EM method and study the performance of all proposed unsupervised restoration methods with comparison to several existing supervised restoration methods under the influence of these two factors. Finally, the work of this Chapter is concluded in Section 2.4.

## 2.1 Filtering and smoothing

### 2.1.1 Definition of CGPMSM and CGOMSM

The CGPMSM is a switching Gaussian linear dynamic stochastic system defined by:

$$\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N) \text{ is Markov with } p\left(r_{n+1} \left| \mathbf{x}_n, r_n, \mathbf{y}_n\right.\right) = p\left(r_{n+1} \left| r_n\right.\right), \tag{2.2}$$

$$\begin{aligned}\underbrace{\begin{bmatrix}\mathbf{X}_{n+1} - \mathbf{M}^{\mathbf{x}}_{n+1}(R_{n+1}) \\ \mathbf{Y}_{n+1} - \mathbf{M}^{\mathbf{y}}_{n+1}(R_{n+1})\end{bmatrix}}_{\mathbf{Z}_{n+1}} &= \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}_{n+1}(\mathbf{R}_n^{n+1}) \\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{F}}_{n+1}(\mathbf{R}_n^{n+1})} \underbrace{\begin{bmatrix}\mathbf{X}_n - \mathbf{M}^{\mathbf{x}}_n(R_n) \\ \mathbf{Y}_n - \mathbf{M}^{\mathbf{y}}_n(R_n)\end{bmatrix}}_{\mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}_n} \\ &+ \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{B}}^{\mathbf{xx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{B}}^{\mathbf{xy}}_{n+1}(\mathbf{R}_n^{n+1}) \\ \boldsymbol{\mathcal{B}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{B}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{B}}_{n+1}(\mathbf{R}_n^{n+1})} \underbrace{\begin{bmatrix}\mathbf{U}_{n+1} \\ \mathbf{V}_{n+1}\end{bmatrix}}_{\mathbf{W}_{n+1}}.\end{aligned} \tag{2.3}$$

$\mathbf{X}_1$, $\mathbf{Y}_1$, $R_1$ are given following Gaussian distribution $p\left(\mathbf{x}_1, \mathbf{y}_1 \left| r_1\right.\right)$ and $p\left(r_1\right)$ respectively. The hidden switches $\mathbf{R}_1^N$ is a Markov chain, which comes from $p\left(r_{n+1} \left| \mathbf{x}_n, r_n, \mathbf{y}_n\right.\right) = p\left(r_{n+1} \left| r_n\right.\right)$. $\mathbf{U}_1^N$ and $\mathbf{V}_1^N$ note the mutually independent centered Gaussian noise with unit variance-covariance matrix which are also assumed independent from $\mathbf{R}_1^N$. The system parameters $\boldsymbol{\mathcal{F}}_{n+1}(\mathbf{r}_n^{n+1})$ and $\boldsymbol{\mathcal{B}}_{n+1}(\mathbf{r}_n^{n+1})$ depend on the switches $\mathbf{r}_n^{n+1} = (r_n, r_{n+1})$[1], so the couple $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is Markovian and Gaussian conditionally on $\mathbf{R}_1^N$. $\mathbf{M}^{\mathbf{x}}_n(r_n)$ and $\mathbf{M}^{\mathbf{y}}_n(r_n)$ are the means of $\mathbf{X}_n$ and $\mathbf{Y}_n$ conditionally on $r_n$, we denote $\mathbf{M}^{\mathbf{z}}_n(r_n) = [\mathbf{M}^{\mathbf{x}}_n(r_n), \mathbf{M}^{\mathbf{y}}_n(r_n)]^{\intercal}$. The original model defined by [1] does not consider $\mathbf{M}^{\mathbf{x}}_n(r_n)$ and $\mathbf{M}^{\mathbf{y}}_n(r_n)$, or we can say they are set to be both zero. (2.3) can be concisely written as

$$\mathbf{Z}_{n+1} - \mathbf{M}^{\mathbf{z}}_{n+1}(R_{n+1}) = \boldsymbol{\mathcal{F}}_{n+1}(\mathbf{R}_n^{n+1})\left(\mathbf{Z}_n - \mathbf{M}^{\mathbf{z}}_n(R_n)\right) + \boldsymbol{\mathcal{B}}_{n+1}(\mathbf{R}_n^{n+1})\mathbf{W}_{n+1}. \tag{2.4}$$

Like HGMM can be taken as a special case under GPMM. Under the general family of CGPMSM, the classic CGLSSM without the consideration of means can

[1] this written simplification will be applied to other symbols though whole text

be represented by setting several zeros and simplifying the correspondence on switch to the present state of $R_{n+1}$ in the (2.3).

$$\underbrace{\begin{bmatrix}\mathbf{X}_{n+1}\\ \mathbf{Y}_{n+1}\end{bmatrix}}_{\mathbf{Z}_{n+1}} = \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(R_{n+1}) & \mathbf{0}\\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(R_{n+1}) & \mathbf{0}\end{bmatrix}}_{\boldsymbol{\mathcal{F}}_{n+1}(R_{n+1})}\underbrace{\begin{bmatrix}\mathbf{X}_{n}\\ \mathbf{Y}_{n}\end{bmatrix}}_{\mathbf{Z}_{n}} + \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{B}}^{\mathbf{xx}}_{n+1}(R_{n+1}) & \mathbf{0}\\ \boldsymbol{\mathcal{B}}^{\mathbf{yx}}_{n+1}(R_{n+1}) & \boldsymbol{\mathcal{B}}^{\mathbf{yy}}_{n+1}(R_{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{B}}_{n+1}(R_{n+1})}\underbrace{\begin{bmatrix}\mathbf{U}_{n+1}\\ \mathbf{V}_{n+1}\end{bmatrix}}_{\mathbf{W}_{n+1}}. \tag{2.5}$$

Compare to the classic CGLSSM (2.1), we have the congruent relationship:

$$\begin{aligned}
\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(R_{n+1}) &= \mathbf{A}_{n+1}(R_{n+1});\\
\boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(R_{n+1}) &= \mathbf{C}_{n+1}(R_{n+1})\mathbf{A}_{n+1}(R_{n+1});\\
\boldsymbol{\mathcal{B}}^{\mathbf{xx}}_{n+1}(R_{n+1}) &= \mathbf{B}_{n+1}(R_{n+1});\\
\boldsymbol{\mathcal{B}}^{\mathbf{yx}}_{n+1}(R_{n+1}) &= \mathbf{C}_{n+1}(R_{n+1})\mathbf{B}_{n+1}(R_{n+1});\\
\boldsymbol{\mathcal{B}}^{\mathbf{yy}}_{n+1}(R_{n+1}) &= \mathbf{D}_{n+1}(R_{n+1}).
\end{aligned} \tag{2.6}$$

Conditionally on $\mathbf{R}_1^N$, $\mathbf{X}_1^N$ is linear Gaussian and Markovian, and the distribution of $\mathbf{Y}_1^N$ conditionally to $\mathbf{X}_1^N$ is simple in CGLSSM. Thus given $\mathbf{R}_1^N = \mathbf{r}_1^N$, the couple $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ degenerates as a HGMM, in which the classical optimal Kalman filter and smoother can be applied. But in case that $\mathbf{R}_1^N$ is unknown, although CGLSSM appears as "natural" switching Gaussian system, neither optimal filtering nor smoothing can be derived.

Of course, the general CGPMSM extended from GPMM also meets this tough problem. But let us pay attention to the pairwise structure of CGPMSM. Actually, the observations and hidden states play symmetrical roles, we can take arbitrarily any of these two as the other's noisy version. If we inverse the roles of $\mathbf{X}_1^N$ and $\mathbf{Y}_1^N$ in CGLSSM, both $p\left(r_n\left|\mathbf{x}_1^N\right.\right)$ and $p\left(\mathbf{y}_n\left|r_n, \mathbf{x}_1^N\right.\right)$ become computable. Based on this idea, the sub-family of CGPMSM named "Conditionally Gaussian Observed Markov Switching Model" (CGOMSM) has also been proposed in [1], [2]. The CGOMSM is with a fixed $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) = \mathbf{0}$ in CGPMSM verifying that:

$$\begin{bmatrix} \mathbf{X}_{n+1} - \mathbf{M}^{\mathbf{x}}_{n+1}(R_{n+1}) \\ \mathbf{Y}_{n+1} - \mathbf{M}^{\mathbf{y}}_{n+1}(R_{n+1}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}_{n+1}(\mathbf{R}_n^{n+1}) \\ \mathbf{0} & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{X}_n - \mathbf{M}^{\mathbf{x}}_n(R_n) \\ \mathbf{Y}_n - \mathbf{M}^{\mathbf{y}}_n(R_n) \end{bmatrix}$$
$$+ \begin{bmatrix} \boldsymbol{\mathcal{B}}^{\mathbf{xx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{B}}^{\mathbf{xy}}_{n+1}(\mathbf{R}_n^{n+1}) \\ \boldsymbol{\mathcal{B}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{B}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{U}_{n+1} \\ \mathbf{V}_{n+1} \end{bmatrix}. \tag{2.7}$$

The prominent advantage of this CGOMSM over CGLSSM is that optimal filtering and smoothing are feasible, as the observations and switches processes $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ becomes a pairwise Markov chain, which is of importance in some real-data applications.

### 2.1.2 Optimal restoration in CGOMSM

Firstly, let us recall the classic optimal filtering and smoothing equations. Optimal filtering consists in computing $\mathbb{E}[\mathbf{X}_n | \mathbf{y}_1^n]$ for each $n = 1, \ldots, N$. In presence of switches

$$\mathbb{E}[\mathbf{X}_n | \mathbf{y}_1^n] = \sum_{r_n} p(r_n | \mathbf{y}_1^n) \, \mathbb{E}[\mathbf{X}_n | r_n, \mathbf{y}_1^n], \tag{2.8}$$

and the optimal smoothing of switching Markov models is classically calculated by

$$\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_1^N \right.\right] = \sum_{r_n} p\left(r_n \left| \mathbf{y}_1^N \right.\right) \mathbb{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^N \right.\right]. \tag{2.9}$$

Profiting from the special structure of CGOMSM, $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a Markov chain. This allows the exact computation of $p(r_n | \mathbf{y}_1^n)$ in optimal filtering which is not possible in classical Markov switching models, which needs to be approximated by Monte-Carlo methods for example.

To make the derivation more legible, we rewrite the expression of CGPMSM

(2.3) for better showing the dependences in the switching regimes to the form

$$\underbrace{\begin{bmatrix}\mathbf{X}_{n+1}\\ \mathbf{Y}_{n+1}\end{bmatrix}}_{\mathbf{Z}_{n+1}} = \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}_{n+1}(\mathbf{R}_n^{n+1})\\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{F}}_{n+1}(\mathbf{R}_n^{n+1})} \underbrace{\begin{bmatrix}\mathbf{X}_{n}\\ \mathbf{Y}_{n}\end{bmatrix}}_{\mathbf{Z}_{n}} + \underbrace{\begin{bmatrix}\boldsymbol{\omega}^{\mathbf{x}}_{n+1}\\ \boldsymbol{\omega}^{\mathbf{y}}_{n+1}\end{bmatrix}}_{\boldsymbol{\omega}_{n+1}} + \underbrace{\begin{bmatrix}\mathbf{N}^{\mathbf{x}}_{n+1}(\mathbf{R}_n^{n+1})\\ \mathbf{N}^{\mathbf{y}}_{n+1}(\mathbf{R}_n^{n+1})\end{bmatrix}}_{\mathbf{N}^{\mathbf{z}}_{n+1}(\mathbf{R}_n^{n+1})}, \tag{2.10}$$

in which the noises follow the Normal distribution:

$$\boldsymbol{\omega}_{n+1} \sim \mathcal{N}\left(\mathbf{0}, \underbrace{\begin{bmatrix}\boldsymbol{\mathcal{Q}}^{\mathbf{xx}}_{n+1}(\mathbf{r}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}_{n+1}(\mathbf{r}_n^{n+1})\\ \boldsymbol{\mathcal{Q}}^{\mathbf{yx}}_{n+1}(\mathbf{r}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}_{n+1}(\mathbf{r}_n^{n+1})\end{bmatrix}}_{\boldsymbol{\mathcal{Q}}_{n+1}(\mathbf{r}_n^{n+1})}\right).$$

Apparently, the covariance of noises $\boldsymbol{\mathcal{Q}}_{n+1}(\mathbf{r}_n^{n+1}) = \boldsymbol{\mathcal{B}}_{n+1}(\mathbf{r}_n^{n+1})\boldsymbol{\mathcal{B}}^{\mathsf{T}}_{n+1}(\mathbf{r}_n^{n+1})$. $\mathbf{N}^{\mathbf{z}}_{n+1}(\mathbf{R}_n^{n+1})$ is the item links to the means $\mathbf{M}^{\mathbf{z}}_n$ and $\mathbf{M}^{\mathbf{z}}_{n+1}$:

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

Now we start to compute the filtering and smoothing in order. Under CGOMSM, we have $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{n+1}(\mathbf{R}_n^{n+1}) = 0$, and consequently $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is Markov with

$$p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) = p\left(r_{n+1} \left| r_n\right.\right) p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right), \tag{2.11}$$

and

$$p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right) = \mathcal{N}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yy}}_{n+1}(\mathbf{R}_n^{n+1})\mathbf{y}_n + \mathbf{N}^{\mathbf{y}}_{n+1}(\mathbf{R}_n^{n+1}), \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}_{n+1}(\mathbf{r}_n^{n+1})\right), \tag{2.12}$$

so that we get the joint probability

$$p\left(r_n, r_{n+1} \left| \mathbf{y}_1^{n+1}\right.\right) = \frac{p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) p\left(r_n \left| \mathbf{y}_1^n\right.\right)}{\sum\limits_{r_n, r_{n+1}} p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) p\left(r_n \left| \mathbf{y}_1^n\right.\right)}, \tag{2.13}$$

and thus

$$p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) = \frac{p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) p\left(r_n \left| \mathbf{y}_1^n\right.\right)}{\sum\limits_{r_n} p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) p\left(r_n \left| \mathbf{y}_1^n\right.\right)}. \tag{2.14}$$

According to (2.10), $(\mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ is Gaussian conditionally on $\mathbf{r}_n^{n+1}$ and $(\mathbf{x}_n, \mathbf{y}_n)$ with mean $\left(\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xx}}(\mathbf{r}_n^{n+1})\mathbf{x}_n + \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xy}}(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_{n+1}^{\mathbf{x}},\ \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{yy}}(\mathbf{r}_n^{n+1})\mathbf{y}_{n+1} + \mathbf{N}_{n+1}^{\mathbf{y}}\right)$ and variance-covariance matrix $\boldsymbol{\mathcal{Q}}_{n+1}(\mathbf{r}_n^{n+1})$. This implies that $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ is also Gaussian with mean

$$\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xx}}(\mathbf{r}_n^{n+1})\mathbf{x}_n + \boldsymbol{\mathcal{H}}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right), \tag{2.15}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right) = {} & \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xy}}(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_{n+1}^{\mathbf{x}}(\mathbf{r}_n^{n+1}) + \boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{xy}}(\mathbf{r}_n^{n+1}) \\ & \left(\boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{yy}}(\mathbf{r}_n^{n+1})\right)^{-1}\left(\mathbf{y}_{n+1} - \boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{yy}}(\mathbf{r}_n^{n+1})\mathbf{y}_n - \mathbf{N}_{n+1}^{\mathbf{y}}(\mathbf{r}_n^{n+1})\right)\end{aligned}, \tag{2.16}$$

and variance-covariance matrix

$$\mathbf{\Pi}_{n+1}^2\left(\mathbf{r}_n^{n+1}\right) = \boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{xx}}(\mathbf{r}_n^{n+1}) - \boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{xy}}(\mathbf{r}_n^{n+1})\left(\boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{yy}}(\mathbf{r}_n^{n+1})\right)^{-1}\boldsymbol{\mathcal{Q}}_{n+1}^{\mathbf{yx}}(\mathbf{r}_n^{n+1}). \tag{2.17}$$

Besides, as $(R_{n+1}, \mathbf{Y}_{n+1})$ and $\mathbf{X}_n$ are independent conditionally on $(R_n, \mathbf{Y}_n)$ in CGOMSM, we have $\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right.\right] = \mathbb{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n\right.\right]$. So the intermediate item of filtering is given according to (2.8), by the iterative computation of

$$\begin{aligned}\mathbb{E}\left[\mathbf{X}_{n+1} \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right] = & \sum_{r_n}\Big(p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) \\ & \left[\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{xx}}(\mathbf{r}_n^{n+1})\mathbb{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n\right.\right] + \boldsymbol{\mathcal{H}}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right)\right]\Big)\end{aligned}. \tag{2.18}$$

The covariance $\mathrm{Cov}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^{\intercal} \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right]$ can be achieved by computing the correlation in a similar way:

$$\begin{aligned}\mathbb{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^{\intercal}|r_{n+1},\mathbf{y}_1^{n+1}\right] = \sum_{r_n}\big(p\left(r_n\left|r_{n+1},\mathbf{y}_1^{n+1}\right.\right) \\ \big[\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{r}_n^{n+1})\mathbb{E}\left[\mathbf{X}_n\mathbf{X}_n^{\intercal}\left|r_n,\mathbf{y}_1^n\right.\right]\left(\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{r}_n^{n+1})\right)^{\intercal} + \\ \boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{r}_n^{n+1})\mathbb{E}\left[\mathbf{X}_n\left|r_n,\mathbf{y}_1^n\right.\right]\mathbf{H}_{n+1}^{\intercal}\left(\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right) + \\ \boldsymbol{\mathcal{H}}_{n+1}\left(\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right)\mathbb{E}\left[\mathbf{X}_n^{\intercal}\left|r_n,\mathbf{y}_1^n\right.\right]\left(\boldsymbol{\mathcal{F}}^{\mathbf{xx}}_{n+1}(\mathbf{r}_n^{n+1})\right)^{\intercal} + \\ \mathbf{\Pi}_{n+1}^2\left(\mathbf{r}_n^{n+1}\right) + \boldsymbol{\mathcal{H}}_{n+1}\left(\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right)\mathbf{H}_{n+1}^{\intercal}\left(\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right)\big]\big)\,.\end{aligned} \tag{2.19}$$

Finally, the filtering is calculated with $\mathbb{E}\left[\mathbf{X}_{n+1}\left|r_{n+1},\mathbf{y}_1^{n+1}\right.\right]$ and $p\left(r_n\left|\mathbf{y}_1^n\right.\right)$. The calculation of $p\left(r_n\left|\mathbf{y}_1^n\right.\right)$ is no more repeated here, since it is the filtering of discrete state-space PMC which has already been tackled in Chapter 1, Section 1.2.1.

Once we have the filtering, optimal smoothing (2.9) in CGOMSM is simple, as we have $\mathbb{E}\left[\mathbf{X}_n\left|r_n,\mathbf{y}_1^N\right.\right] = \mathbb{E}\left[\mathbf{X}_n\left|r_n,\mathbf{y}_1^n\right.\right]$ from

$$p\left(\mathbf{x}_n\left|r_n,\mathbf{y}_1^N\right.\right) = \frac{p\left(\mathbf{x}_n\left|r_n,\mathbf{y}_1^n\right.\right)p\left(\mathbf{y}_{n+1}^N\left|\mathbf{x}_n,r_n,\mathbf{y}_1^n\right.\right)}{p\left(\mathbf{y}_{n+1}^N\left|r_n,\mathbf{y}_1^n\right.\right)}, \tag{2.20}$$

and $p\left(\mathbf{y}_{n+1}^N\left|\mathbf{x}_n,r_n,\mathbf{y}_1^n\right.\right) = p\left(\mathbf{y}_{n+1}^N\left|r_n,\mathbf{y}_1^n\right.\right)$. Thus, smoothing can be calculated by (2.9). Meanwhile, the calculation of $p\left(r_n\left|\mathbf{y}_1^N\right.\right)$ is the smoothing of discrete state-space PMC, which has been also derived in Chapter 1, Section 1.2.1.

### 2.1.3 Parameterization of stationary models

Stationary models are more widely used for unsupervised data restoration than the time-varying ones. A CGPMSM is stationary if the distributions $p\left(\mathbf{t}_n^{n+1}\right) = p\left(\mathbf{x}_n^{n+1},\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right)$ of $\mathbf{T}_n^{n+1}$ do not depend on $n$, and thus are equal to $p\left(\mathbf{x}_1^2,\mathbf{r}_1^2,\mathbf{y}_1^2\right)$. Let us write the latter as

$$p\left(\mathbf{x}_1^2,\mathbf{r}_1^2,\mathbf{y}_1^2\right) = p\left(\mathbf{r}_1^2\right)p\left(\mathbf{x}_1^2,\mathbf{y}_1^2\left|\mathbf{r}_1^2\right.\right). \tag{2.21}$$

Besides, according to $p\left(r_2\left|\mathbf{x}_1, r_1, \mathbf{y}_1\right.\right) = p\left(r_2\left|r_1\right.\right)$ in (2.3), we have

$$\begin{aligned} p\left(\mathbf{x}_2, \mathbf{y}_2\left|\mathbf{r}_1^2\right.\right) &= p\left(\mathbf{x}_2, \mathbf{y}_2\left|r_2\right.\right), \\ p\left(\mathbf{x}_1, \mathbf{y}_1\left|\mathbf{r}_1^2\right.\right) &= p\left(\mathbf{x}_1, \mathbf{y}_1\left|r_1\right.\right). \end{aligned} \tag{2.22}$$

Finally, a stationary CGPMSM distribution is defined by $p\left(\mathbf{r}_1^2\right)$ and Gaussian distributions $p\left(\mathbf{x}_1^2, \mathbf{y}_1^2\left|\mathbf{r}_1^2\right.\right)$ on $\mathbb{R}^{s+q}$ defined by $K$ mean vectors and $K^2$ variance-covariance matrices. Thus, for $1 \leq j,\ k \leq K$, the model parameters are

$$p_{j,k} = p\left(r_1 = j, r_2 = k\right); \tag{2.23}$$

$$\mathbf{M}_j^{\mathbf{z}} = \begin{bmatrix} \mathbb{E}\left[\mathbf{X}_1 | r_1 = j\right] \\ \mathbb{E}\left[\mathbf{Y}_1 | r_1 = j\right] \end{bmatrix} = \begin{bmatrix} \mathbf{M}_j^{\mathbf{x}} \\ \mathbf{M}_j^{\mathbf{y}} \end{bmatrix}; \tag{2.24}$$

$$\begin{aligned} \mathbf{\Gamma}_{j,k}^{\mathbf{z}_1^2} &= \mathbb{E}\left[ \begin{bmatrix} \mathbf{Z}_1 - \mathbf{M}_j^{\mathbf{z}} \\ \mathbf{Z}_2 - \mathbf{M}_k^{\mathbf{z}} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_1 - \mathbf{M}_j^{\mathbf{z}} \\ \mathbf{Z}_2 - \mathbf{M}_k^{\mathbf{z}} \end{bmatrix}^{\intercal} \middle| r_1 = j, r_2 = k \right] \\ &= \begin{bmatrix} \mathbf{\Gamma}_j^{\mathbf{z}} & \mathbf{\Sigma}_{j,k}^{\mathbf{z}} \\ \left(\mathbf{\Sigma}_{j,k}^{\mathbf{z}}\right)^{\intercal} & \mathbf{\Gamma}_k^{\mathbf{z}} \end{bmatrix}, \end{aligned} \tag{2.25}$$

in which

$$\mathbf{\Gamma}_j^{\mathbf{z}} = \begin{bmatrix} \mathbf{\Gamma}_j^{\mathbf{xx}} & \mathbf{\Gamma}_j^{\mathbf{xy}} \\ \left(\mathbf{\Gamma}_j^{\mathbf{xy}}\right)^{\intercal} & \mathbf{\Gamma}_j^{\mathbf{yy}} \end{bmatrix}; \quad \mathbf{\Sigma}_{j,k}^{\mathbf{z}} = \begin{bmatrix} \mathbf{\Sigma}_{j,k}^{\mathbf{xx}} & \mathbf{\Sigma}_{j,k}^{\mathbf{xy}} \\ \mathbf{\Sigma}_{j,k}^{\mathbf{yx}} & \mathbf{\Sigma}_{j,k}^{\mathbf{yy}} \end{bmatrix}. \tag{2.26}$$

There we have two parameterizations of CGPMSM. We will call first parametrization the following one:

1. The set $\mathbf{\Theta}_1$ of $K^2$ probabilities $(p_{j,k})_{j,k\in\mathbf{\Omega}}$ given by (2.23);

2. The set $\mathbf{\Theta}_2$ of $K$ mean vectors $(\mathbf{M}_j^{\mathbf{z}})_{j\in\mathbf{\Omega}}$ given by (2.24);

3. The set $\mathbf{\Theta}_3$ of $K^2$ variance–covariance matrices given by (2.25).

Sets $\mathbf{\Theta}_2$, $\mathbf{\Theta}_3$ define the Gaussian distributions $p\left(\mathbf{z}_1, \mathbf{z}_2\left|r_1 = j, r_2 = k\right.\right)$. We will denote this first parametrization as $\mathbf{\Theta}^1 = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3\}$.

In the second parametrization, we will keep the same $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$, while $\mathbf{\Theta}_4$ will respect the regimes of the CGPMSM (2.10), and replace $\mathbf{\Theta}_3$. More precisely, in stationary CGPMSM, transitions of the triplet Markov chain do not depend on $n$, so (2.10) can be rewritten as:

$$\underbrace{\begin{bmatrix} \mathbf{X}_{n+1} \\ \mathbf{Y}_{n+1} \end{bmatrix}}_{\mathbf{Z}_{n+1}} = \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{F}}^{\mathbf{xx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{xy}}(\mathbf{R}_n^{n+1}) \\ \boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{R}_n^{n+1}) & \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(\mathbf{R}_n^{n+1}) \end{bmatrix}}_{\boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})} \underbrace{\begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix}}_{\mathbf{Z}_n} + \underbrace{\begin{bmatrix} \boldsymbol{\omega}_{n+1}^{\mathbf{x}} \\ \boldsymbol{\omega}_{n+1}^{\mathbf{y}} \end{bmatrix}}_{\boldsymbol{\omega}_{n+1}} + \underbrace{\begin{bmatrix} \mathbf{N}^{\mathbf{x}}(\mathbf{R}_n^{n+1}) \\ \mathbf{N}^{\mathbf{y}}(\mathbf{R}_n^{n+1}) \end{bmatrix}}_{\mathbf{N}^{\mathbf{z}}(\mathbf{R}_n^{n+1})}, \quad (2.27)$$

in which

$$\boldsymbol{\omega}_{n+1} \sim \mathcal{N}\left(\mathbf{0}, \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{Q}}^{\mathbf{xx}}(\mathbf{r}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}(\mathbf{r}_n^{n+1}) \\ \boldsymbol{\mathcal{Q}}^{\mathbf{yx}}(\mathbf{r}_n^{n+1}) & \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}(\mathbf{r}_n^{n+1}) \end{bmatrix}}_{\boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1})}\right).$$

$\boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1}) = \boldsymbol{\mathcal{B}}(r_{n+1})\boldsymbol{\mathcal{B}}^\intercal(r_{n+1})$ and the item considering the means:

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

where $\mathbf{M}^{\mathbf{z}}(R_n = j) = \mathbf{M}_j^{\mathbf{z}}$ in (2.24). Setting $\boldsymbol{\mathcal{F}}_{j,k} = \boldsymbol{\mathcal{F}}(r_n = j, r_{n+1} = k)$ and $\boldsymbol{\mathcal{Q}}_{j,k} = \boldsymbol{\mathcal{Q}}(r_n = j, r_{n+1} = k)$, we can say that the model is also defined by the parameters $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$ above, and by

4) The set $\mathbf{\Theta}_4$ of $K^2$ matrices $(\boldsymbol{\mathcal{F}}_{j,k})_{j,k\in\mathbf{\Omega}}$ and of $K^2$ variance–covariance matrices of noise $(\boldsymbol{\mathcal{Q}}_{j,k})_{j,k\in\mathbf{\Omega}}$ given by

$$\boldsymbol{\mathcal{F}}_{j,k} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{xx}} & \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{xy}} \\ \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} & \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yy}} \end{bmatrix}, \quad \boldsymbol{\mathcal{Q}}_{j,k} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_{j,k}^{\mathbf{xx}} & \boldsymbol{\mathcal{Q}}_{j,k}^{\mathbf{xy}} \\ \boldsymbol{\mathcal{Q}}_{j,k}^{\mathbf{yx}} & \boldsymbol{\mathcal{Q}}_{j,k}^{\mathbf{yy}} \end{bmatrix}. \quad (2.28)$$

We will call second parametrization, the set $\mathbf{\Theta}^2 = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_4\}$.

Let us specify the links between $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$. Classically, $\mathbf{\Theta}_4$ can be obtained from $\mathbf{\Theta}_3$ with

$$\boldsymbol{\mathcal{F}}_{j,k} = \left(\mathbf{\Sigma}^{\mathbf{z}}_{j,k}\right)^{\intercal} \left(\mathbf{\Gamma}^{\mathbf{z}}_{j}\right)^{-1}; \quad \boldsymbol{\mathcal{Q}}_{j,k} = \mathbf{\Gamma}^{\mathbf{z}}_{k} - \boldsymbol{\mathcal{F}}_{j,k}\mathbf{\Sigma}^{\mathbf{z}}_{j,k}. \tag{2.29}$$

And, conversely, using Lyapunov equation [63], (2.29) implies that

$$\begin{aligned} \mathbf{\Gamma}^{\mathbf{z}}_{j} &= argvec\left[\left(\mathbf{I} - \boldsymbol{\mathcal{F}}_{j,j} \otimes \boldsymbol{\mathcal{F}}_{j,j}\right)^{-1} vec\left(\boldsymbol{\mathcal{Q}}_{j,j}\right)\right]; \\ \mathbf{\Sigma}^{\mathbf{z}}_{j,k} &= \left(\boldsymbol{\mathcal{F}}_{j,k}\mathbf{\Gamma}^{\mathbf{z}}_{j}\right)^{\intercal}, \end{aligned} \tag{2.30}$$

in which $argvec\left(\cdot\right)$ is the inverse function of the operator vector $vec\left(\cdot\right)$ that stacks the columns of a matrix and $\otimes$ represents the Kronecker product.

We display in Figures 2.1 and 2.2, the variable dependences of the general stationary CGPMSM and stationary CGLSSM respectively.

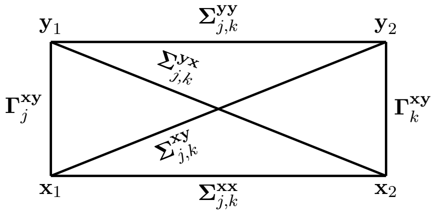

$\mathbf{y}_1$
$\mathbf{\Sigma}^{\mathbf{yy}}_{j,k}$
$\mathbf{y}_2$
$\mathbf{\Sigma}^{\mathbf{yx}}_{j,k}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{j}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{k}$
$\mathbf{\Sigma}^{\mathbf{xy}}_{j,k}$
$\mathbf{x}_1$
$\mathbf{\Sigma}^{\mathbf{xx}}_{j,k}$
$\mathbf{x}_2$

Figure 2.1: Dependence graph of CGPMSM.

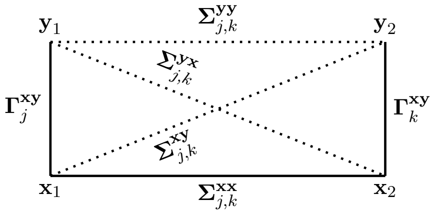

$\mathbf{y}_1$
$\mathbf{\Sigma}^{\mathbf{yy}}_{j,k}$
$\mathbf{y}_2$
$\mathbf{\Sigma}^{\mathbf{yx}}_{j,k}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{j}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{k}$
$\mathbf{\Sigma}^{\mathbf{xy}}_{j,k}$
$\mathbf{x}_1$
$\mathbf{\Sigma}^{\mathbf{xx}}_{j,k}$
$\mathbf{x}_2$

Figure 2.2: Dependence graph of CGLSSM.

#### 2.1.3.1 Reversible CGOMSM

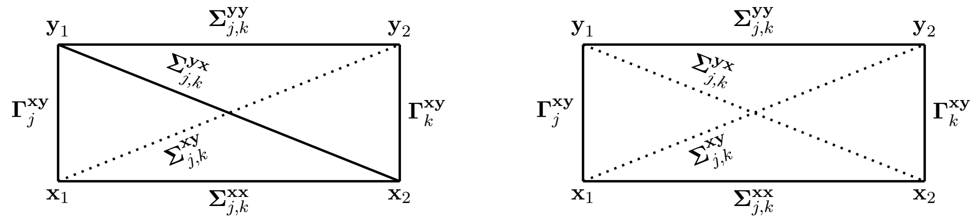
If a CGPMSM is concurrently a CGOMSM, then in $\mathbf{\Theta}_4$, the parameter $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}_{j,k} = \mathbf{0}$. From (2.29), we can get the relation of elements in $\mathbf{\Theta}_3$ of a CGOMSM that

$$\mathbf{\Sigma}^{\mathbf{xy}}_{j,k} = \mathbf{\Gamma}^{\mathbf{xy}}_{j} \left(\mathbf{\Gamma}^{\mathbf{yy}}_{j}\right)^{-1} \mathbf{\Sigma}^{\mathbf{yy}}_{j,k}, \tag{2.31}$$

see the variable dependence in Figure 2.3a. And if CGOMSM is reversible, then the optimal restoration can be conducted forwardly and backwardly. Symmetrically, the stationary reversible CGOMSM (CGOMSM-R) holds an extra condition that

$$\mathbf{\Sigma}^{\mathbf{yx}}_{j,k} = \mathbf{\Sigma}^{\mathbf{yy}}_{j,k} \left(\mathbf{\Gamma}^{\mathbf{yy}}_{k}\right)^{-1} \left(\mathbf{\Gamma}^{\mathbf{xy}}_{k}\right)^{\intercal}, \tag{2.32}$$

see the dependence in Figure 2.3b.


$\mathbf{y}_1$
$\mathbf{\Sigma}^{\mathbf{yy}}_{j,k}$
$\mathbf{y}_2$
$\mathbf{\Sigma}^{\mathbf{yx}}_{j,k}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{j}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{k}$
$\mathbf{\Sigma}^{\mathbf{xy}}_{j,k}$
$\mathbf{x}_1$
$\mathbf{\Sigma}^{\mathbf{xx}}_{j,k}$
$\mathbf{x}_2$
$\mathbf{y}_1$
$\mathbf{\Sigma}^{\mathbf{yy}}_{j,k}$
$\mathbf{y}_2$
$\mathbf{\Sigma}^{\mathbf{yx}}_{j,k}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{j}$
$\mathbf{\Gamma}^{\mathbf{xy}}_{k}$
$\mathbf{\Sigma}^{\mathbf{xy}}_{j,k}$
$\mathbf{x}_1$
$\mathbf{\Sigma}^{\mathbf{xx}}_{j,k}$
$\mathbf{x}_2$

(a) General CGOMSM.

(b) Reversible CGOMSM.

Figure 2.3: Dependence graph of CGOMSM.

The interesting point in the reversibility of CGOMSM-R is that exact backward filtering and smoothing can be calculated, which may be competitive comparing to the forward one when we apply this model to any data.

### 2.1.4 Restoration of simulated stationary data

We present two experiments here to verify the property of CGOMSM-R (Series 1) and show the efficiency of both the exact forward and backward restorations of CGOMSM when approximating the CGPMSM (Series 2). All results presented here are averages of 100 independent experiments. The abbreviations of methods used in the following experiments are

1. Opt-F: Optimal forward restoration knowing the true switches.
2. Opt-B: Optimal backward restoration knowing the true switches.
3. CGO-F: Exact forward CGOMSM restoration with unknown switches.
4. CGO-B: Exact backward CGOMSM restoration with unknown switches.

Each method includes filtering and smoothing.

***Series 1***

This experiment is conducted to test the equality of the forward and backward exact restorations of CGOMSM-R.

Assume a simple case that $\mathbf{X}_n$, $\mathbf{Y}_n$ are both scalar, $K = 2$, probabilities ($\mathbf{\Theta}_1$) $p_{1,1} = p_{2,2} = 0.45$, $p_{1,2} = p_{2,1} = 0.05$, and means ($\mathbf{\Theta}_2$) are all zero. The elements

Table 2.1: $\mathbf{\Theta}_3$ of series 1 (CGOMSM-R).

| $\mathbf{\Gamma}_{j,k}^{\mathbf{z}_1^2}$ | $k = 1$ | $k = 2$ |
|---|---|---|
| $j = 1$ | $\begin{bmatrix} 1.00 & 0.30 & 0.10 & 0.12 \\ 0.30 & 1.00 & 0.12 & 0.40 \\ 0.10 & 0.12 & 1.00 & 0.30 \\ 0.12 & 0.40 & 0.30 & 1.00 \end{bmatrix}$ | $\begin{bmatrix} 1.00 & 0.30 & 0.50 & 0.27 \\ 0.30 & 1.00 & 0.45 & 0.90 \\ 0.50 & 0.45 & 1.00 & 0.50 \\ 0.27 & 0.90 & 0.50 & 1.00 \end{bmatrix}$ |
| $j = 2$ | $\begin{bmatrix} 1.00 & 0.50 & 0.10 & 0.20 \\ 0.50 & 1.00 & 0.12 & 0.40 \\ 0.10 & 0.12 & 1.00 & 0.30 \\ 0.20 & 0.40 & 0.30 & 1.00 \end{bmatrix}$ | $\begin{bmatrix} 1.00 & 0.50 & 0.50 & 0.45 \\ 0.50 & 1.00 & 0.45 & 0.90 \\ 0.50 & 0.45 & 1.00 & 0.50 \\ 0.45 & 0.90 & 0.50 & 1.00 \end{bmatrix}$ |

Table 2.2: $\mathbf{\Theta}_4$ of series 1 (CGOMSM-R).

| $(j, k)$ | $(1, 1)$ | $(1, 2)$ | $(2, 1)$ | $(2, 2)$ |
|---|---|---|---|---|
| $\boldsymbol{\mathcal{F}}_{j,k}$ | $\begin{bmatrix} 0.07 & 0.10 \\ 0.00 & 0.40 \end{bmatrix}$ | $\begin{bmatrix} 0.40 & 0.33 \\ 0.00 & 0.90 \end{bmatrix}$ | $\begin{bmatrix} 0.05 & 0.09 \\ 0.00 & 0.40 \end{bmatrix}$ | $\begin{bmatrix} 0.37 & 0.27 \\ 0.00 & 0.90 \end{bmatrix}$ |
| $\boldsymbol{\mathcal{F}}_{bj,k}$ | $\begin{bmatrix} 0.70 & 0.10 \\ 0.00 & 0.40 \end{bmatrix}$ | $\begin{bmatrix} 0.04 & 0.19 \\ 0.00 & 0.40 \end{bmatrix}$ | $\begin{bmatrix} 0.49 & 0.03 \\ 0.00 & 0.90 \end{bmatrix}$ | $\begin{bmatrix} 0.37 & 0.27 \\ 0.00 & 0.90 \end{bmatrix}$ |

Table 2.3: Restoration result in Series 1.

|  | Obser-vation | Filtering/Smoothing | Filtering |  | Smoothing |
|---|---|---|---|---|---|
|  |  | Opt-F/Opt-B | CGO-F | CGO-B | CGO-F/CGO-B |
| Error Ratio | / | 0 | 0.203 | 0.263 | 0.155 |
| MSE | 1.201 | 0.829 | 0.834 | 0.836 | 0.833 |

set in the covariance matrix ($\mathbf{\Theta}_3$) are: $\mathbf{\Gamma}_j^{\mathbf{xx}} = \mathbf{\Gamma}_j^{\mathbf{yy}} = 1$, $\mathbf{\Gamma}_1^{\mathbf{xy}} = 0.3$, $\mathbf{\Gamma}_2^{\mathbf{xy}} = 0.5$, $\mathbf{\Sigma}_{j,1}^{\mathbf{xx}} = 0.1$, $\mathbf{\Sigma}_{j,1}^{\mathbf{yy}} = 0.4$, $\mathbf{\Sigma}_{j,2}^{\mathbf{xx}} = 0.5$, $\mathbf{\Sigma}_{j,2}^{\mathbf{yy}} = 0.9$, $\forall j, k \in \mathbf{\Omega} = \{1, 2\}$. $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}}$ and $\mathbf{\Sigma}_{j,k}^{\mathbf{yx}}$ are given by (2.31) and (2.32) respectively.

Parameters in $\mathbf{\Theta}_3$ are reported in Table 2.1. We denote the parameters of reverse model by adding a subscript $b$ to the notation of parameters of forward model. For example, the corresponding $\boldsymbol{\mathcal{F}}_{j,k}$ in reverse model to the original one is $\boldsymbol{\mathcal{F}}_{bj,k}$. Parameter $\boldsymbol{\mathcal{F}}_{j,k}$ (CGO-F) and $\boldsymbol{\mathcal{F}}_{bj,k}$ (CGO-B) in $\mathbf{\Theta}_4$ in this Series are listed in Table 2.2. We see no matter forwardly or backwardly, CGOMSM-R are CGOMSM as both $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}}$ and $\boldsymbol{\mathcal{F}}_{b_{j,k}}^{\mathbf{yx}}$ are zero, but the other parameters can be different.

10000 samples are simulated according to the parameter setting of this CGOMSM-R and then restored according to the four filtering as well as the four smoothing algorithms. We evaluate the restoration performance by error ratio of estimated switches comparing to the true ones and the Mean Square Error (MSE) of the restored hidden states comparing to true states which computes

$$MSE = \sum_{n=1}^{N} \left(\hat{\mathbf{x}}_n - \mathbf{x}_n\right)^2 / N. \tag{2.33}$$

Results of 100 independent experiments are reported in Table 2.3. It verifies that the smoothing results calculated from forward and backward directions are exactly equal, as both of them use all the information from observation with $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N \right.\right)$. Filtering results can be different. The forward filtering relies on $p\left(\mathbf{x}_n \left| \mathbf{y}_1^n \right.\right)$, while backward filtering relies on $p\left(\mathbf{x}_n \left| \mathbf{y}_n^N \right.\right)$. Turn to the optimal restorations knowing the switches, under the special structure of CGOMSM, $\mathbf{X}_n$ and $\mathbf{Y}_{n+1}$ are independent conditionally on $\mathbf{Y}_n$, which means that $p\left(\mathbf{y}_{n+1}^N \left| \mathbf{x}_n, \mathbf{y}_1^n \right.\right) = p\left(\mathbf{y}_{n+1}^N \left| \mathbf{y}_1^n \right.\right)$, and it leads to $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N \right.\right) = p\left(\mathbf{x}_n \left| \mathbf{y}_1^n \right.\right)$. So the optimal filtering and smoothing perform equivalently in CGOMSM knowing $\mathbf{r}_1^N$. Moreover, the optimal smoothing of CGOMSM-R calculated forwardly and backwardly are equal, that is why all the four optimal restorations give the same results. Trajectories of one experiment in this Series is illustrated in Figure 2.4.

### *Series 2*

This Series focuses on CGOMSM-based approximation for CGPMSM, to study the performance of CGO-B and decide weather CGO-F or CGO-B is closer to a given CGPMSM.

In this Series, data is simulated from the general CGPMSM with the same parameters $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ set in Series 1, but with different $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$. We consider $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} = 0.2$ for CGOMSM, then $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}}$ is given from the relation of $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$ (2.30) by calculating

$$\mathbf{\Sigma}_{j,k}^{\mathbf{xy}} = \left( \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} \mathbf{\Gamma}_j^{\mathbf{xx}} \mathbf{\Gamma}_j^{\mathbf{yy}} - \boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} \left( \mathbf{\Gamma}_j^{\mathbf{xy}} \right)^2 + \mathbf{\Sigma}_{j,k}^{\mathbf{yy}} \mathbf{\Gamma}_j^{\mathbf{xy}} \right) / \mathbf{\Gamma}_j^{\mathbf{yy}}.$$

This equation is only valid when both $\mathbf{X}_n$ and $\mathbf{Y}_n$ are scalar, and therefore all

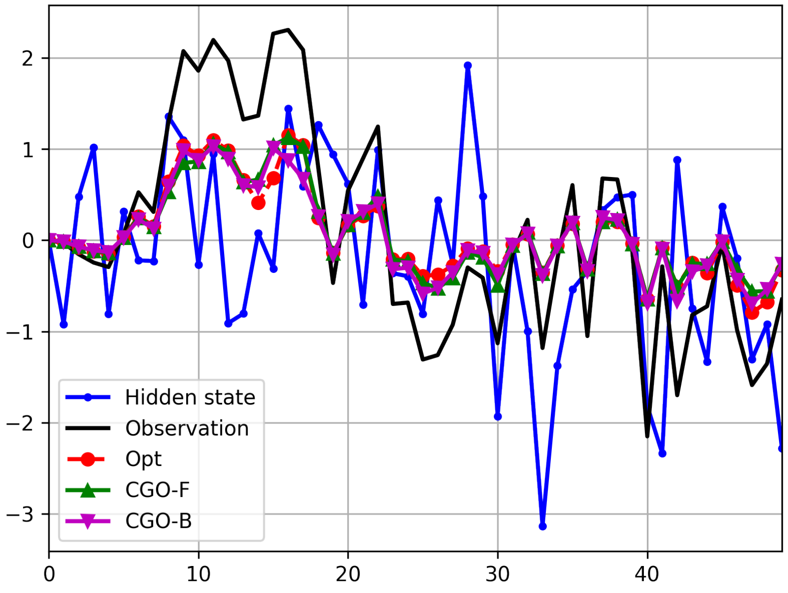


Figure 2.4: Trajectory example of Series 1 (50 samples).

parameters in $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$ are scalars. For vector case, it should be more complex but still computable. Moreover, we consider in this series, different difficult conditions for reverse CGOMSM to approximate the CGPMSM by changing $\boldsymbol{\mathcal{F}}_{b_{j,k}^{\mathbf{yx}}}$ from 0.0 to 0.3. The related parameter $\mathbf{\Sigma}_{j,k}^{\mathbf{yx}}$ in $\mathbf{\Theta}_4$ linked to this setting are calculated in a similar way of the calculation of $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}}$ by

$$\mathbf{\Sigma}_{j,k}^{\mathbf{yx}} = \left( \boldsymbol{\mathcal{F}}_{b_{j,k}^{\mathbf{yx}}} \mathbf{\Gamma}_k^{\mathbf{xx}} \mathbf{\Gamma}_k^{\mathbf{yy}} - \boldsymbol{\mathcal{F}}_{b_{j,k}^{\mathbf{yx}}} \left( \mathbf{\Gamma}_k^{\mathbf{xy}} \right)^2 + \mathbf{\Sigma}_{j,k}^{\mathbf{yy}} \mathbf{\Gamma}_k^{\mathbf{xy}} \right) / \mathbf{\Gamma}_k^{\mathbf{yy}}.$$

.

For all four CGPMSMs with different $\boldsymbol{\mathcal{F}}_{b_{j,k}^{\mathbf{yx}}}$, and each individual experiment, 10000 samples are simulated to test the exact restoration methods of approximated models. When using the CGOMSM to approximate the CGPMSM, the parameters used for CGO-F is modified from the CGPMSM ones. In detail, we replace $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}}$ by $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}} = \mathbf{\Gamma}_j^{\mathbf{xy}} \left( \mathbf{\Gamma}_j^{\mathbf{yy}} \right)^{-1} \mathbf{\Sigma}_{j,k}^{\mathbf{yy}}$ to get $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} = \mathbf{0}$ according to (2.31) for exact restoration of the approximated CGOMSM. Similarly, when using the reverse CGOMSM to approximate the CGPMSM, the original $\mathbf{\Sigma}_{j,k}^{\mathbf{yx}}$ is replaced according to (2.32) for the exact restoration of approximated CGO-B.

The error ratio of estimated switches comparing to the true ones are listed in Table 2.4. Under this parameter setting, the CGO-F performs better than CGO-B for filtering, but for smoothing, CGO-B surpasses CGO-F.

Table 2.4: Error ratio of estimated $\mathbf{R}_1^N$ in Series 2.

| $\mathcal{F}_{b_{j,k}}^{\mathbf{yx}}$ | Filtering | | Smoothing | |
|---|---|---|---|---|
| | CGO-F | CGO-B | CGO-F | CGO-B |
| 0.0 | 0.205 | 0.263 | 0.158 | 0.155 |
| 0.1 | 0.206 | 0.264 | 0.159 | 0.157 |
| 0.2 | 0.208 | 0.265 | 0.161 | 0.159 |
| 0.3 | 0.209 | 0.266 | 0.163 | 0.161 |

Table 2.5: MSE of estimated $\mathbf{X}_1^N$ in Series 2.

| $\mathcal{F}_{b_{j,k}}^{\mathbf{yx}}$ | Filtering | | | | Smoothing | | |
|---|---|---|---|---|---|---|---|
| | Opt-F | Opt-B | CGO-F | CGO-B | Opt-F/Opt-B | CGO-F | CGO-B |
| 0.0 | 0.829 | 0.743 | 0.839 | 0.762 | 0.743 | 0.837 | 0.758 |
| 0.1 | 0.807 | 0.742 | 0.818 | 0.761 | 0.726 | 0.817 | 0.757 |
| 0.2 | 0.743 | 0.741 | 0.765 | 0.761 | 0.676 | 0.765 | 0.756 |
| 0.3 | 0.633 | 0.740 | 0.680 | 0.761 | 0.587 | 0.679 | 0.756 |

Table 2.5 shows the MSE of estimated hidden states from all forward and backward methods. For optimal methods, knowing the switches, forward and backward smoothing are equal. Noticed that in this experiment, $\mathcal{F}_{j,k}^{\mathbf{yx}}$ were set to 0.2, when $\mathcal{F}_{b_{j,k}}^{\mathbf{yx}}$ is less than 0.2, the CGO-B has large opportunity to be a better approximation to CGPMSM, so that CGO-B gets better restoration than CGO-F. The performance superiority of CGO-B over CGO-F is more prominent when $\mathcal{F}_{b_{j,k}}^{\mathbf{yx}}$ gets closer to 0.0, and CGO-B becomes the exact restoration method when $\mathcal{F}_{b_{j,k}}^{\mathbf{yx}} = 0$. While $\mathcal{F}_{j,k}^{\mathbf{yx}} = \mathcal{F}_{b_{j,k}}^{\mathbf{yx}} = 0.2$, the forward and backward performance are difficult to compare from the complex dependence of the model. Nevertheless, it is reasonable to have CGO-B better than CGO-F, as referring to the optimal restoration, Opt-B is better than Opt-R. The smoothing of CGO-B has not too much improvements from filtering with better estimation of switches since only $p\left(r_n \left| \mathbf{y}_n^N \right.\right)$ is updated to $p\left(r_n \left| \mathbf{y}_1^N \right.\right)$ from filtering to smoothing, as in the same case of CGO-F for CGOMSM, see (2.20).

We may conclude that, both algorithms CGO-F and CGO-B are competitive when approximating a given CGPMSM. Normally, if we have all $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} > \boldsymbol{\mathcal{F}}_{b_{j,k}}^{\mathbf{yx}}$, it could be better to chose CGO-B to approximate the CGPMSM. But when it comes to the case (most of the time) that in one covariance set, for several classes of $(j, k)$, $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} > \boldsymbol{\mathcal{F}}_{b_{j,k}}^{\mathbf{yx}}$, while the other classes hold the contrary, it is hard to predict which of the two is the closer CGOMSM for a given CGPMSM. In our work presented by the following Sections, when mention CGOMSM, we will only consider CGOMSM forwardly, so as the corresponding CGO-F for restoration.

## 2.2 EM-based parameter estimation of stationary CGPMSM

So far, we consider only the restorations of CGPMSM assuming that all the parameters are known. From this Section, we are going to cope with the unsupervised restoration without knowledge of parameters. The primary problem we need to solve under unsupervised case, is the parameter estimation problem. We are going to deal the parameter estimation problem by using the classic EM principle, since under Gaussian linear case, the derivatives are computable in M-Step and EM shows more stability after converging comparing to ICE, see Figure 1.2a. However, when applying EM principle, the exact computation of $\mathbb{E}\left[\mathbf{X}_n \left|\mathbf{y}_1^N\right.\right]$ given by (2.9) is not possible under the general CGPMSM, which brings the out come that either $\mathbb{E}_{\mathbf{\Theta}^1}\left[\mathbf{Z}_1^N \left|\mathbf{y}_1^N\right.\right]$ or $\mathbb{E}_{\mathbf{\Theta}^2}\left[\mathbf{Z}_1^N \left|\mathbf{y}_1^N\right.\right]$ ($\mathbf{\Theta}^1$ and $\mathbf{\Theta}^2$ are the two equivalent parameter sets of CGPMSM defined in Section 2.1.3) in the E-step of the EM iterations can not be computed in a reasonable time.

The main contribution of this Section is to propose a general estimation method. Based on applying EM principle twice, this method allows one to estimate all model parameters[2] from all observation $\mathbf{Y}_1^N = \mathbf{y}_1^N$ only, with a certain number of possible switches $K$. The estimated parameters could be used for smoothing which results in unsupervised restoration.

Firstly, we notice that if $\mathbf{R}_1^N$ can be estimated, CGPMSM will degenerated

[2] $\mathbf{M}_j^{\mathbf{x}}$ is always assumed to be known since it can not be recovered.

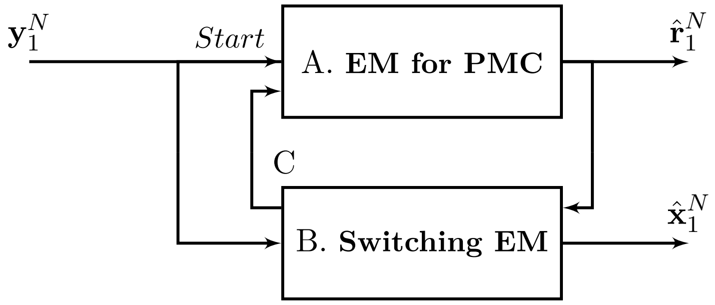


Figure 2.5: DEM-CGPMSM scheme.

to a GPMM with switching parameters, which allows EM to work. Secondly, for making the estimation of $\mathbf{R}_1^N$ possible, we approximate $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ to be a PMC. As consequence, the conditional probability $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ becomes computable through EM principle, and we can obtain the estimation of $\mathbf{R}_1^N$ with suitable criterion.

The algorithm we propose for parameter estimation of CGPMSM is constructed according to three successive steps as depicted in Fig. 2.5:

A. Assuming that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is a stationary PMC in CGPMSM, apply EM to estimate the parameters of $p\left(\mathbf{y}_1, \mathbf{y}_2 \left|r_1 = j, r_2 = k\right.\right)$ and $\hat{\boldsymbol{\theta}}_1 \in \boldsymbol{\Theta}_1$ from observation $\mathbf{y}_1^N$. Estimate $\mathbf{R}_1^N = \hat{\mathbf{r}}_1^N$ with Bayesian MPM method based on the estimated distribution $p\left(\mathbf{r}_1^N \left|\mathbf{y}_1^N\right.\right)$ and get $\hat{\mathbf{M}}_j^{\mathbf{y}} \in \boldsymbol{\Theta}_2$ with classical empirical estimations (see (2.34)).

B. Apply EM the second time[3] to estimate $\hat{\boldsymbol{\theta}}_4 \in \boldsymbol{\Theta}_4$ or equivalently $\hat{\boldsymbol{\theta}}_3 \in \boldsymbol{\Theta}_3$ from $\hat{\mathbf{r}}_1^N$ and $\hat{\boldsymbol{\theta}}_2 \in \boldsymbol{\Theta}_2$ obtained in step A above.

C. Go back to step A, and use the estimated distribution of $p\left(\mathbf{y}_1, \mathbf{y}_2 \left|r_1 = j, r_2 = k\right.\right)$ given by $\hat{\boldsymbol{\theta}}_2$ and $\hat{\boldsymbol{\theta}}_3$ to initialize the EM in step A.

The repeat of these three steps can be stopped with respect to some criterion. Let us remark that the first two steps above are sufficient to estimate all the parameters, however, the repeating of them by using result of step B as a new initialization of the EM in step A may improve the final result.

[3] This second EM algorithm is called Switching EM with details in following Section.

Let us detail the three steps which constitute the entire Double EM algorithm we propose.

Step A is for estimating $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$ and getting a realization $\hat{\mathbf{r}}_1^N$ of $\mathbf{R}_1^N$ for conducting the second EM later. By assuming $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is a PMC, the EM algorithm for estimating all parameters between $\mathbf{R}_1^N$ and $\mathbf{Y}_1^N$ and getting the realize $\mathbf{r}_1^N$ has actually already depicted in Section 1.2.2.1 of previous Chapter. Tracing back to 1.2.2.1, we get $\hat{\boldsymbol{\theta}}_1$ from the last M-step (1.19), and realization of $\hat{\mathbf{r}}_1^N$ by applying MPM criterion on $\phi_n(j)$ where $j \in \mathbf{\Omega}$ from the last E-step. $\mathbf{M}_j^{\mathbf{y}}$ in $\mathbf{\Theta}_2$ is then estimated from the observations classified by $\hat{\mathbf{r}}_1^N$ with

$$\hat{\mathbf{M}}_j^{\mathbf{y}} = \frac{\sum_{n=1}^N \mathbf{Y}_n \left(\hat{r}_n = j\right)}{\boldsymbol{Card}\left(j\right)}, \tag{2.34}$$

in which $\boldsymbol{Card}\left(j\right) = \sum_{n=1}^N \mathbb{1}\left(\hat{r}_n = j\right)$. We should notice that the parameters of $p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n, r_{n+1}\right.\right)$ estimated from EM principle in Step A are not going to be considered for the estimation of $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$, although they show parts of the covariance in $\mathbf{\Theta}_3$.

Step B is for estimating the remaining parameters $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$ by applying EM principle the second time, with hypothesis that $\hat{\boldsymbol{\theta}}_2$ and $\hat{\mathbf{r}}_1^N$ are the true ones. We detail the second EM algorithm in next Section.

### 2.2.1 EM estimation for CGPMSM with known switches

Knowing $\mathbf{\Theta}_2$ (the means) and the switches, a CGPMSM is actually a GPMM with switching parameters. In this Section, we extend the constant parameter GPMM-based EM algorithm [5] to the switching parameter case that we are dealing with here. We call this extension "Switching EM".

Under the assumption that $\mathbf{r}_1^N$ is known. For the convenience of likelihood expression, let $\mathbf{\Theta}^{\mathbf{z}}$ be the parameter set of the likelihood, which is actually constituted by the parameter sets defined in Section 2.1.3. The function to update $\mathbf{\Theta}^{\mathbf{z}}$ is

$$\mathbf{\Theta}^{\mathbf{z}(l+1)} = \arg\max_{\mathbf{\Theta}^{\mathbf{z}}} L(\mathbf{\Theta}^{\mathbf{z}(l)}, \mathbf{\Theta}^{\mathbf{z}}). \tag{2.35}$$

Index $l$ here denotes the Switching EM iteration, and

$$L(\boldsymbol{\Theta}^{\mathbf{z}(l)}, \boldsymbol{\Theta}^{\mathbf{z}}) = \mathbb{E}_{\boldsymbol{\Theta}^{\mathbf{z}(l)}} \left[\ln p_{\boldsymbol{\Theta}^{\mathbf{z}}} \left(\mathbf{z}_1^N\right) \left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right], \tag{2.36}$$

is the complete data likelihood. The joint distribution $p_{\boldsymbol{\Theta}^{\mathbf{z}}} \left(\mathbf{z}_1^N\right)$ can be factorized as

$$p_{\boldsymbol{\Theta}^{\mathbf{z}}} \left(\mathbf{z}_1^N\right) = p_{\mathbf{M}^{\mathbf{z}}_{r_1}, \boldsymbol{\Gamma}^{\mathbf{z}}_{r_1}} \left(\mathbf{z}_1\right) \prod_{n=1}^{N-1} p_{\boldsymbol{\Theta}_4} \left(\mathbf{z}_{n+1} | \mathbf{z}_n\right), \tag{2.37}$$

with

$$\begin{aligned} p_{\mathbf{M}^{\mathbf{z}}_{r_1}, \boldsymbol{\Gamma}^{\mathbf{z}}_{r_1}} \left(\mathbf{z}_1\right) &= \mathcal{N}\left(\mathbf{M}^{\mathbf{z}}_{r_1}, \boldsymbol{\Gamma}^{\mathbf{z}}_{r_1}\right); \\ p_{\boldsymbol{\Theta}_4} \left(\mathbf{z}_{n+1} | \mathbf{z}_n\right) &= \mathcal{N}\left(\boldsymbol{\mathcal{F}}(\mathbf{r}_n^{n+1}) + \mathbf{N}^{\mathbf{z}}(\mathbf{r}_n^{n+1}), \boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1})\right). \end{aligned} \tag{2.38}$$

$\mathbf{M}^{\mathbf{z}}_{r_1} \in \boldsymbol{\Theta}_2$ is given, while $\boldsymbol{\Gamma}^{\mathbf{z}}_{r_1} \in \boldsymbol{\Theta}_3$, is linked to $\boldsymbol{\Theta}_4$ with equation (2.29) and (2.30). To avoid complex derivation when doing maximization, we remove $p_{\mathbf{M}^{\mathbf{z}}_{r_1}, \boldsymbol{\Gamma}^{\mathbf{z}}_{r_1}} \left(\mathbf{z}_1\right)$ from the complete data likelihood, since only one point of data makes nearly no influence on the update of $\boldsymbol{\Theta}_4$. So, in M-step we calculate

$$\boldsymbol{\Theta}_4^{(l+1)} = \arg\max_{\boldsymbol{\Theta}_4} L(\boldsymbol{\Theta}_4^{(l)}, \boldsymbol{\Theta}_4), \tag{2.39}$$

which maximizes the simplified likelihood:

$$\begin{aligned} L(\boldsymbol{\Theta}_4^{(l)}, \boldsymbol{\Theta}_4) &= \mathbb{E}_{\boldsymbol{\Theta}_4^{(l)}} \left[\ln \prod_{n=1}^{N-1} p_{\boldsymbol{\Theta}_4} \left(\mathbf{z}_{n+1} | \mathbf{z}_n\right) \left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right], \\ &= \mathbb{E}_{\boldsymbol{\Theta}_4^{(l)}} \left[\sum_{n=1}^{N-1} \ln p_{\boldsymbol{\Theta}_4} \left(\mathbf{z}_{n+1} | \mathbf{z}_n\right) \left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right]. \end{aligned} \tag{2.40}$$

***E-step:***

With assumption that $\mathbf{R}_1^N = \hat{\mathbf{r}}_1^N$, $\boldsymbol{\Theta}_2 = \hat{\boldsymbol{\theta}}_2$, and $\boldsymbol{\Theta}_4 = \boldsymbol{\Theta}_4^{(l)}$, the E-step calculates $p\left(\mathbf{x}_n \left|\mathbf{r}_1^N, \mathbf{y}_1^N\right.\right)$ of the switching GPMM model. $p\left(\mathbf{x}_n \left|\mathbf{r}_1^n, \mathbf{y}_1^n\right.\right)$ is needed during its calculation.

As no confusion will be introduced, let us remove the dependence notation related to $\mathbf{r}$ for simplification in the calculation of this E-step. Then the computation is just similar to the computation of filtering and smoothing for a stationary GPMM discussed in Section 1.3. So we can just follow the equations from (1.31) to (1.34) to get in order $p\left(\mathbf{x}_n \left|\mathbf{y}_1^n\right.\right)$, and then the target of E-step $p\left(\mathbf{x}_n \left|\mathbf{y}_1^N\right.\right)$.

The computation of initial $p\left(\mathbf{x}_1 \left|\mathbf{y}_1\right.\right)$ is trivial. $p\left(\mathbf{x}_n \left|\mathbf{y}_1^n\right.\right)$, $\forall n \in 1, \ldots, N$ are calculated in a forward direction from $p\left(\mathbf{x}_n \left|\mathbf{y}_1^n\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|n}, \boldsymbol{P}_{n|n}\right)$ to $p\left(\mathbf{x}_{n+1} \left|\mathbf{y}_1^{n+1}\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n+1|n+1}, \boldsymbol{P}_{n+1|n+1}\right)$ through several intermediate variables:

$$\begin{aligned} \hat{\mathbf{x}}_{n|n+1} &= \hat{\mathbf{x}}_{n|n} + \boldsymbol{K}_{n|n+1}\tilde{\mathbf{y}}_{n+1|n}; \\ \boldsymbol{P}_{n|n+1} &= \boldsymbol{P}_{n|n} - \boldsymbol{K}_{n|n+1}\boldsymbol{S}_{n|n+1}\left(\boldsymbol{K}_{n|n+1}\right)^{\intercal}, \end{aligned} \tag{2.41}$$

with

$$\begin{aligned} \boldsymbol{S}_{n|n+1} &= \boldsymbol{\mathcal{Q}}^{\mathbf{yy}} + \boldsymbol{\mathcal{F}}^{\mathbf{yx}}\boldsymbol{P}_{n|n}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yx}}\right)^{\intercal}; \\ \boldsymbol{K}_{n|n+1} &= \boldsymbol{P}_{n|n}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yx}}\right)^{\intercal}\left(\boldsymbol{S}_{n|n+1}\right)^{-1}; \\ \hat{\mathbf{y}}_{n+1|n} &= \boldsymbol{\mathcal{F}}^{\mathbf{yx}}\hat{\mathbf{x}}_{n|n} + \boldsymbol{\mathcal{F}}^{\mathbf{yy}}\mathbf{y}_n + \mathbf{N}^{\mathbf{y}}; \\ \tilde{\mathbf{y}}_{n+1|n} &= \mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1|n}. \end{aligned} \tag{2.42}$$

Thus, we get

$$\begin{aligned} \hat{\mathbf{x}}_{n+1|n+1} &= \boldsymbol{\mathcal{A}}_n\hat{\mathbf{x}}_{n|n+1} + \boldsymbol{\mathcal{C}}_n; \\ \boldsymbol{P}_{n+1|n+1} &= \boldsymbol{\mathcal{Q}}_2 + \boldsymbol{\mathcal{A}}_n\boldsymbol{P}_{n|n+1}\left(\boldsymbol{\mathcal{A}}_n\right)^{\intercal}, \end{aligned} \tag{2.43}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{A}}_n &= \boldsymbol{\mathcal{F}}^{\mathbf{xx}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{F}}^{\mathbf{yx}}; \\ \boldsymbol{\mathcal{C}}_n &= \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\mathbf{y}_{n+1} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\mathbf{N}^{\mathbf{y}} \\ &\quad + \left(\boldsymbol{\mathcal{F}}^{\mathbf{xy}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{F}}^{\mathbf{yy}}\right)\mathbf{y}_n + \mathbf{N}^{\mathbf{x}}; \\ \boldsymbol{\mathcal{Q}}_2 &= \boldsymbol{\mathcal{Q}}^{\mathbf{xx}} - \boldsymbol{\mathcal{Q}}^{\mathbf{xy}}\left(\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)^{-1}\boldsymbol{\mathcal{Q}}^{\mathbf{yx}}. \end{aligned} \tag{2.44}$$

$p\left(\mathbf{x}_n \left|\mathbf{y}_1^N\right.\right)$, $\forall n \in 1, \ldots, N$ is calculated in a backward direction from $p\left(\mathbf{x}_{n+1} \left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n+1|N}, \boldsymbol{P}_{n+1|N}\right)$ to $p\left(\mathbf{x}_n \left|\mathbf{y}_1^N\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|N}, \boldsymbol{P}_{n|N}\right)$ according to

$$\begin{aligned} \hat{\mathbf{x}}_{n|N} &= \hat{\mathbf{x}}_{n|n+1} + \boldsymbol{K}_{n|N}(\hat{\mathbf{x}}_{n+1|N} - \hat{\mathbf{x}}_{n+1|n+1}); \\ \boldsymbol{P}_{n|N} &= \boldsymbol{P}_{n|n+1} + \boldsymbol{K}_{n|N}(\boldsymbol{P}_{n+1|N} - \boldsymbol{P}_{n+1|n+1})\left(\boldsymbol{K}_{n|N}\right)^{\intercal}, \end{aligned} \tag{2.45}$$

in which $\boldsymbol{K}_{n|N} = \boldsymbol{P}_{n|n+1} \left(\boldsymbol{\mathcal{A}}_n\right)^{\intercal} \left(\boldsymbol{P}_{n+1|n+1}\right)^{-1}$. For later use, we compute also the covariance between $\mathbf{x}_{n+1}$ and $\mathbf{x}_n$ knowing $\mathbf{y}_1^N$:

$$\boldsymbol{C}_{n+1,n|N} = \boldsymbol{P}_{n+1|N} \left(\boldsymbol{K}_{n|N}\right)^{\intercal}. \tag{2.46}$$

We should notice that this computation is of difference from the one in [5] and [101], 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 (2.27), and our model considers an extra mean item $\mathbf{N}_n^{\mathbf{y}}$.

***M-step:***

For the calculation in M-step, we need to take back the notation of $\mathbf{r}$, but the explicit dependence on the current iteration $l$ in (2.40) can be dropped, also, the dependence on $\mathbf{y}_1^N$ in the notation is removed for brevity. Then, the log-likelihood we need to maximize writes

$$L(\boldsymbol{\Theta}_4) = \sum_{n=1}^{N-1} L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right), \tag{2.47}$$

with

$$L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right) = \mathbb{E}\left[\ln p\left(\mathbf{z}_{n+1}\left|\mathbf{z}_n\right.\right)\right] = \mathbb{E}\left[\ln p\left(\mathbf{z}'_{n+1}\left|\mathbf{z}'_n\right.\right)\right], \tag{2.48}$$

in which

$$p\left(\mathbf{z}'_{n+1}\left|\mathbf{z}'_n\right.\right) = \mathcal{N}\left(\boldsymbol{\mathcal{F}}(\mathbf{r}_n^{n+1})\mathbf{z}'_n, \boldsymbol{\mathcal{Q}}(\mathbf{r}_n^{n+1})\right) \tag{2.49}$$

and

$$\begin{cases} \mathbf{z}'_{n+1} = \mathbf{z}_{n+1} - \mathbf{M}^{\mathbf{z}}(r_{n+1}), \\ \mathbf{z}'_n = \mathbf{z}_n - \mathbf{M}^{\mathbf{z}}(r_n). \end{cases} \tag{2.50}$$

We define covariances by

$$\begin{aligned} \boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n} &= \mathbb{E}\left[\mathbf{z}'_n\mathbf{z}'^t_n|\mathbf{y}_1^N\right] \\ &= \begin{bmatrix} \hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n) \end{bmatrix}^t + \begin{bmatrix} \boldsymbol{P}_{n|N} & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{2.51}$$

$$\begin{aligned}
\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n} &= \mathbb{E}\left[\mathbf{z}'_{n+1}\mathbf{z}'^{t}_{n}|\mathbf{y}_1^N\right] \\
&= \begin{bmatrix} \hat{\mathbf{x}}_{n+1|N} - \mathbf{M}^{\mathbf{x}}(r_{n+1}) \\ \mathbf{y}_{n+1} - \mathbf{M}^{\mathbf{y}}(r_{n+1}) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n) \\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n) \end{bmatrix}^t \\
&+ \begin{bmatrix} \boldsymbol{C}_{n+1,n|N} & 0 \\ 0 & 0 \end{bmatrix}.
\end{aligned} \tag{2.52}$$

Then, taking the derivative of the likelihood (2.47) with respect to each $\boldsymbol{\mathcal{F}}_{j,k}$, with $\forall j,k \in \boldsymbol{\Omega}$. We have

$$\begin{aligned}
\frac{\partial L(\boldsymbol{\Theta}_4)}{\partial \boldsymbol{\mathcal{F}}_{j,k}} &= \frac{\partial \sum_{n=1}^{N-1} L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right)}{\partial \boldsymbol{\mathcal{F}}_{j,k}} \\
&= \frac{\partial \sum_{n=1}^{N-1} \delta_n\left(j,k\right) L_n\left(\boldsymbol{\Theta}_4(r_n=j, r_{n+1}=k)\right)}{\partial \boldsymbol{\mathcal{F}}_{j,k}},
\end{aligned} \tag{2.53}$$

where $\delta_n\left(j,k\right)$ denotes the function $\mathbb{1}\left(r_n=j, r_{n+1}=k\right)$. Similarly, we take the derivative of the likelihood with respect to each $\boldsymbol{\mathcal{Q}}_{j,k}$

$$\begin{aligned}
\frac{\partial L(\boldsymbol{\Theta}_4)}{\partial \boldsymbol{\mathcal{F}}_{j,k}} &= \frac{\partial \sum_{n=1}^{N-1} L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right)}{\partial \boldsymbol{\mathcal{Q}}_{j,k}} \\
&= \frac{\partial \sum_{n=1}^{N-1} \delta_n\left(j,k\right) L_n\left(\boldsymbol{\Theta}_4(r_n=j, r_{n+1}=k)\right)}{\partial \boldsymbol{\mathcal{Q}}_{j,k}},
\end{aligned} \tag{2.54}$$

Making both (2.53) and (2.54) equals to zero, we can get the update expression of $\boldsymbol{\mathcal{F}}_{j,k}$. $\boldsymbol{\mathcal{Q}}_{j,k}$ in $\boldsymbol{\Theta}_4$ given by

$$\begin{aligned}
\hat{\boldsymbol{\mathcal{F}}}_{j,k} &= \widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_n,\mathbf{z}'_n}\right)^{-1}; \\
\hat{\boldsymbol{\mathcal{Q}}}_{j,k} &= \frac{1}{\boldsymbol{Card}\left(j,k\right)}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}} - \hat{\boldsymbol{\mathcal{F}}}_{j,k}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\intercal}\right),
\end{aligned} \tag{2.55}$$

where $\boldsymbol{Card}\,(j,k) = \sum_{n=1}^{N-1} \delta_n\,(j,k)$ and

$$\begin{aligned}
\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_n,\mathbf{z}'_n} &= \sum_{n=1}^{N-1} \delta_n\,(j,k)\,\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n};\\
\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n} &= \sum_{n=1}^{N-1} \delta_n\,(j,k)\,\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n};\\
\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}} &= \sum_{n=1}^{N-1} \delta_n\,(j,k)\,\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}}.
\end{aligned} \tag{2.56}$$

The details of all derivatives in M-step is available in Appendix A.

The log-likelihood $L\left(\boldsymbol{\Theta}^{\mathbf{z}};\mathbf{y}_1^N\right)$ is given by

$$\begin{aligned}
\mathbb{E}\left[\ln p\left(\mathbf{y}_1^N\,|\boldsymbol{\Theta}^{\mathbf{z}}\right)\right] = -\frac{1}{2}\sum_{n=1}^{N-1}\Big\{q\ln(2\pi) + \ln|\boldsymbol{S}_{n|n+1}|+\\
\left(\tilde{\mathbf{y}}_{n+1|n}\right)^{\intercal}\left(\boldsymbol{S}_{n|n+1}\right)^{-1}\left(\tilde{\mathbf{y}}_{n+1|n}\right)\Big\},
\end{aligned} \tag{2.57}$$

with $q$ denoting the dimension of $\mathbf{Y}_n$.

We provide here an experiment on simulated data to test the robustness of the proposed Switching EM under the assumption that true $\mathbf{r}_1^N$ is known, as Switching EM is an indispensable part of the entire Double EM algorithm for estimating the parameters $\boldsymbol{\Theta}_4$ and $\boldsymbol{\Theta}_3$ parallelly.

Consider a simple case of stationary CGPMSM, where $s = q = 1$, $\boldsymbol{\Omega} = \{1,2\}$, and the variance–covariance matrices is of the form:

$$\boldsymbol{\Gamma}_{j,k}^{\mathbf{z}_1^2} = \begin{bmatrix} 1 & b_j & a_{j,k} & d_{j,k} \\ b_j & 1 & e_{j,k} & c_{j,k} \\ a_{j,k} & e_{j,k} & 1 & b_k \\ d_{j,k} & c_{j,k} & b_k & 1 \end{bmatrix}. \tag{2.58}$$

with $\forall j,k \in \boldsymbol{\Omega}$. All variance are ones. The joint probabilities $\boldsymbol{\Theta}_1$ is set by $p_{1,1} = p_{2,2} = 0.45$, $p_{1,2} = p_{2,1} = 0.05$; all means in $\boldsymbol{\Theta}_2$ set to be zero and assumed known; In $\boldsymbol{\Theta}_3$: $b_1 = 0.3$, $b_2 = 0.5$, $a_{j,1} = 0.1$, $a_{j,2} = 0.5$, $c_{j,1} = 0.4$, $c_{j,2} = 0.9$, $e_{j,1} = 0.75$, $e_{j,2} = 0.33$, while $d_{j,1}$ and $d_{j,2}$ varies to make $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}}$ in $\boldsymbol{\Theta}_4$ which is the unique value defined later using the relationship between $\mathbf{\Theta}_3$ and $\mathbf{\Theta}_4$ in (2.29) varies between 0.05 and 0.40 (see Figure 2.6b)

$N = 10000$ samples of $\mathbf{T}_1^N = \left(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ are simulated with a ranging value of all $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} = \boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ from 0.05 to 0.40 to show the behavior of the Switching EM on CGPMSM with respect to the optimal smoothing for comparison. Larger $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ means that the CGPMSM specified here is less similar to a CGOMSM, which implies that the pair $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is less like a PMC. This remark will be of interest in unsupervised smoothing.

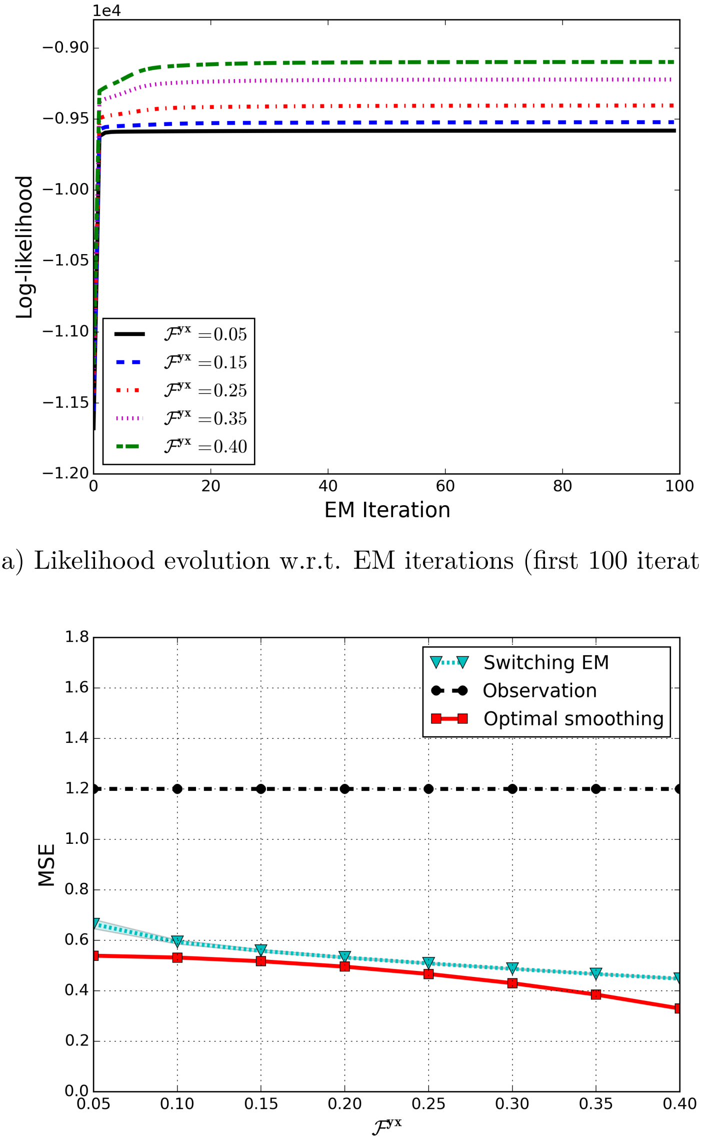
The initialization of $\mathbf{\Theta}_4$ [4] are always set to be the same through all experiments as

$$\boldsymbol{\mathcal{F}}_{j,1}^{(0)} = \begin{bmatrix} -0.5 & 1.0 \\ 0.2 & 0.5 \end{bmatrix}; \ \boldsymbol{\mathcal{F}}_{j,2}^{(0)} = \begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 0.5 \end{bmatrix};$$
$$\boldsymbol{\mathcal{Q}}_{j,k}^{(0)} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}.$$

$\mathcal{L} = 500$ iterations are set for Switching EM to converge, and all results are averages of 100 independent experiments. Figure 2.6a draws the likelihoods (first 100 EM iterations) of five different $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ values from Switching EM calculated by (2.57), they are all monotone increasing and convergent. Specifically in the case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.05$, the likelihood converged fastest as indicated in Fig. 2.6a, but the MSE hasn't been stable even at the last iteration in fact. This behavior shows that when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ becomes smaller (which means that the model gets closer to the CGOMSM) it is easier to get likelihood converged but get worse parameter estimation. The extreme case is obtained when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$, the likelihood can be maximized (converged) within one step, and only parameters $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yy}}$ and $\boldsymbol{\mathcal{Q}}_{j,k}^{\mathbf{yy}}$ can be estimated from the maximization. This point will be discussed later in Section 2.2.3.

The restoration result is illustrated in Figure 2.6b. Over all, it shows a good performance of Switching EM based restoration compared to the optimal result

[4] Initializations are set not too far from the true parameters, to make the local maximum approached from EM more possibly to be the global one.



(a) Likelihood evolution w.r.t. EM iterations (first 100 iterations).

(b) MSE of restoration.

Figure 2.6: Experiment of Switching EM (8 different values of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$).

knowing all the parameters. The restoration performs better and steadier with an increasing value of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ as indicated by the blue shadow which shows the 95% confidence interval of the result of Switching EM.

The true and estimated parameters under $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$ in this series are displayed as an example in Table 2.6 and Table 2.7

Table 2.6: True and estimated $\boldsymbol{\Theta}_4$ in experiment of Switching EM ($\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$).

| $\mathbf{R}_1^2$ | $(1,1)$ | $(1,2)$ | $(2,1)$ | $(2,2)$ |
|---|---|---|---|---|
| $\boldsymbol{\mathcal{F}}_{true}$ | $\begin{bmatrix} -0.137 & 0.791 \\ 0.400 & 0.280 \end{bmatrix}$ | $\begin{bmatrix} 0.440 & 0.202 \\ 0.400 & 0.780 \end{bmatrix}$ | $\begin{bmatrix} -0.367 & 0.933 \\ 0.400 & 0.200 \end{bmatrix}$ | $\begin{bmatrix} 0.444 & 0.111 \\ 0.400 & 0.700 \end{bmatrix}$ |
| $\boldsymbol{\mathcal{F}}^{(500)}_{SwitchingEM}$ | $\begin{bmatrix} -0.220 & 0.956 \\ 0.289 & 0.358 \end{bmatrix}$ | $\begin{bmatrix} 0.357 & 0.072 \\ 0.288 & 0.857 \end{bmatrix}$ | $\begin{bmatrix} -0.382 & 1.059 \\ 0.271 & 0.313 \end{bmatrix}$ | $\begin{bmatrix} 0.321 & 0.011 \\ 0.263 & 0.816 \end{bmatrix}$ |
| $\boldsymbol{\mathcal{Q}}_{True}$ | $\begin{bmatrix} 0.420 & 0.050 \\ 0.050 & 0.694 \end{bmatrix}$ | $\begin{bmatrix} 0.713 & 0.040 \\ 0.040 & 0.044 \end{bmatrix}$ | $\begin{bmatrix} 0.337 & 0.110 \\ 0.110 & 0.720 \end{bmatrix}$ | $\begin{bmatrix} 0.741 & 0.067 \\ 0.067 & 0.070 \end{bmatrix}$ |
| $\boldsymbol{\mathcal{Q}}^{(500)}_{SwitchingEM}$ | $\begin{bmatrix} 0.586 & -0.132 \\ -0.132 & 0.716 \end{bmatrix}$ | $\begin{bmatrix} 0.611 & 0.069 \\ 0.069 & 0.064 \end{bmatrix}$ | $\begin{bmatrix} 0.481 & -0.145 \\ -0.145 & 0.782 \end{bmatrix}$ | $\begin{bmatrix} 0.720 & 0.157 \\ 0.157 & 0.140 \end{bmatrix}$ |

Table 2.7: True and estimated $\boldsymbol{\Theta}_3$ in experiment of Switching EM ($\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$).

| $\boldsymbol{\Gamma}_j^{\mathbf{z}}$ | $\boldsymbol{\Gamma}_1^{\mathbf{z}}$ | $\boldsymbol{\Gamma}_2^{\mathbf{z}}$ |
|---|---|---|
| $True$ | $\begin{bmatrix} 1.000 & 0.300 \\ 0.300 & 1.000 \end{bmatrix}$ | $\begin{bmatrix} 1.000 & 0.500 \\ 0.500 & 1.000 \end{bmatrix}$ |
| $Switching\ EM^{(500)}$ | $\begin{bmatrix} 1.518 & 0.142 \\ 0.142 & 1.003 \end{bmatrix}$ | $\begin{bmatrix} 0.810 & 0.319 \\ 0.319 & 0.995 \end{bmatrix}$ |
| $Switching\ EM^{(0)}$ | $\begin{bmatrix} 1.411 & 0.251 \\ 0.251 & 0.809 \end{bmatrix}$ | $\begin{bmatrix} 0.696 & 0.146 \\ 0.146 & 0.743 \end{bmatrix}$ |

| $\boldsymbol{\Sigma}_{j,k}^{\mathbf{z}}$ | $\boldsymbol{\Sigma}_{1,1}^{\mathbf{z}}$ | $\boldsymbol{\Sigma}_{1,2}^{\mathbf{z}}$ | $\boldsymbol{\Sigma}_{2,1}^{\mathbf{z}}$ | $\boldsymbol{\Sigma}_{2,2}^{\mathbf{z}}$ |
|---|---|---|---|---|
| $True$ | $\begin{bmatrix} 0.100 & 0.484 \\ 0.750 & 0.400 \end{bmatrix}$ | $\begin{bmatrix} 0.500 & 0.634 \\ 0.333 & 0.900 \end{bmatrix}$ | $\begin{bmatrix} 0.100 & 0.500 \\ 0.750 & 0.400 \end{bmatrix}$ | $\begin{bmatrix} 0.500 & 0.750 \\ 0.333 & 0.900 \end{bmatrix}$ |
| $Switching\ EM^{(500)}$ | $\begin{bmatrix} -0.199 & 0.491 \\ 0.927 & 0.401 \end{bmatrix}$ | $\begin{bmatrix} 0.553 & 0.560 \\ 0.123 & 0.900 \end{bmatrix}$ | $\begin{bmatrix} 0.027 & 0.320 \\ 0.932 & 0.398 \end{bmatrix}$ | $\begin{bmatrix} 0.264 & 0.474 \\ 0.114 & 0.896 \end{bmatrix}$ |
| $Switching\ EM^{(0)}$ | $\begin{bmatrix} -0.455 & 0.408 \\ 0.683 & 0.455 \end{bmatrix}$ | $\begin{bmatrix} 0.730 & 0.408 \\ 0.206 & 0.455 \end{bmatrix}$ | $\begin{bmatrix} -0.202 & 0.212 \\ 0.670 & 0.401 \end{bmatrix}$ | $\begin{bmatrix} 0.363 & 0.212 \\ 0.147 & 0.401 \end{bmatrix}$ |

True parameters $\boldsymbol{\Theta}_4$ are reported in Table 2.6 in row "$\boldsymbol{\mathcal{F}}_{true}$ ", "$\boldsymbol{\mathcal{Q}}_{true}$" and the equivalent $\boldsymbol{\Theta}_3$ are in Table 2.7 of row "$True$", while rows "$\boldsymbol{\mathcal{F}}^{(500)}_{SwitchingEM}$", "$\boldsymbol{\mathcal{Q}}^{(500)}_{SwitchingEM}$" in Table 2.6, and "$SwitchingEM^{(500)}$" in Table 2.7 record the estimated $\boldsymbol{\Theta}_4$ and $\boldsymbol{\Theta}_3$ through Switching EM with 500 iterations. "$SwitchingEM^{(0)}$"

shows the initial parameters at the beginning of the Switching EM. Comparing to the initialization, the parameters estimated through Switching EM are closer to the true ones (more prominent in $\mathbf{\Theta}_4$).

### 2.2.2 Overall double-EM algorithm

So far, we have explained the Step A and Step B, and actually these two steps are already enough to estimate all the parameters. However, if we consider an improvement of initialization in Step A, the entire Double EM is then constructed by applying Step A, Step B sequentially and a feedback Step C, which can update the initialization of the parameters for Step A, so that we can iterate these three steps several loops to get better estimation.

In detail, what the feedback Step C does, is to return $\hat{\boldsymbol{\theta}}_1$, $\hat{\boldsymbol{\theta}}_2$ given by Step A, and the variance–covariance of $p\left(\mathbf{y}_1, \mathbf{y}_2 \left| r_1 = j, r_2 = k \right.\right)$ extracted from $\hat{\boldsymbol{\theta}}_3$ given by Step B, to be the initialization of the EM for discrete state-space PMC in next loop's Step A to replace the K-means initialization, which may cause failure.

The entire Double EM parameter estimation algorithm is summarized in Algorithm 1.

### 2.2.3 Discussion about special failure case of double-EM algorithm

The Double EM assumes that $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markovian to approach $\mathbf{r}_1^N$. When the model comes to be CGOMSM with $\mathbf{Y}_{n+1} = \boldsymbol{\mathcal{F}}^{\mathbf{yy}}(\mathbf{R}_n^{n+1})\mathbf{Y}_n$ (see (2.10)) and $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ truly Markovian, it seems it naturally possesses the assumption that we made for estimating the switches, but one should look out that, Switching EM becomes invalid when dealing with the parameter estimation of CGOMSM. Let us explain this point with details.

The general EM tries to find the maximum likelihood, where the model depends on unobserved latent variables, by alternating E-step and M-step through iterations. But under the case that $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}} = 0$, EM becomes invalid, since parameters defining $p\left(\mathbf{x}_1^N \left| \mathbf{y}_1^N \right.\right)$ have no influence on $p\left(\mathbf{y}_1^N\right)$. Here we give a simple proof on constant parameter GPMM that $\boldsymbol{\mathcal{F}}(\mathbf{R}_n^{n+1})$ is simplified to $\boldsymbol{\mathcal{F}}$, thus, $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}}$ is simplified to $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$

**Algorithm 1** Double EM

```
Inputs:
    $\mathbf{y}_1^N$, $K$.
Initialize:
    $\hat{\boldsymbol{\theta}}^{\mathbf{h}} = \left\{ \hat{p}_{j,k}, \hat{\mathbf{M}}^{\mathbf{y}_1^2}_{j,k}, \hat{\boldsymbol{\Gamma}}^{\mathbf{y}_1^2}_{j,k} \right\}$, $\hat{\boldsymbol{\theta}}_4 = \left\{ \hat{\boldsymbol{\mathcal{F}}}_{j,k}, \hat{\boldsymbol{\mathcal{Q}}}_{j,k} \right\}$.
Compute:
    for fb = 0 to $\mathcal{FB}$ do
        for i = 0 to $\mathcal{I}$ do
            1) EM for discrete state-space PMC.
            E-step: calculate $\phi_n(j)$, $\psi_n(j,k)$ in (1.7), (1.8) by (1.11)-(1.14).
            M-step: update $\hat{\boldsymbol{\theta}}_1 = \hat{p}_{j,k}$, $\hat{\mathbf{M}}^{\mathbf{y}_1^2}_{j,k}$ and $\hat{\boldsymbol{\Gamma}}^{\mathbf{y}_1^2}_{j,k}$ by (1.19).
        Estimate $\hat{\mathbf{M}}^{\mathbf{y}}_j$ of $\hat{\boldsymbol{\theta}}_2$ from (2.34), and $\hat{\mathbf{r}}_{n|N}$ from $\phi_n(j)$ with MPM criterion.
        for l = 0 to $\mathcal{L}$ do
            2) Switching EM.
            E-step: calculate $\hat{\mathbf{x}}_{n|N}$, $\boldsymbol{P}_{n|N}$, $\boldsymbol{C}_{n+1,n|N}$ with (2.41)-(2.46).
            M-step: get update of $\hat{\boldsymbol{\theta}}_4$ with (2.55).
        Calculate $\hat{\boldsymbol{\theta}}_3 = \left\{ \boldsymbol{\Gamma}^{\mathbf{z}_1^2}_{j,k} \right\}$ through (2.30) from estimated $\hat{\boldsymbol{\theta}}_4$,
        and extract $\boldsymbol{\Gamma}^{\mathbf{y}_1^2}_{j,k}$ as feedback to EM for discrete state-space PMC.
Outputs:
    $\hat{\boldsymbol{\theta}}_1$, $\hat{\boldsymbol{\theta}}_2$, $\hat{\boldsymbol{\theta}}_3$, $\hat{\boldsymbol{\theta}}_4$
```

and so as the other parameters as it is the same case in CGPMSM. The likelihood of observed data is:

$$p\left(\mathbf{y}_1^N \left|\boldsymbol{\Theta}^{\mathbf{z}}\right.\right) = \int_{\mathbf{x}_1^N} p\left(\mathbf{x}_1^N, \mathbf{y}_1^N \left|\boldsymbol{\Theta}^{\mathbf{z}}\right.\right) d\mathbf{x}_1^N, \tag{2.59}$$

as we consider the general case of GPMM, $\boldsymbol{\Theta}^{\mathbf{z}} = \{\boldsymbol{\Theta}_0, \boldsymbol{\Theta}_4\}$. $\boldsymbol{\Theta}_0$ represents the parameter of $p\left(\mathbf{x}_1, \mathbf{y}_1\right)$, and $\boldsymbol{\Theta}_4 = \{\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{Q}}\}$. While specially with $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$, we have $p\left(\mathbf{y}_{n+1} \left|\mathbf{y}_n\right.\right) = \mathcal{N}\left(\boldsymbol{\mathcal{F}}^{\mathbf{yy}}\mathbf{y}_n, \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right)$. The likelihood can be extended and simplified as:

$$\begin{aligned}
&\int_{\mathbf{x}_1^N} \left\{ p\left(\mathbf{x}_1, \mathbf{y}_1 \left|\boldsymbol{\Theta}_0\right.\right) \prod_{n=1}^{N} p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_n, \boldsymbol{\Theta}_4\right.\right) \right\} d\mathbf{x}_1^N \\
=& \int_{\mathbf{x}_1^N} \left\{ p\left(\mathbf{x}_1 \left|\mathbf{y}_1, \boldsymbol{\Theta}_0\right.\right) p\left(\mathbf{y}_1 \left|\boldsymbol{\Theta}_0\right.\right) \right. \\
&\left. \prod_{n=1}^{N} \left[ p\left(\mathbf{x}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_n, \boldsymbol{\Theta}_4\right.\right) p\left(\mathbf{y}_{n+1} \left|\mathbf{y}_n, \boldsymbol{\mathcal{F}}^{\mathbf{yy}}, \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right.\right) \right] \right\} d\mathbf{x}_1^N \\
=& p\left(\mathbf{y}_1 \left|\boldsymbol{\Theta}_0\right.\right) \prod_{n=1}^{N} p\left(\mathbf{y}_{n+1} \left|\mathbf{y}_n, \boldsymbol{\mathcal{F}}^{\mathbf{yy}}, \boldsymbol{\mathcal{Q}}^{\mathbf{yy}}\right.\right) \\
&\underbrace{\int_{\mathbf{x}_1^N} \left\{ p\left(\mathbf{x}_1 \left|\mathbf{y}_1, \boldsymbol{\Theta}_0\right.\right) \prod_{n=1}^{N} p\left(\mathbf{x}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_n, \boldsymbol{\Theta}_4\right.\right) \right\} d\mathbf{x}_1^N}_{=1}.
\end{aligned} \tag{2.60}$$

It means that when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$, we meet a special case where the parameters other than $\boldsymbol{\mathcal{F}}^{\mathbf{yy}}$, $\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}$ in $\boldsymbol{\mathcal{F}}$, $\boldsymbol{\mathcal{Q}}$ are not identifiable through maximum likelihood. But this extreme case can be rare since usually $\mathbf{Y}_1^N$ is a noised process of $\mathbf{X}_1^N$.

## 2.3 Unsupervised restoration in CGPMSM

This section aims to find a proper restoration approach for CGPMSM, so that we can get an unsupervised restoration method for the general CGPMSM by fusing the proposed Double EM algorithm for parameter estimation, and the restoration approach.

### 2.3.1 Two restoration approaches in CGPMSM

Once having the parameters of the switching model, one wants to restore the hidden states. As described before, optimal restoration is not feasible in CGPMSM, and a common way is to use MCMC methods to get its approximation. We derive the widely used particle filter for CGPMSM in Appendix B. To reduce the calculation burden of sampling method, here, we avoid the MCMC methods, and discuss two approaches based on the assumption that $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markovian. First switching model of this kind called "Conditionally Markov Switching Hidden Linear Models"(CMSHLMs) proposed in [115]. Subsequently, it has been shown that CGOMSMs which are not only particular CGPMSMs but also particular CMSHLMs, can be quite close to CGLSSMs [36], [112].

Tracing back to the E-step of Switching EM, we now reconsider the $\mathbf{r}_1^N$ in all conditional hidden state probabilities, as $\mathbf{r}_1^N$ is no more assumed to be known.

Let $\mathbf{T}_1^N = \left(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ be a CGPMSM, (2.2) implies:

$$\begin{aligned} p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^{n+1}\right.\right) &= \frac{p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^n\right.\right) p\left(r_{n+1} \left|\mathbf{x}_n, \mathbf{y}_1^n, \mathbf{r}_1^n\right.\right)}{p\left(r_{n+1} \left|\mathbf{y}_1^n, \mathbf{r}_1^n\right.\right)} \\ &= p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^n\right.\right). \end{aligned} \tag{2.61}$$

Then, one step forward calculation conditionally on $(\mathbf{r}_n^{n+1})$ is possible. As from

$$p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^n\right.\right) = p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^{n+1}\right.\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{n|n}(\mathbf{r}_1^n), \boldsymbol{P}_{n|n}(\mathbf{r}_1^n)\right), \tag{2.62}$$

we can calculate forwardly

$$\begin{aligned} p\left(\mathbf{x}_{n+1} \left|\mathbf{y}_1^{n+1}, \mathbf{r}_1^{n+1}\right.\right) &= \mathcal{N}\left(\hat{\mathbf{x}}_{n+1|n+1}(\mathbf{r}_1^{n+1}), \boldsymbol{P}_{n+1|n+1}(\mathbf{r}_1^{n+1})\right) \\ &= \mathcal{N}\left(\mathfrak{F}_{r_n^{n+1}}\left\{\hat{\mathbf{x}}_{n|n}(\mathbf{r}_1^n), \boldsymbol{P}_{n|n}(\mathbf{r}_1^n)\right\}\right). \end{aligned} \tag{2.63}$$

$\mathfrak{F}_{r_n^{n+1}}\{\cdot\}$ notes the current forward transform function for the mean and variance from (2.62) to (2.63), which is the same as from (2.41) to (2.43).

Also, as $p\left(\mathbf{x}_n \left|\mathbf{y}_1^N, \mathbf{r}_1^n\right.\right) = p\left(\mathbf{x}_n \left|\mathbf{y}_1^N, \mathbf{r}_1^{n+1}\right.\right)$, we have one step backward calculation similarly from

$$p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^N,\mathbf{r}_1^{n+1}\right.\right)=\mathcal{N}\left(\hat{\mathbf{x}}_{n+1|N}(\mathbf{r}_1^{n+1}),\boldsymbol{P}_{n+1|N}(\mathbf{r}_1^{n+1})\right),\tag{2.64}$$

we get

$$\begin{aligned}p\left(\mathbf{x}_n\left|\mathbf{y}_1^N,\mathbf{r}_1^n\right.\right)&=\mathcal{N}\left(\hat{\mathbf{x}}_{n|N}(\mathbf{r}_1^n),\boldsymbol{P}_{n|N}(\mathbf{r}_1^n)\right)\\&=\mathcal{N}\left(\mathfrak{B}_{r_n^{n+1}}\left\{\hat{\mathbf{x}}_{n+1|N}(\mathbf{r}_1^{n+1}),\boldsymbol{P}_{n+1|N}(\mathbf{r}_1^{n+1})\right\}\right).\end{aligned}\tag{2.65}$$

with $\mathfrak{B}_{r_n^{n+1}}\{\cdot\}$ denotes the current backward transformation function for the mean and variance from (2.64) to (2.65), which is the same as the calculation (2.45), paying attention that $(\hat{\mathbf{x}}_{n|n}(\mathbf{r}_1^n),\boldsymbol{P}_{n|n}(\mathbf{r}_1^n))$ are considered as known constants in $\mathfrak{B}_{r_n^{n+1}}\{\cdot\}$.

For filtering, notice that conditionally on $\mathbf{R}_n^{n+1}=\mathbf{r}_n^{n+1}$, $\mathbf{X}_{n+1}$ depend on $(\mathbf{X}_n,\mathbf{Y}_n^{n+1})$ linearly Gaussian, the linear forward transformation for mean and variance from $p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ to $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ is the same as from $p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_1^{n+1}\right.\right)$ to $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1},\mathbf{r}_1^{n+1}\right.\right)$.

In CGOMSM, $(\mathbf{R}_1^N,\mathbf{Y}_1^N)$ is Markov, then we have:

$$\begin{aligned}p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_n^{n+1}\right.\right)&=p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,r_n\right.\right),\\p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_1^{n+1}\right.\right)&=p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,\mathbf{r}_1^n\right.\right).\end{aligned}\tag{2.66}$$

This is the way of CGOMSM approximation, but if we need to use the original parameters of the model which is not with $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}=0$, then (2.66) may be too arbitrary. Here we adopt the transformation of mean and variance from $p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,r_n\right.\right)$ to $p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ the same as $p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,\mathbf{r}_1^n\right.\right)$ to $p\left(\mathbf{x}_n\left|\mathbf{y}_1^{n+1},\mathbf{r}_1^{n+1}\right.\right)$, which means that we adopt

$$\left\{\hat{\mathbf{x}}_{n+1|n+1}(\mathbf{r}_n^{n+1}),\boldsymbol{P}_{n+1|n+1}(\mathbf{r}_n^{n+1})\right\}=\mathfrak{F}_{r_n^{n+1}}\left\{\hat{\mathbf{x}}_{n|n}(r_n),\boldsymbol{P}_{n|n}(r_n)\right\}.\tag{2.67}$$

The mean and variance approach forwardly of $p\left(\mathbf{x}_{n+1}\left|\mathbf{y}_1^{n+1},r_{n+1}\right.\right)$ calculated from

$p\left(\mathbf{x}_n \left| \mathbf{y}_1^n, r_n\right.\right)$ is that

$$\left\{\hat{\mathbf{x}}_{n+1|n+1}(r_{n+1}), \boldsymbol{P}_{n+1|n+1}(r_{n+1})\right\} = \sum_{r_n} p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right) \mathfrak{F}_{r_n^{n+1}} \left\{\hat{\mathbf{x}}_{n|n}(r_n), \boldsymbol{P}_{n|n}(r_n)\right\}. \tag{2.68}$$

Although $p\left(\mathbf{x}_n \left| \mathbf{y}_1^n, r_n\right.\right)$ is actually non-Gaussian. So finally, the filtering is approximated by

$$\hat{\mathbf{x}}_{n+1|n+1} = \sum_{r_{n+1}} p\left(r_{n+1} \left| \mathbf{y}_1^{n+1}\right.\right) \hat{\mathbf{x}}_{n+1|n+1}(r_{n+1}). \tag{2.69}$$

For smoothing, if $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$, we easily have $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N, r_n\right.\right) = p\left(\mathbf{x}_n \left| \mathbf{y}_1^n, r_n\right.\right)$. But as $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ is none zero, we need to approximate $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N, r_n\right.\right)$. Similarly to (2.67), we approximate the backward transformation for mean and variance from $p\left(\mathbf{x}_{n+1} \left| \mathbf{y}_1^N, r_{n+1}\right.\right)$ to $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N, \mathbf{r}_n^{n+1}\right.\right)$ by the same transformation as from $p\left(\mathbf{x}_{n+1} \left| \mathbf{y}_1^N, \mathbf{r}_1^{n+1}\right.\right)$ to $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N, \mathbf{r}_1^{n+1}\right.\right)$, thus

$$\left\{\hat{\mathbf{x}}_{n|N}(\mathbf{r}_n^{n+1}), \boldsymbol{P}_{n|N}(\mathbf{r}_n^{n+1})\right\} = \mathfrak{B}_{r_n^{n+1}} \left\{\hat{\mathbf{x}}_{n+1|N}(r_{n+1}), \boldsymbol{P}_{n+1|N}(r_{n+1})\right\}, \tag{2.70}$$

but with constants $\hat{\mathbf{x}}_{n|n}(r_n), \boldsymbol{P}_{n|n}(r_n)$ calculated in filtering process in place of $\hat{\mathbf{x}}_{n|n}(\mathbf{r}_1^n), \boldsymbol{P}_{n|n}(\mathbf{r}_1^n)$ in $\mathfrak{B}_{r_n^{n+1}}\{\cdot\}$. then we have the mean and variance approach of $p\left(\mathbf{x}_n \left| \mathbf{y}_1^N, r_n\right.\right)$ calculated from $p\left(\mathbf{x}_n \left| \mathbf{y}_1^n, r_n\right.\right)$ and $p\left(\mathbf{x}_{n+1} \left| \mathbf{y}_1^N, r_{n+1}\right.\right)$ that

$$\left\{\hat{\mathbf{x}}_{n|N}(r_n), \boldsymbol{P}_{n|N}(r_n)\right\} = \sum_{r_{n+1}} p\left(r_{n+1} \left| r_n\right.\right) \mathfrak{B}_{r_n^{n+1}} \left\{\hat{\mathbf{x}}_{n+1|N}(r_{n+1}), \boldsymbol{P}_{n+1|N}(r_{n+1})\right\}. \tag{2.71}$$

Finally, the smoothing is approached by

$$\hat{\mathbf{x}}_{n|N} = \sum_{r_n} p\left(r_n \left| \mathbf{y}_1^N\right.\right) \hat{\mathbf{x}}_{n|N}(r_n). \tag{2.72}$$

The approximation proposed above is milder than assuming the model to be CGOMSM (it considers the information of $\mathbf{y}_{n+1}^N$ in $p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^N\right.\right)$ while CGOMSM holds that $p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^N\right.\right) = p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^n\right.\right)$.) and is equal to optimal one when the model is actually a CGOMSM. To conclude, we need to approximate three items still: $p\left(r_{n+1} \left| \mathbf{y}_1^{n+1}\right.\right)$, $p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1}\right.\right)$, $p\left(r_n \left| \mathbf{y}_1^N\right.\right)$. Here, we consider two ways to carry out the approximation. One way is based on parameter modification to CGOMSM with known transition probabilities. The other way is based on EM with unknown transition probabilities.

#### 2.3.1.1 Approximation based on parameter modification

We modify the parameters of the original CGPMSM to be CGOMSM, according to $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}\prime} = \mathbf{\Gamma}_j^{\mathbf{xy}} \left(\mathbf{\Gamma}_j^{\mathbf{yy}}\right)^{-1} \mathbf{\Sigma}_{j,k}^{\mathbf{yy}}$, then, $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markov chain and

$$p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right) = \mathcal{N}(\boldsymbol{\mathcal{F}}^{\mathbf{yx}\prime}(\mathbf{r}_n^{n+1}), \boldsymbol{\mathcal{Q}}^{\mathbf{yy}\prime}(\mathbf{r}_n^{n+1})), \tag{2.73}$$

where $\boldsymbol{\mathcal{F}}^{\mathbf{yx}\prime}(\mathbf{r}_n^{n+1})$ and $\boldsymbol{\mathcal{Q}}^{\mathbf{yy}\prime}(\mathbf{r}_n^{n+1})$ are calculated from the modified variance-covariance matrix with $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}}$ replaced by $\mathbf{\Sigma}_{j,k}^{\mathbf{xy}\prime}$. The three key probabilities of $p\left(\mathbf{r}_1^N \left| \mathbf{y}_1^N \right.\right)$ can be computed iteratively like under model discrete state-space PMC.

As $p\left(r_{n+1} \left| r_n, \mathbf{y}_n \right.\right) = p\left(r_{n+1} \left| r_n \right.\right)$, we have

$$p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) = p\left(r_{n+1} \left| r_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right). \tag{2.74}$$

Besides, since $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markov, we have

$$p\left(r_n, r_{n+1} \left| \mathbf{y}_1^{n+1} \right.\right) = \frac{p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) p\left(r_n \left| \mathbf{y}_1^n \right.\right)}{\sum_{r_n, r_{n+1}} p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) p\left(r_n \left| \mathbf{y}_1^n \right.\right)}, \tag{2.75}$$

and thus,

$$p\left(r_{n+1} \left| \mathbf{y}_1^{n+1} \right.\right) = \sum_{r_n} p\left(r_n, r_{n+1} \left| \mathbf{y}_1^{n+1} \right.\right); \tag{2.76}$$

$$p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right) = \frac{p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) p\left(r_n \left| \mathbf{y}_1^n \right.\right)}{\sum_{r_n} p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) p\left(r_n \left| \mathbf{y}_1^n \right.\right)}. \tag{2.77}$$

To iteratively calculate $p\left(r_n \left| \mathbf{y}_1^N \right.\right)$, $\beta(r_n) = \frac{p\left(\mathbf{y}_{n+1}^N \left| r_n, \mathbf{y}_n \right.\right)}{p\left(\mathbf{y}_{n+1}^N \left| \mathbf{y}_1^n \right.\right)}$ is introduced with $\beta(r_N) = 1$. Then

$$\beta(r_n) = \frac{\sum_{r_{n+1}} \beta(r_{n+1}) p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right)}{\sum_{r_n, r_{n+1}} p\left(r_n \left| \mathbf{y}_1^n \right.\right) p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right)}, \tag{2.78}$$

and

$$p\left(r_n \left| \mathbf{y}_1^N \right.\right) = p\left(r_n \left| \mathbf{y}_1^n \right.\right) \beta(r_n). \tag{2.79}$$

#### 2.3.1.2 Approximation based on EM

With EM, we can also estimated the three items under the same assumption that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is a PMC. Since the knowledge of transition probabilities of $\mathbf{R}_1^N$ as well as the parameters of $p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n, r_{n+1}\right.\right)$ is not required by EM, this approximation is more suitable for unsupervised case. It can be applied for smoothing after we get the parameters from Double EM, since modification of the estimated parameters to be CGOMSM can meet non-positive definite matrix problem (while a covariance matrix should be always positive semi-definite).

Still, we take $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ as a Markov chain. Different from the previous approximation proposed, we calculate $p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right)$ no more from $p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right)$ which follows the modified parameters as (2.73), but with equation

$$p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n\right.\right) = \frac{p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n, r_{n+1}\right.\right) p\left(r_n, r_{n+1}\right)}{\sum_{r_{n+1}} p\left(\mathbf{y}_n \left| r_n, r_{n+1}\right.\right) p\left(r_n, r_{n+1}\right)} \tag{2.80}$$

in which, $p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n = j, r_{n+1} = k\right.\right) = \mathcal{N}\left\{\hat{\mathbf{M}}_{j,k}, \hat{\mathbf{\Gamma}}_{j,k}\right\}$, $j, k \in \mathbf{\Omega}$ is estimated from the last M-step (1.19) of the EM algorithm.

This EM approach might be considered as a modification of $\mathbf{\Gamma}_j^{\mathbf{yy}}$ and $\mathbf{\Sigma}_{j,k}^{\mathbf{yy}}$ in the variance-covariance matrix, but the modified value is learned by EM. Just for a mention, as alternative methods of EM, SEM or ICE works also for approximation.

Here we discuss the performance of these two restoration approaches through an experiment considering supervised case. The model and parameters set in this experiment is the same as the former experiment we made in Section 2.2.1 for Switching EM.

All $\boldsymbol{\mathcal{F}}_{j,k}^{\mathbf{yx}}$ with $\forall j, k \in \mathbf{\Omega}$ are set to be equal represented by $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$, and $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ is varied from 0.00 to 0.40 to adjust the similarity of CGPMSM to CGOMSM. 100 iteration is set for EM to converge when doing the EM approach, and 200 particles is set for particle filter. For comparing the filtering performance of all methods, we simulate 200 samples from the model, while for smoothing we take 10000 samples, as the EM based approach requires enough amount of samples to find suitable $p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n, r_{n+1}\right.\right)$ of the assumed pairwise $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$. The performance of EM approach only applied on smoothing. Besides, the particle smoother is not considered here[5].

Shortly, we call the two approximation methods proposed above:

1) CGO-Appro: Restoration approach with partial approximation based on parameter modified to CGOMSM, using all parameters (as described in Section 2.3.1.1);

2) EM-Appro: Restoration approach with partial approximation based on EM without making use of the transition probability of switches (as described in Section 2.3.1.2).

To better understand the performance of these two approximations, we also do another approach:

3) Rough-Appro: Restoration approach with partial approximation like CGO-Appro, but with $\boldsymbol{\mathcal{F}}^{\mathbf{yx}\prime}(\mathbf{r}_n^{n+1})$ and $\boldsymbol{\mathcal{Q}}^{\mathbf{yy}\prime}(\mathbf{r}_n^{n+1})$ in (2.73) roughly replaced by the original $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}(\mathbf{r}_n^{n+1})$ and $\boldsymbol{\mathcal{Q}}^{\mathbf{yy}}(\mathbf{r}_n^{n+1})$.

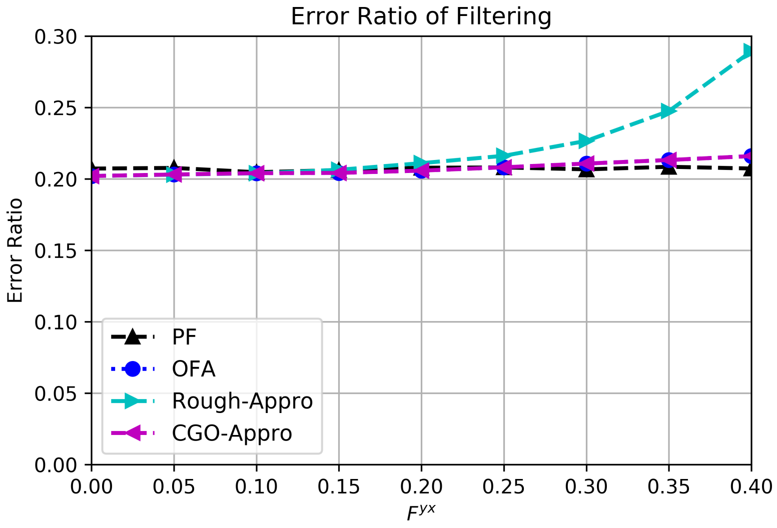
These three approaches are compared to several existing restoration methods listed bellow, which may appear also in the other experiments in later Sections.

- OF: Optimal filtering knowing true switches and true parameters;
- PF: Particle filter for CGPMSM;
- OFA: Optimal filtering approximation with unknown switches and true parameters modified to become a CGOMSM using equation (2.31) proposed in [1][6].

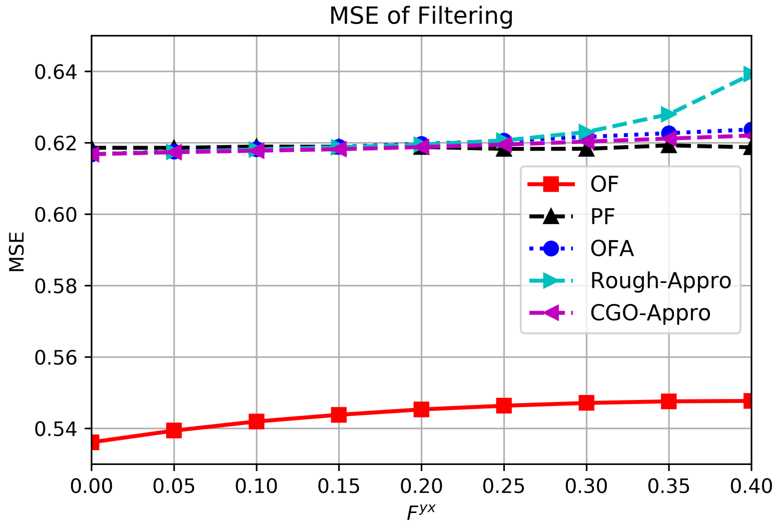
Correspondingly, the abbreviations of smoothing methods are represented by changing the “F” to “S”, *e.g.* “OS” represents Optimal smoothing; “OSA” represents optimal smoothing extended from “OFA”.

[5] we have not yet found a proper way to sample $p\left(\mathbf{r}_1^N \left| \mathbf{y}_1^N \right.\right)$ for smoothing, direct extension of the distribution $p\left(\mathbf{r}_1^N \left| \mathbf{y}_1^N \right.\right)$ estimated from particle filter suffers the sample depletion and gets very bad result.

[6] which actually has been used once under the name of CGO-F in Section 2.1.4.



(a) Error ratio of estimated switches.



(b) MSE of estimated hidden state.

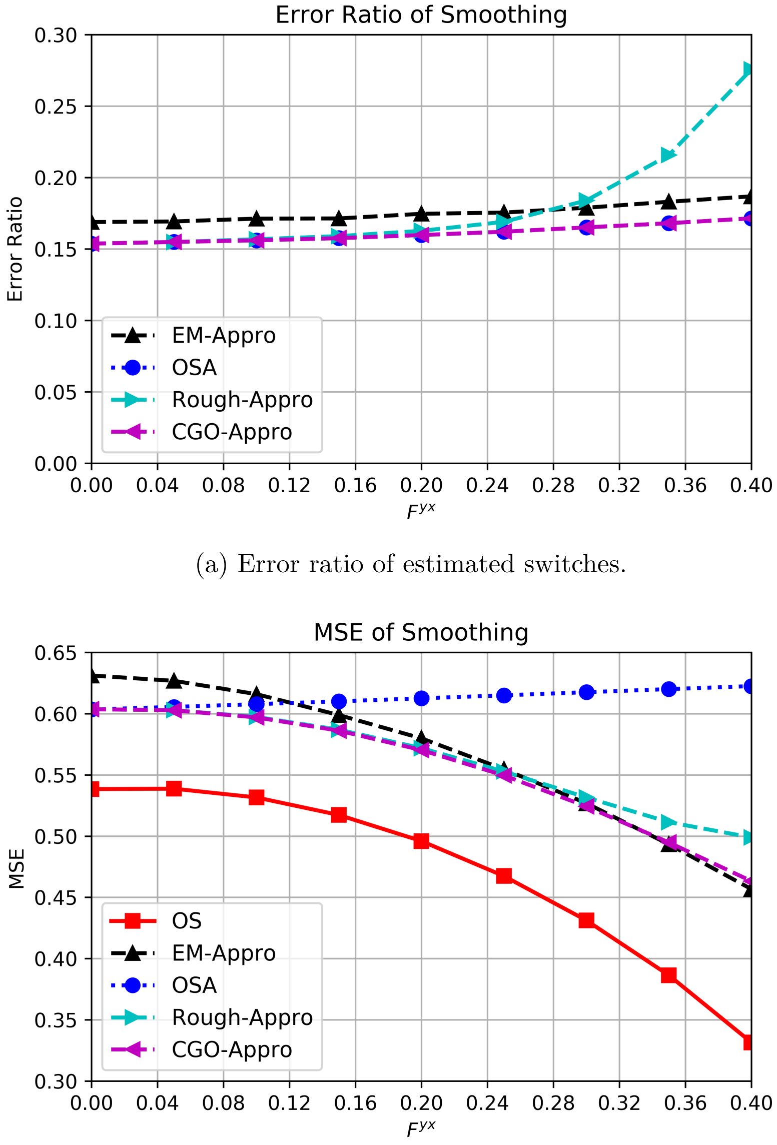
Figure 2.7: Experiment of CGPMSM filtering approaches (9 different values of $\mathcal{F}^{\mathbf{yx}}$).

Figure 2.7 shows the result of the methods listed above. The MSE of observation of the series is around 1.2.

See the error ratio of estimated $\mathbf{r}_1^N$ in Fig. 2.7a, OFA and CGO-Appro perform exactly the same, as their approximation process on $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ are the same. They get competitive error ratio as PF, and when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ is smaller (model is more like CGOMSM), CGO-Appro and OFA can perform better than PF as when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$ they are equal and both are optimal restoration. When data is far from model CGOMSM, PF works better as it has no approximation related to CGOMSM. Rough-Appro is affected a lot by the value of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$, when the model is going far from CGOMSM, it can not recover $\mathbf{R}_1^N$ appropriately.

The MSE of filtering result is displayed in Fig. 2.7b, the methods proposed with partial approximation and OFA perform nearly the same. Still, Rough-Appro gets worse result when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ becomes larger. PF can be considered as optimal filter when $\mathbf{r}_1^N$ is unknown once there are enough particles (under this experimental setting, we found empirically PF behaves asymptotically for 200 particles). Thus the proposed CGO-Appro is quite efficient as it performs quite close to PF but much less time consuming. Implemented with Python 3.6 on a 3.7GHz CPU, the CGO-Appro takes around 0.36 seconds while PF takes 36.20 seconds.

Turning to the smoothing result, the performance of different methods becomes more prominent. Figure 2.8a shows that with unknown transition matrix, EM-Appro performs little worse than OSA and CGO-Appro, and also the iteration of EM requires sufficient sample numbers. See the MSE of smoothing result in Fig. 2.8b, the approach methods proposed who still use the original parameters can maintain the same tendency as the OS, while OSA can not keep this tendency. The main reason is that, the assumption $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$ through out the restoration process of OSA means that $p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_1^N\right.\right)$ is equal to $p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_1^n\right.\right)$, thus, $\mathbf{y}_{n+1}^N$ only brings new information in $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ for smoothing comparing to filtering. This point has been explained in Section 2.1.4. The two approximation methods proposed here, although still partially based on the assumption that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is Markovian, adopt milder approximation for $p\left(\mathbf{x}_n \left|r_n, \mathbf{y}_1^N\right.\right)$, in which the information given by $\mathbf{y}_{n+1}^N$ is considered. CGO-Appro gets exactly the same result as OSA



(a) Error ratio of estimated switches.

(b) MSE of estimated hidden state.

Figure 2.8: Experiment of CGPMSM smoothing approaches (9 different values of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$).

when $\mathcal{F}^{\mathbf{yx}} = 0$, and then performs better when the model becomes less likely to a CGOMSM. Rough-Appro behaves the same, but not so well as CGO-Appro when $\mathcal{F}^{\mathbf{yx}}$ increases. Meanwhile, without relying on modification of the variance-covariance matrix by fixed value, EM-Appro has almost the same tendency as CGO-Appro when $\mathcal{F}^{\mathbf{yx}}$ increases, and in fact, no artificial modification of the elements in variance-covariance matrix can avoid the non-positive definite problem, which makes sense when doing smoothing. Regarding the time consumption of the two proposed method in this experiment, CGO-Appro takes around 26 seconds, while EM-Appro takes around 3 minutes because of the EM learning process. However, EM-Appro is still much less time-consuming comparing to particle methods.

We can conclude from this series of experiment that, normally, CGO-Appro works better than OSA, which shows that the partial approach we made in proposed method can be a milder one comparing to CGOMSM approach (when $\mathcal{F}^{\mathbf{yx}} = 0$ they are equal to each other). When it comes to smoothing case, with enough samples, EM-Appro can get also appropriate performance, especially when the model is far from CGOMSM.

### 2.3.2 Double EM based unsupervised restorations

Having both the strategies for parameter estimation and methods for approaching restoration, we can now study the way to accomplish the unsupervised restoration of the general CGPMSM. This Section of experiment aims to verify the performance of the unsupervised methods, which combines the Double EM with different restoration approaches, and also analyzes some impacts that can influent their performance.

The parameter estimation and unsupervised restoration methods, which will appear in the following experiments are listed bellow with their abbreviations. To avoid duplication, we omit the definitions of the methods which has been appeared in previous experiments, so as their abbreviations.

Two Double EM methods with different feedback times:

1. DEM ($\mathcal{FB} = 0$): Double EM without feedback for parameter estimation;

2. DEM ($\mathcal{FB} = 1$): Double EM with one feedback for parameter estimation.

One extra supervised restoration based on parameter modification:

- CGLSSM: Classical restoration with true $\mathbf{R}_1^N = \mathbf{r}_1^N$ and true parameters based CGLSSM obtained from CGPMSM[7].

Five unsupervised restoration methods based on Double EM and combined with different restoration approaches:

1. DEM-EM-Appro: DEM[8] combined with EM-Appro as entire unsupervised restoration.

2. DEM-CGO-Appro: DEM combined with CGO-Appro as entire unsupervised restoration.

3. DEM-R-MPM: Parameters estimated from DEM. The smoothing adopts the realization of $\hat{\mathbf{r}}_1^N$ using MPM criterion, and $\mathbb{E}\left[\mathbf{X}_n \left|\mathbf{y}_1^N\right.\right] = \mathbb{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^N\right.\right]$, assuming that $\hat{\mathbf{r}}_1^N$ is a proper estimation.

4. DEM-CGOMSM: Parameters estimated from DEM then modified into CGOMSM for smoothing.

5. DEM-CGLSSM: Parameters estimated from DEM then modified into CGLSSM, and take the realization $\hat{\mathbf{r}}_1^N$ from DEM for restoration.

As Double EM is based on all observation $\mathbf{y}_1^N$, the restoration methods we talk about here are all smoothing.

We present too series of experiments to better understand all these methods. The experiments are based on $N = 10000$ data simulated from specific model settings and for each setting, 100 independent experiments are conducted to provide average results. Iterations for the Double EM are set the same through the experiments as $\mathcal{I} = 100$ for EM in Step A and $\mathcal{L} = 500$ for Switching EM in Step B.

[7] Practically in this experiment, parameters obtained by modifying $d_{j,k}$, $e_{j,k}$, $c_{j,k}$ and setting them to $a_{j,k}b_k$, $a_{j,k}b_j$, $a_{j,k}b_jb_k$ respectively.

[8] While applying Double EM algorithm, no $\mathcal{FB}$ specified implies that one feedback is applied.

#### 2.3.2.1 Experiment on varying switching observation means

This series of experiments is designed for analyzing the performance of the Double EM based unsupervised restoration approach methods compared to several other supervised restoration approaches.

We change the focus of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ in the former experiments to the means of observation in $\mathbf{\Theta}_2$ (see the parameterization in Section 2.1.3), to adjust the difficulty of the circumstance for finding $\mathbf{\Theta}_1$ and estimating the switches. Data is generated with $|\mathbf{M}^{\mathbf{y}}|$ ranging from 0.0 to 2.5, where $|\mathbf{M}^{\mathbf{y}}|$ represents the absolute value of the mean of $\mathbf{Y}_n$, defined by the two possible values of $R_n$. As an example, $|\mathbf{M}^{\mathbf{y}}| = 2.5$ indicates that $\mathbf{M}_1^{\mathbf{y}} = 2.5$ and $\mathbf{M}_2^{\mathbf{y}} = -2.5$. Other parameter settings are the same as experiments of Swithing EM in Section 2.2.1, and initialization of the parameters for Double EM are also the same as (2.59).

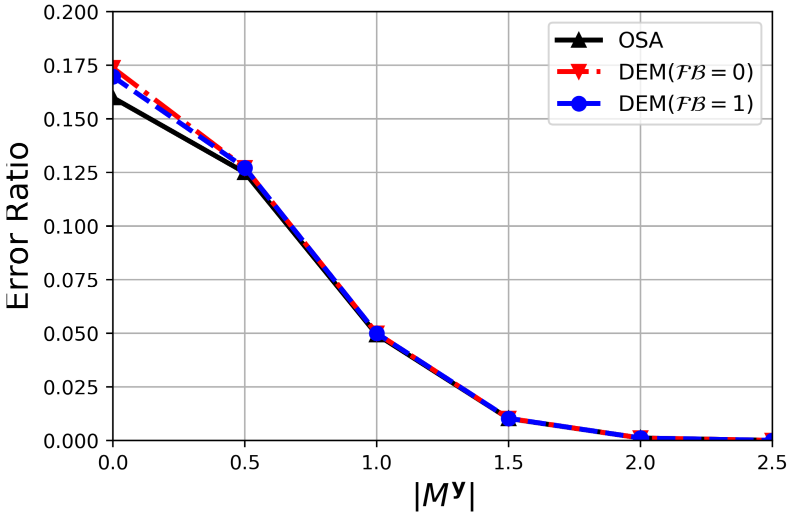
In this experiment, we chose only DEM-EM-Appro and DEM-R-MPM for unsupervised smoothing, to avoid the modification of estimated $\mathbf{\Theta}_3$ in CGOMSM or CGLSSM, which can meet non-positive definite matrix and any adjustment for turning it into positive definite one introduces more error.

Table 2.8: Estimated $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ in Series 2 ($\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$).

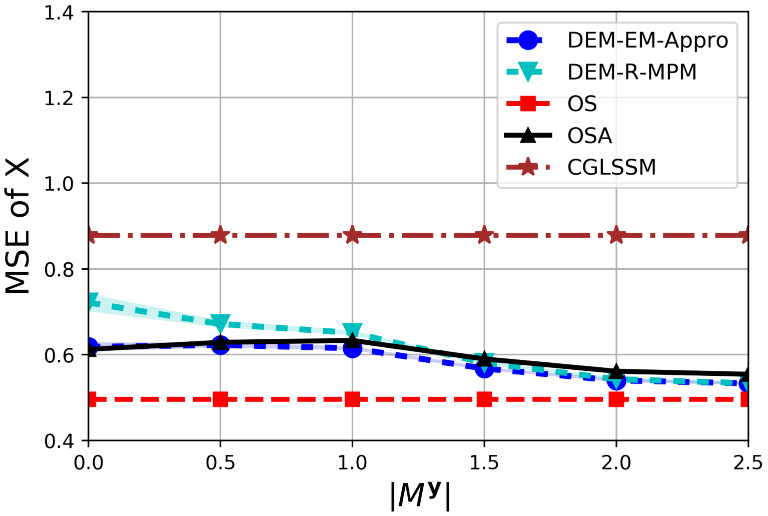
| $\lvert\mathbf{M}^y\rvert$ | | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
|---|---|---|---|---|---|---|---|
| $\hat{\boldsymbol{\theta}}_1$ | $p_{1,1}$ | 0.369 | 0.406 | 0.447 | 0.450 | 0.450 | 0.450 |
| | $p_{1,2} = p_{2,1}$ | 0.063 | 0.058 | 0.051 | 0.050 | 0.050 | 0.050 |
| | $p_{2,2}$ | 0.506 | 0.479 | 0.451 | 0.449 | 0.450 | 0.450 |
| $\hat{\boldsymbol{\theta}}_2$ | $\mathbf{M}_1^y$ | −0.007 | 0.540 | 1.007 | 1.503 | 2.000 | 2.499 |
| | $\mathbf{M}_2^y$ | −0.001 | −0.456 | −1.023 | −1.502 | −1.996 | −2.496 |

The restoration results under $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.20$ and $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$ are illustrated in Fig. 2.9 and Fig. 2.10 respectively. From these two figures, we observe that the performance of EM is very similar to OSA regarding the estimation of $\mathbf{R}_1^N$, even though OSA knows the transition probabilities of the switches. When $|\mathbf{M}^{\mathbf{y}}|$ reaches 2.5 we get $\mathbf{R}_1^N$ exactly estimated (0 error ratio).

Both of Figure 2.9a and Figure 2.10a verify that the feedback from the Step B to the initialization of Step A in Double EM can bring improvement of the error ratio when $|\mathbf{M}^{\mathbf{y}}|$ is small, which means a difficult situation for K-means to initialize

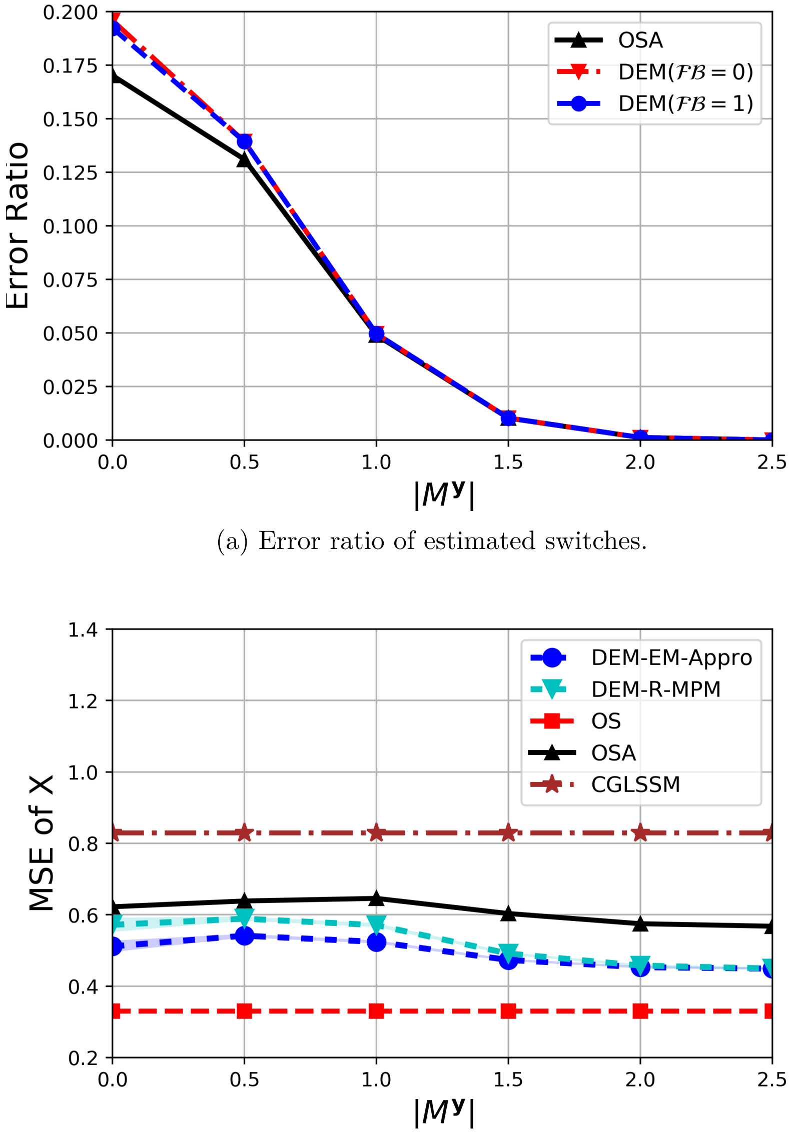


(a) Error ratio of estimated switches.



(b) MSE of estimated hidden state.

Figure 2.9: Result of restoration methods with varying $|\mathbf{M}^{\mathbf{y}}|$ ($\mathcal{F}^{\mathbf{yx}} = 0.20$).



(a) Error ratio of estimated switches.

(b) MSE of estimated hidden state.

Figure 2.10: Result of restoration methods with varying $|\mathbf{M}^{\mathbf{y}}|$ ($\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$).

the switches. And usually, one feedback is enough to find the proper initialization perceived from the experiments.

During the calculation of OSA, the intermediate probability $p\left(r_n \left| \mathbf{y}_1^N \right.\right)$ is computed. So, applying MPM, we get $\mathbf{R}_1^N$ estimated as its error ratio is drawn in the two Figures also. We can see that, $\hat{\mathbf{r}}_1^N$ given by Double EM get a reasonable worse error ratio than OSA, as OSA assumes that $\boldsymbol{\Theta}_1$ is known. Also, when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.20$, we get a better estimation of $\mathbf{R}_1^N$ compared to $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$. Noticed that the smaller $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ means the model is closer to CGOMSM, and consequently, the pair $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ generated from the CGPMSM is more similar to a PMC which contributes to this result. The estimated parameters $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ under $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.40$ are listed in Table 2.8. We do not list the estimated $\boldsymbol{\Theta}_4$ any more to save place, since the parameter estimation of Switching EM has been evaluated in Section 2.2.1.

Figure 2.9b and Figure 2.10b show the performance of all methods on the restoration of hidden states. With the increasing of $|\mathbf{M}^{\mathbf{y}}|$, $\mathbf{R}_1^N$ is better estimated, the observation data classified by the value of $\hat{\mathbf{r}}_1^N$ for the following Switching EM to estimate $\boldsymbol{\Theta}_4$ becomes more accurate, so the parameter estimation of Double EM becomes also more accurate. It is obvious that DEM-EM-Appro reaches better restoration than DEM-R-MPM, even though parameters are estimated from observation classified by $\hat{\mathbf{r}}_1^N$ realized with MPM (the light shadow of DEM-EM-Appro and DEM-R-MPM shows their 95% confidence interval). Only when $p\left(\mathbf{r}_1^N \left| \mathbf{y}_1^N \right.\right)$ refers to the MPM realization of $\hat{\mathbf{r}}_1^N$ by probabilities 1 or 0, DEM-EM-Appro and DEM-R-MPM are equal (case $|\mathbf{M}^{\mathbf{y}}| = 2.5$). In this series of experiment, all methods show much more efficient than CGLSSM. The performance of these two Double EM based methods is competitive to supervised OSA, and even has great chance to surpass it, which implies the advantage of keeping the parameters as CGPMSM when doing restoration for a general CGPMSM over the parameter modification approaches.

It is needed to be mentioned here that, the tendency of the restoration MSE through the two Double EM based methods displayed in both Figure 2.9b and Figure 2.10b are not monotonous decreasing, although with decreasing error ratio of estimated $\mathbf{R}_1^N$. This is caused by the error introduced when removing the mean of each individual wrong classification of $\mathbf{y}_1^N$. For example, if $\mathbf{y}_n$ is classified in a wrong class of $r_n$, when $|\mathbf{M}^{\mathbf{y}}|$ is larger, removing the mean cause larger error introduced when recovering $\mathbf{x}_n$. This tendency can be observed also in the result of any other method applied in the condition that $\mathbf{R}_1^N$ is unknown, the OSA in the figures for instance.

#### 2.3.2.2 Experiment on varying noise levels

To better describe the interest of the new methods we proposed in unsupervised smoothing, we continue comparing the efficiency of different methods increasing the "level of noise", which means decreasing the degree of stochastic dependence between the observed process $\mathbf{Y}_1^N$ and the hidden ones $(\mathbf{R}_1^N, \mathbf{X}_1^N)$.

Here, we take the same parameter for $\mathbf{\Theta}_1$ as the previous experimental series, set means in $\mathbf{\Theta}_2$ all zero. Then the noise level will be evolved through the parameters of the distribution $p\left(\mathbf{x}_1^N, \mathbf{y}_1^N \left| \mathbf{r}_1^N \right.\right)$ defined by $p\left(\mathbf{x}_1^2, \mathbf{y}_1^2 \left| \mathbf{r}_1^2 \right.\right)$. Indeed, the noise level is linked with covariances $b_j$, $d_{j,k}$, $e_{j,k}$ (see (2.58) and Fig. 2.1): the lower they are, the higher the noise level is. Thus, the MSE of smoothing based on true parameters will increase when the covariances $b_j$, $d_{j,k}$, $e_{j,k}$ decrease, and the interest in this series is to study whether unsupervised smoothing results are not too far from the real parameters based one. Of course, when these covariances are very small, the link between the observed signal and the hidden one is very tiny, thus the proposed parameter estimation method can not provide good results like any other methods. Let us mention that the covariances $a_{j,k}$, $c_{j,k}$ also play a role in the noise level. However, it is much more difficult to evaluate them theoretically.

We fix the value of $a_{j,k}$ and $c_{j,k}$ as: $a_{j,1} = 0.1$, $a_{j,2} = 0.5$, $c_{j,1} = 0.5$, $c_{j,2} = 0.9$ with $\forall j \in \{1, 2\}$. Consider two cases with $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$ and $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.3$, and five sub-cases with decreasing noise, which means that $b_j$, $e_{j,k}$, $d_{j,k}$ increase, whose parameters are given in Table 2.9. The initialization of $\mathbf{\Theta}_4$ is chosen also to be the same, except that the initial $\boldsymbol{\mathcal{F}}_{j,1}^{\mathbf{xx}}$ is changed to 0.1 for suiting this series.

As it has been proved in previous Series that DEM-EM-Appro performs better than DEM-R-MPM, in this series, instead of DEM-R-MPM, we consider the other three unsupervised restoration approaches with parameters estimated from Double

EM: DEM-CGO-Appro, DEM-CGOMSM and DEM-CGLSSM. When modification of parameters into CGOMSM and CGLSSM meets non-positive definite covariance matrix, we replace their negative eigenvalues with a small positive value. This adjustment assures the process but possibly drive parameters inappropriate. The mean MSE of observation is set as a threshold for the selection of proper instances in 100 experiments to show the average result.

Table 2.9: Parameters of 5 different noise cases.

| Sub-case | $b_1$ | $b_2$ | $e_{j,1}$ | $e_{j,2}$ | $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$ | | | | $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.3$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $d_{1,1}$ | $d_{1,2}$ | $d_{2,1}$ | $d_{2,2}$ | $d_{1,1}$ | $d_{1,2}$ | $d_{2,1}$ | $d_{2,2}$ |
| 1 | 0.00 | 0.20 | 0.40 | 0.10 | 0.10 | 0.10 | 0.20 | 0.18 | 0.30 | 0.30 | 0.39 | 0.47 |
| 2 | 0.10 | 0.30 | 0.50 | 0.20 | 0.15 | 0.19 | 0.24 | 0.36 | 0.35 | 0.39 | 0.42 | 0.54 |
| 3 | 0.20 | 0.40 | 0.60 | 0.30 | 0.20 | 0.28 | 0.28 | 0.44 | 0.39 | 0.47 | 0.45 | 0.61 |
| 4 | 0.30 | 0.50 | 0.70 | 0.40 | 0.24 | 0.36 | 0.33 | 0.53 | 0.42 | 0.54 | 0.48 | 0.68 |
| 5 | 0.40 | 0.60 | 0.80 | 0.50 | 0.28 | 0.44 | 0.36 | 0.60 | 0.45 | 0.61 | 0.49 | 0.73 |

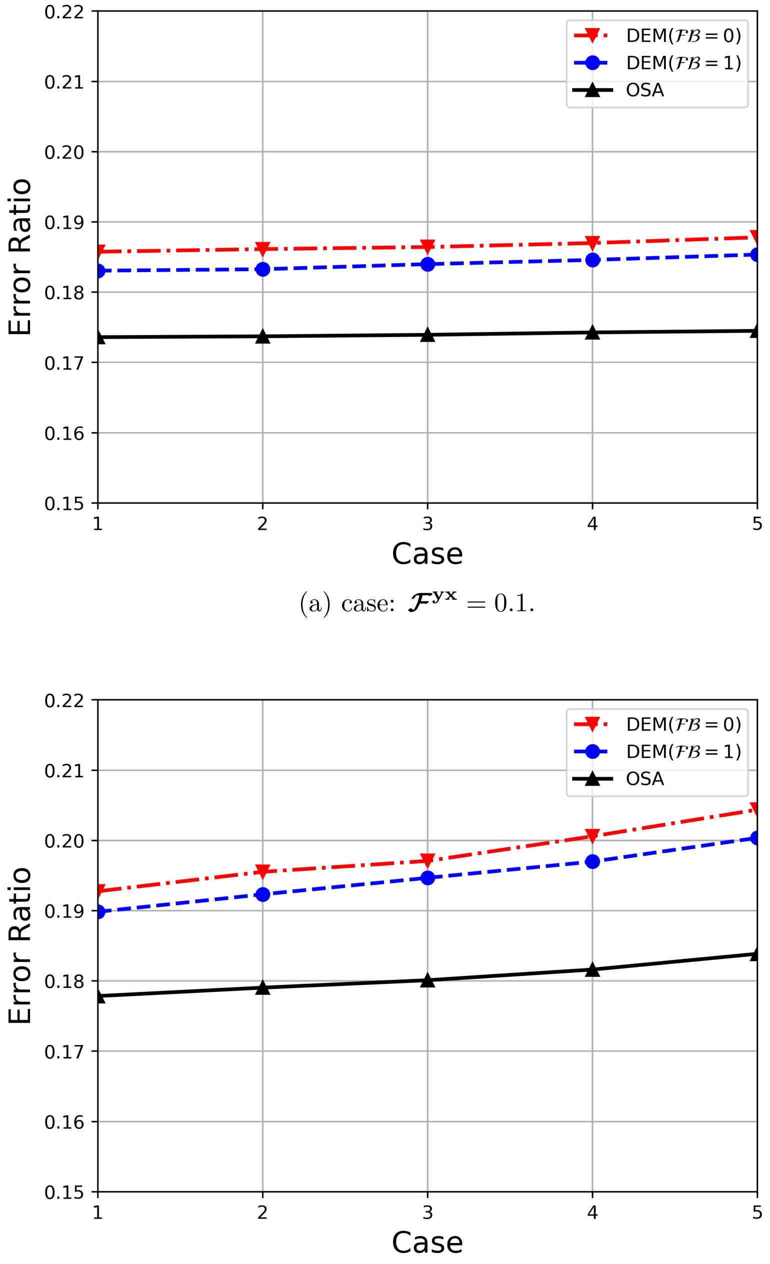
Results of this series is given in Figure 2.11 (shows the error ratio of restored switches) and Figure 2.12 (shows the restoration MSE of all methods considered).

Under supervised case, the approximated models CGOMSM of Case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$ is the same as of Case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.4$ so as the approximated model CGLSSM. The reason is that they both modify $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ to zero.

Let us see the estimation result of $\mathbf{R}_1^N$. Both of the Figures 2.11a and 2.11b present the improvement of the error ratio after one feedback in Double EM. Also, small increasing error ratio which can be observed with the increasing number of sub-cases in these two Figures indicates that, the more $\mathbf{Y}_n$ links to $\mathbf{X}_{n-1}$, $\mathbf{X}_n$, $\mathbf{X}_{n+1}$ (by increasing $b_{j,k}$, $e_{j,k}$, $d_{j,k}$, $\forall j, k \in \boldsymbol{\Omega}$) the less $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ can be considered as PMC. In fact, although not being proved in this Series, $c_{j,k}$ has also significant influence on the error ratio of estimated switches, as it defines the noise level between $\mathbf{Y}_n$ and $\mathbf{Y}_{n+1}$.

Comparing Figures 2.11a and 2.11b, for each sub-case, case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.3$ always gets more error in the restored switches than $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$, since $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is less PMC like with larger $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ value, so that the EM in Step A of Double EM is less efficient.

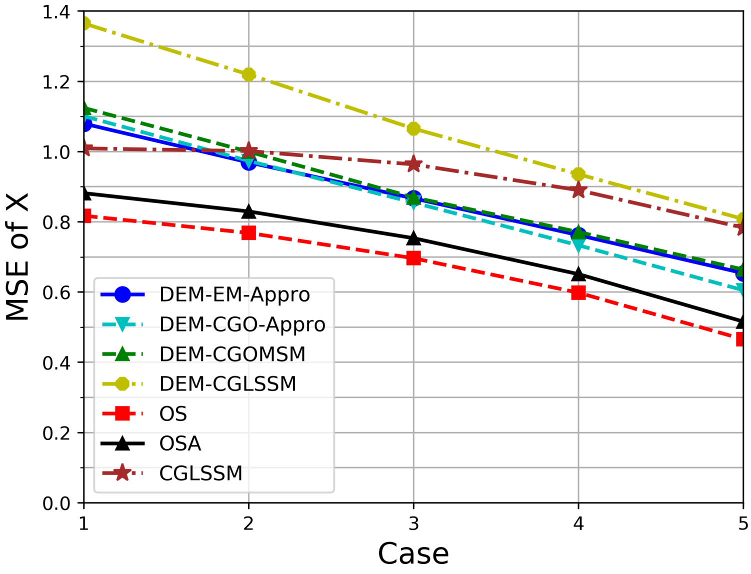
Combining the restoration MSE of the methods considered illustrated in Figure



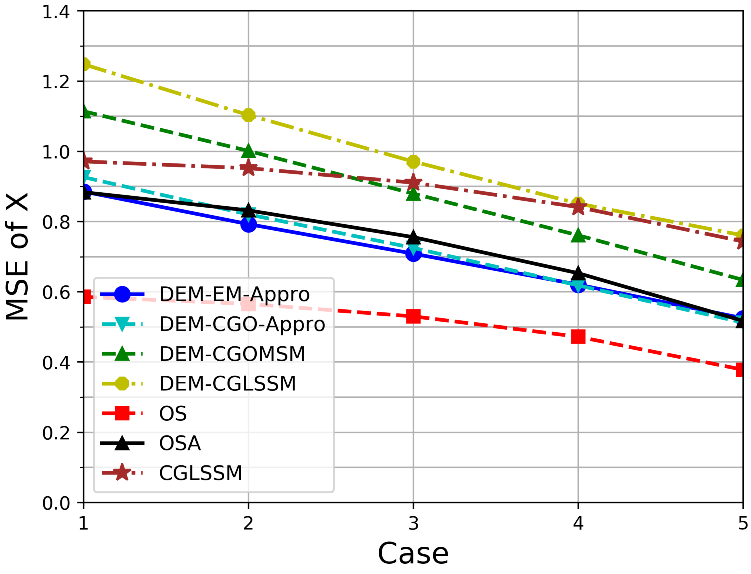
(a) case: $\mathcal{F}^{\mathbf{yx}} = 0.1$.

(b) case: $\mathcal{F}^{\mathbf{yx}} = 0.3$.

Figure 2.11: Error ratio of estimated switches in five different noise levels.



(a) case: $\mathcal{F}^{\mathbf{yx}} = 0.1$.



(b) case: $\mathcal{F}^{\mathbf{yx}} = 0.3$.

Figure 2.12: Restoration MSE of hidden states in five different noise levels.

2.12a and Figure 2.12b, some points can be concluded that:

1) Considering the three supervised cases, OS reaches the optimal restoration. And regarding the two parameter modification based methods, OSA ranks the second (smaller $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ makes it perform closer to OS), and CGLSSM ranks the last.

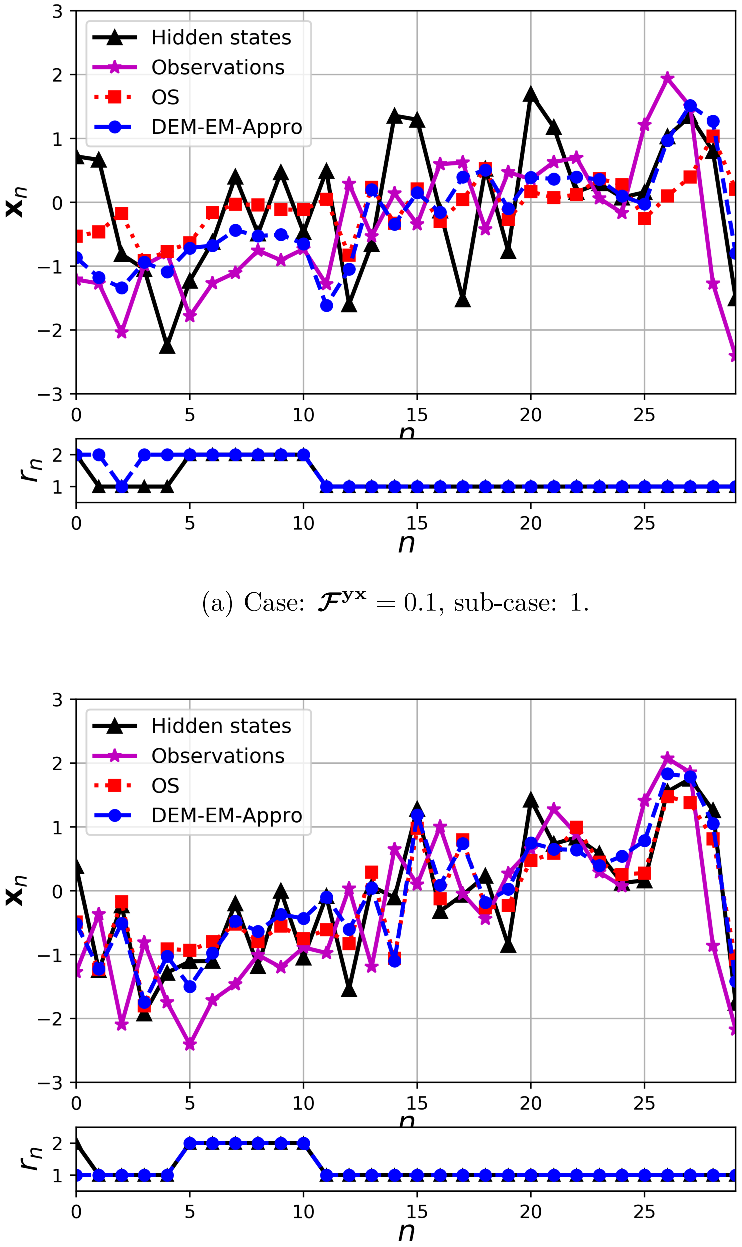
2) For unsupervised cases, whose parameters estimated through Double EM, DEM-EM-Appro and DEM-CGO-Appro get similar efficiency when approaching the restoration, but we prefer DEM-EM-Appro, since the EM-Appro makes no further modification on the estimated parameters which sometimes causes problems and introduces more error. DEM-CGOMSM and DEM-CGLSSM are inferior methods comparing to DEM-EM-Appro and DEM-CGO-Appro, because of the modification on estimated parameters in their entire approximation process. Only when the model comes nearer to CGOMSM, DEM-CGOMSM gets closer to DEM-CGO-Appro (let us recall that under supervised cases, OSA is equal to CGO-Appro when $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$).

3) All methods in either Figure 2.12a or Figure 2.12b show the same tendency that the lower noise level is, the better restoration result they get, although with a little worse estimated $\hat{\mathbf{r}}_1^N$ through unsupervised methods.

4) Relatively, when $d_{j,k}$ increases (equals to increasing $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ in these two series), the noise level is diminished, that is why integrally, OS under the series of case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.3$ has better restoration result than the series under case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$.

5) The affection of $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ on DEM-CGPMSM is kind of subtle, smaller $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ is required by EM in Step A with assumption of $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$, while larger $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ is preferred by Switching EM in Step B. Decreasing $\boldsymbol{\mathcal{F}}^{\mathbf{yx}}$ of the model make Double EM better estimate the switches, but harder for Switching EM to get proper parameters (as proved in Section 2.2.3, when encounter $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0$, the parameters can not be recovered).

Two trajectories of $(\mathbf{x}_1^N, \mathbf{y}_1^N, \mathbf{r}_1^N)$, restored with OS and DEM-EM-Appro, belongs to "sub-case 1" of case $\boldsymbol{\mathcal{F}}^{\mathbf{yx}} = 0.1$ (the most noisy one) and "Case 5" of case



(a) Case: $\mathcal{F}^{\mathbf{yx}} = 0.1$, sub-case: 1.

(b) Case: $\mathcal{F}^{\mathbf{yx}} = 0.3$, sub-case: 5.

Figure 2.13: Examples of a trajectory of $(\mathbf{x}_1^N, \mathbf{y}_1^N, \mathbf{r}_1^N)$ (30 sample points) and restoration with OS and DEM-EM-Appro.

$\mathcal{F}^{\mathbf{yx}} = 0.3$ (the least noisy one) are given respectively as an example in Figure 2.13. It is obvious that, under less noisy case, the observation is closer to hidden state, so as the restorations. Meanwhile, both Figure 2.13a and 2.13b show a closer restoration of DEM-EM-Appro to OS than the observations even with misclassification of switches.

## 2.4 Conclusion

Among all Markov switching models, this Chapter focuses on the recent CGPMSM family extended from GPMM. The CGPMSM considers more complete variable dependence comparing to the widely used CGLSSM, just similar to the advantage brings by GPMM from the classic HGMM. Moreover, benefit from the pairwise structure, under CGPMSM family, there is a special sub-model CGOMSM which allows optimal restoration. The existing supervised restoration approach which does not use MCMC methods in CGPMSM is based on parameter modification to CGOMSM, while according to author's knowledge, no previous work considers the unsupervised restoration of CGPMSM.

This Chapter contributes to enriching both supervised and unsupervised restoration methods in CGPMSM. Firstly, we broaden the scope of filtering in CGOMSM. The reversible CGOMSM is considered, which provides a backward way to approximate the CGPMSM by CGOMSM and experiment shows that, the backward approximation is competitive to the forward one. But leave us the problem to find out a suitable criterion to decide which one is better for a specific case. Secondly, we deal with parameter estimation problem in CGPMSM. The EM method for parameter estimation of GPMM is extended to a switching one with known switches, call Switching EM. Further, with the essential assumption that the processes $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is a PMC, an EM principle based parameter estimation method for CGPMSM is proposed, called Double EM, incorporating the Switching EM. Thirdly, for supervised restoration in CGPMSM, two restoration approaches, "CGO-Appro" and "DEM-Appro" are proposed based on parameter modification to CGOMSM and EM principle respectively. These two approaches are milder comparing to the original parameter modification approach in [1], since they take assumption in their partial process. Experiments conducted proof the efficiency of these two approaches against the other restoration methods including Particle Filter, and that they are much less time consuming comparing to Particle filer. Finally, combining th e Double EM for parameter estimation and the proposed restoration approaches, we get unsupervised restoration solutions for CGPMSM. Simulations show the competitive performances of DEM-EM-Appro among all considered unsupervised strategies, which can even surpass the supervised restoration approaches, such as CGOMSM based one and CGLSSM based one.

Chapter 3

# Non-Gaussian Markov switching model with copulas

The switching models we discussed in previous Chapter are all conditionally Gaussian linear. In this Chapter, we will deal with the non-Gaussian switching models while still considering the feasibility for optimal restorations.

The aim of this Chapter is to propose a new non-Gaussian non-linear switching model, in which optimal restoration is workable. To further explain, we extend the CGOMSM, which allows exact filtering and smoothing, to a more general switching model, in which $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ are no longer limited to be Gaussian or Gaussian mixture, and the auto-regressive functions from $\left(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_{n+1}\right)$ to $\mathbf{x}_{n+1}$ conditionally on $\mathbf{r}_n^{n+1}$ are no longer limited to be linear. The new model, called "Generalized Conditionally Observed Markov Switching Model" (GCOMSM), is based on copulas, which enriches the distributions of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ of CGOMSM.

Developed by Sklar [124], copula has become one of the most popular methods to analyze multiple variables in many fields especially financial markets [89], [56], [53], [110], as it can model flexible multivariate joint distributions in a simple way. All joint distributions with continues margins can be decomposed by univariate marginal distributions and their joining copula, which means that defining the univariate marginal CDF by $F_1, F_2, \cdots, F_d$ and the corresponding univariate densities $f_1, f_2, \cdots, f_d$, the density $f$ of the d-dimensional joint distribution can be represented by

$$f\left(y_1, y_2, \cdots, y_d\right) = c\left(F_1\left(y_1\right), F_2\left(y_2\right), \cdots, F_d\left(y_d\right)\right) \prod_{i=1}^{d} f_i\left(y_i\right), \qquad (3.1)$$

where $c(\cdot):\ [0,1]^d \Rightarrow \mathbb{R}$ is the density of the d-dimensional Copula.

Copulas were firstly introduced into HMC with dependent noise by [24] and the importance of their role in segmentation efficiency is proved in [37]. However, to our best knowledge, no work consider them in switching state-space models. In the GCOMSM which we propose, the couple $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ becomes a HMC-DN with copulas, and the regime $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right)$ is linear on $\mathbf{x}_n$ but can be of any form on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$ (see (3.4)). Optimal restorations are workable for this general model. Moreover, the model identification of time independent GCOMSM is also possible, since in time independent GCOMSM, the stationary distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ with copulas is with possibilities of automatic search of forms of both copulas and margins, and automatic estimation of the parameters associated to the chosen forms from only observations, as recently proposed in [38]. Combining this automatic identification method for finding the distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$, and regime estimation method for finding $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right)$, such as Least-square (LS) principle, can fuse to an integral identification method to learn the form and all necessary parameters for the restoration under GCOMSM.

This chapter is organized as follows. In Section 3.1, we define the general model GCOMSM, give its simulation method, explain its properties and advantages over the classic CGOMSM. The optimal filtering and smoothing of this newly proposed model are derived in Section 3.2 with experiments to show the efficiency of these GCOMSM matched optimal restorations. In Section 3.3, we propose the "GICE-LS" strategy, which combines the Generalized Iterative Conditional Estimation (GICE [38]) and Least-square (LS) parameter estimation for identifying GCOMSM. In Section 3.4, experiments are conducted on simulated data which follows the general GCOMSM to show the appropriate performance of GICE-LS. Then GCOMSM is applied on data of some generable non-linear non-Gaussian models (Kitagawa and Stochastic volatility) with GICE-LS for parameter estimation to get the restoration of their hidden state with comparison to some existing supervised and unsupervised methods. Finally, Section 3.5 concludes the contributions of this Chapter.

## 3.1 Generalization of conditionally observed Markov switching model

Inspired by CGOMSM, which specially has the advantage over the other switching Markov models that optimal filtering and smoothing are possible, we propose this new general model called " Generalized Conditionally Observed Markov Switching Model" (GCOMSM) as an extension of CGOMSM, which incorporates richer distributions and non-linear auto-regressive functions.

### 3.1.1 Definition of GCOMSM

Like the common used Markov switching models, GCOMSM considers three random process $\mathbf{X}_1^N = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N)$, $\mathbf{R}_1^N = (R_1, R_2, \cdots, R_N)$, $\mathbf{Y}_1^N = (\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N)$. Each $\mathbf{X}_n$, $R_n$, $\mathbf{Y}_n$ takes their values in $\mathbb{R}^s$, $\mathbf{\Omega} = \{1, 2, \ldots, K\}$, and $\mathbb{R}^q$ respectively. As usual, the triplet $\left(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is assumed to be a Markov chain. The distribution of $\left(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is defined by the initial distribution $p\left(\mathbf{x}_1, r_1, \mathbf{y}_1\right)$ and the transitions $p\left(r_{n+1} \left| r_n\right.\right) p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{x}_n, \mathbf{y}_n\right.\right)$, which implies the Markovianity of $\mathbf{R}_1^N$. Importantly, we assume that

$$p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{x}_n, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right) p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right), \quad (3.2)$$

which is also the essential property of CGOMSM. The distribution $p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right)$ in equation (3.2) is limited to be Gaussian in CGOMSM as described by equation (2.12), while in GCOMSM, it is enriched by Copula and extended to:

$$\begin{aligned} p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right) = & f_{n+1}\left(\mathbf{y}_{n+1} | \mathbf{r}_{n+1}^{n}\right) \\ & c_{n+1}\left(F_{n+1}\left(\mathbf{y}_n | \mathbf{r}_n^{n+1}\right), F_{n+1}\left(\mathbf{y}_{n+1} | \mathbf{r}_{n+1}^{n}\right) | \mathbf{r}_n^{n+1}\right), \end{aligned} \quad (3.3)$$

where we use $f_{n+1}\left(\mathbf{y}_n | \mathbf{r}_n^{n+1}\right)$ and $f_{n+1}\left(\mathbf{y}_{n+1} | \mathbf{r}_{n+1}^{n}\right)$ to denote the Probability Density Function (PDF) of the left and right margins respectively[1].

[1] (3.3) has been presented in a joint form in equation (1.17) in Section 1.2 when introducing the continuous state-space PMC. To simplify the notation, the left and right margins which are denoted by $(l)$ and $(r)$ in (1.17) will be replaced by the special combination forms of $\mathbf{r}_n^{n+1}$ and $\mathbf{r}_{n+1}^{n}$

Similarly, $F_{n+1}\left(\mathbf{y}_n|\mathbf{r}_n^{n+1}\right)$, $F_{n+1}\left(\mathbf{y}_{n+1}|\mathbf{r}_{n+1}^{n}\right)$ are their associated CDF, while $c_{n+1}\left(F_{n+1}\left(\mathbf{y}_n|\mathbf{r}_n^{n+1}\right), F_{n+1}\left(\mathbf{y}_{n+1}|\mathbf{r}_{n+1}^{n}\right)|\mathbf{r}_n^{n+1}\right)$ represents the density of this two-dimensional Copula. The copula then completes the two margins to form a joint distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$, which makes the model can embrace theoretically, any distribution of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$. As a consequence, the conditional distributions of hidden states are enriched.

Moreover, the simple regime form from $\mathbf{x}_n$, $\mathbf{y}_n$, $\mathbf{y}_{n+1}$ to $\mathbf{x}_{n+1}$ knowing $\mathbf{r}_n^{n+1}$ referred by equation (2.15) in CGOMSM is extended to:

$$\mathbf{x}_{n+1} = \mathbf{A}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right)\mathbf{x}_n + \mathbf{B}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right) + \boldsymbol{\nu}_{n+1}, \tag{3.4}$$

in which, $\mathbf{A}_{n+1}(\cdot)$ and $\mathbf{B}_{n+1}(\cdot)$ can be any function forms of $r_n$, $r_{n+1}$, $\mathbf{y}_n$, $\mathbf{y}_{n+1}$. $\boldsymbol{\nu}_{n+1} \sim \mathcal{N}\left(0, \boldsymbol{\mathcal{V}}_{n+1}\left(\mathbf{r}_n^{n+1}\right)\right)$. We see this regime is only linear on $\mathbf{x}_n$, but can be non-linear on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$.

The higher generality of this GCOMSM compared to CGOMSM is then based on these two extensions. In a word, in CGOMSM, $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right)$ is Gaussian, and thus $p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right)$ and $p\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ are both Gaussian. However, in GCOMSM, $p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right)$ can be of any form, and $p\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ no more needs to be Gaussian. Thus, the CGOMSM can be taken as a special Gaussian linear case of the general GCOMSM[2].

### 3.1.2 Model simulation

As defined in the previous Section, GCOMSM is a switching model in which

$$\begin{aligned} p\left(\mathbf{x}_{n+1}, r_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{x}_n, r_n, \mathbf{y}_n\right.\right) =& p\left(r_{n+1}\left|r_n\right.\right) p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right) \\ & p\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right) \end{aligned}. \tag{3.5}$$

in this Section. One should pay attention that, the PDF of left margin $f_{n+1}\left(\cdot|\mathbf{r}_n^{n+1}\right)$ is not equal to the right margin $f_{n+1}\left(\cdot|\mathbf{r}_{n+1}^{n}\right)$.

[2] When talking about Gaussian linear GCOMSM, Gaussian indicates that the conditional distribution $p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right)$ is Gaussian, and linear refers particularly to $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right)$ linear on $\mathbf{x}_n$.

The simulation of this model can be done in the order: $\mathbf{r}_1^N \Rightarrow \mathbf{y}_1^N \Rightarrow \mathbf{x}_1^N$. Firstly, we simulate the Markov chain $\mathbf{r}_1^N$ according to $p(r_1)$, $p(r_{n+1}|r_n)$. Then as $\left(\mathbf{y}_1^N, \mathbf{r}_1^N\right)$ is a HMC-DN, the Acceptance-Rejection method can be used for simulating $\mathbf{y}_1^N$ knowing $\mathbf{r}_1^N$ [126].

Knowing $\mathbf{r}_n^{n+1}$ and $\mathbf{y}_n$ we write $p\left(\mathbf{y}_{n+1}\left|\mathbf{r}_n^{n+1}, \mathbf{y}_n\right.\right)$ as equation (3.3), the simulation of $\mathbf{y}_{n+1}$ conditionally on $\mathbf{y}_n$ and $\mathbf{r}_n^{n+1}$ can be done with the following two steps:

1. Sample $y$ according to $f_{n+1}\left(y|\mathbf{r}_{n+1}^n\right)$ and $V = v$ according to the uniform law, written as $\mathcal{U}([0,1])$.

2. Accept $\mathbf{y}_{n+1} = y$ if

$$v \leq \frac{c_{n+1}\left(u_1, F_{n+1}\left(y|\mathbf{r}_{n+1}^n\right)|\mathbf{r}_n^{n+1}\right)}{\underset{u_2\in[0,1]}{max}\ c_{n+1}\left(u_1, u_2|\mathbf{r}_n^{n+1}\right)}, \tag{3.6}$$

   where $u_1 = F_{n+1}\left(\mathbf{y}_n|\mathbf{r}_n^{n+1}\right)$.

So that $\mathbf{y}_1^N$ can be generated in series. We listed the marginal distributions and copulas which will be studied in the following statement respectively in Table C.1 and Table C.2 of Appendix C. Table C.3 shows the analytical solutions for $\underset{u_2\in[0,1]}{max}\ c_{n+1}\left(u_1, u_2|\mathbf{r}_n^{n+1}\right)$ of several copulas, those copulas who have no closed-form solution can be numerically maximized.

Finally, having $\mathbf{r}_1^N$ and $\mathbf{y}_1^N$, $\mathbf{x}_1^N$ are easily simulated with equation (3.4).

## 3.2 Optimal restoration in GCOMSM

Similar to CGOMSM, GCOMSM has the advantage that optimal filtering and smoothing are still feasible. Let us recall that filtering calculates $\mathbb{E}\left[\mathbf{X}_n\left|\mathbf{y}_1^n\right.\right]$ and smoothing calculates $\mathbb{E}\left[\mathbf{X}_n\left|\mathbf{y}_1^N\right.\right]$ for which the classical way of calculation has been given in (2.8) and (2.9).

### 3.2.1 Optimal filtering in GCOMSM

As in GCOMSM, we have $p\left(r_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{x}_n, r_n, \mathbf{y}_n \right.\right) = p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right)$, it leads to

$$p\left(\mathbf{x}_n \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right) = p\left(\mathbf{x}_n \left| r_n, \mathbf{y}_1^n \right.\right). \tag{3.7}$$

Besides, since $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1} \right.\right)$ is Gaussian defined in (3.4) in a GCOMSM, we have

$$\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1} \right.\right] = \mathbf{A}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right) + \mathbf{B}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right). \tag{3.8}$$

Then $\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right]$ is computable from $\mathbb{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n \right.\right]$ with

$$\begin{aligned} \mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathbf{X}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right]\right] \\ &= \mathbf{A}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right) \mathbb{E}\left[\mathbf{X}_n \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right] + \mathbf{B}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right) \\ &= \mathbf{A}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right) \mathbb{E}\left[\mathbf{X}_n \left| r_n, \mathbf{y}_1^n \right.\right] + \mathbf{B}_{n+1}\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right), \end{aligned} \tag{3.9}$$

$$\mathbb{E}\left[\mathbf{X}_{n+1} \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right] = \sum_{r_n} p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right) \mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1} \right.\right]. \tag{3.10}$$

in which $p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right)$ is computable because of the Markovianity of $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$. More precisely, we can write

$$p\left(r_n \left| r_{n+1}, \mathbf{y}_1^{n+1} \right.\right) = \frac{p\left(\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right)}{\sum_{r_n} p\left(\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right)}, \tag{3.11}$$

and $p\left(\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right)$ can be calculated recursively with

$$\begin{aligned} p\left(\mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right) &= \sum_{r_{n-1}} p\left(r_{n-1}, \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right) \\ &= \sum_{r_{n-1}} p\left(r_{n-1}^n, \mathbf{y}_1^n\right) p\left(r_{n+1}, \mathbf{y}_{n+1} \left| r_n, \mathbf{y}_n \right.\right) \\ &= \sum_{r_{n-1}} p\left(r_{n-1}^n, \mathbf{y}_1^n\right) p\left(r_{n+1} \left| r_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{y}_n \right.\right), \end{aligned} \tag{3.12}$$

where $p\left(\mathbf{y}_{n+1}\left|\mathbf{r}_n^{n+1},\mathbf{y}_n\right.\right)$ is computed from its distribution set by equation (3.3). Finally, the filtering is given by

$$\mathbb{E}\left[\mathbf{X}_{n+1}\left|r_{n+1},\mathbf{y}_1^{n+1}\right.\right]=\sum_{r_{n+1}}p\left(r_{n+1}\left|\mathbf{y}_1^{n+1}\right.\right)\mathbb{E}\left[\mathbf{X}_{n+1}\left|r_{n+1},\mathbf{y}_1^{n+1}\right.\right]. \tag{3.13}$$

### 3.2.2 Optimal smoothing in GCOMSM

For smoothing, as in GCOMSM we have $p\left(\mathbf{y}_{n+1}^N\left|\mathbf{x}_n,r_n,\mathbf{y}_1^n\right.\right)=p\left(\mathbf{y}_{n+1}^N\left|r_n,\mathbf{y}_1^n\right.\right)$ from the Markovianity of the PMC $\left(\mathbf{R}_1^N,\mathbf{Y}_1^N\right)$, only the update of $p\left(r_n\left|\mathbf{y}_1^N\right.\right)$ instead of $p\left(r_n\left|\mathbf{y}_1^n\right.\right)$ brings new information to smoothing comparing to filtering. The smoothing writes

$$\begin{aligned}\mathbb{E}\left[\mathbf{X}_n\left|\mathbf{y}_1^N\right.\right]&=\sum_{r_n}p\left(r_n\left|\mathbf{y}_1^N\right.\right)\mathbb{E}\left[\mathbf{X}_n\left|r_n,\mathbf{y}_1^N\right.\right]\\&=\sum_{r_n}p\left(r_n\left|\mathbf{y}_1^N\right.\right)\mathbb{E}\left[\mathbf{X}_n\left|r_n,\mathbf{y}_1^n\right.\right].\end{aligned} \tag{3.14}$$

We see that optimal restorations in GCOMSM have quite similar forms as the optimal restorations in CGOMSM. Nevertheless, their data distributions are quite different since the distributions of $\mathbf{Y}_1^N$ conditional on $\mathbf{R}_1^N$ are defined with the distributions $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ of any form in GCOMSM, while they are all Gaussian mixture in general CGOMSM. Also, the distribution $p\left(\mathbf{x}_n^{n+1},\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ is quite different from the general CGOMSM which can only be Gaussian mixtures.

In the remaining of this Chapter, we will focus on the time independent case of the general GCOMSM, which means that the parameters depend only on the switches $\left(\mathbf{r}_n^{n+1}\right)$, since we are going to tackle the parameter estimation problem. Further, we reduce the number of margins and copulas which construct $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ with assumption that the pair $\left(\mathbf{r}_1^N,\mathbf{y}_1^N\right)$ is stationary reversible, which implies that $p\left(\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right)$ does not depend on $n$ and the distribution $p\left(r_{n+1},\mathbf{y}_{n+1}\left|r_n,\mathbf{y}_n\right.\right)$ and $p\left(r_n,\mathbf{y}_n\left|r_{n+1},\mathbf{y}_{n+1}\right.\right)$ are equal. These assumptions result in

$$p\left(\mathbf{y}_{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)=p\left(\mathbf{y}_{n+1}\left|r_{n+1}\right.\right). \tag{3.15}$$

So, we do not need to consider the margin is “left” or “right” any more as they are the same under these assumptions. In this simple GCOMSM, the definition of $p\left(\mathbf{y}_{n+1}\left|\mathbf{r}_n^{n+1},\mathbf{y}_n\right.\right)$ in (3.3) is simplified to

$$p\left(\mathbf{y}_{n+1}\left|\mathbf{r}_n^{n+1},\mathbf{y}_n\right.\right)=f_{r_{n+1}}\left(\mathbf{y}_{n+1}\right)c_{\mathbf{r}_n^{n+1}}\left(F_{r_n}\left(\mathbf{y}_n\right),F_{r_{n+1}}\left(\mathbf{y}_{n+1}\right)\right). \tag{3.16}$$

The dependence on switches is then moved to subscript in $f_{r_n}\left(\mathbf{y}_n\right)$ since $n$ is no more needed for referring the time, so as in the other expressions of distributions and functions. The time independent auto-regressive function of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n,\mathbf{r}_n^{n+1},\mathbf{y}_n^{n+1}\right.\right)$ in (3.4) becomes

$$\mathbf{X}_{n+1}=\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)\mathbf{x}_n+\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)+\boldsymbol{\nu}_{n+1}, \tag{3.17}$$

with $\boldsymbol{\nu}\sim\mathcal{N}\left\{0,\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}\right\}$. It is the same that sometimes we write integrally

$$\mathbf{x}_{n+1}\sim\mathcal{N}\left\{\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)\mathbf{x}_n+\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right),\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}\right\}. \tag{3.18}$$

We should notice that, the model is set to be time independent, which means $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ and $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n,\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ are both time independent. However, $p\left(\mathbf{x}_n^{n+1},\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ are complex mixtures that may be not stable, which means that the model could be non-stationary.

### 3.2.3 Examples of GCOMSM and the optimal restoration in them

We present here two experimental examples to show the flexibility of the proposed GCOMSM, as well as the performance of its optimal filtering and smoothing. First example aims to verify that the CGOMSM can be considered as a special Gaussian linear case of GCOMSM. Then the second one is a general non-Gaussian non-linear[3] case of GCOMSM.

Both these two examples assume that the Markov chain $\mathbf{R}_1^N$ has $K=2$, and $p_{1,1}=p_{2,2}=0.45$, $p_{1,2}=p_{2,1}=0.05$. To further simplify the notification, we will denote similarly $f_{j,k}\left(\mathbf{y}_n^{n+1}\right)=f_{r_n=j,r_{n+1}=k}\left(\mathbf{y}_n^{n+1}\right)$, $f_j\left(\mathbf{y}_n\right)=f_{r_n=j}\left(\mathbf{y}_n\right)$, $c_{j,k}\left(F_j\left(\mathbf{y}_n\right),F_k\left(\mathbf{y}_{n+1}\right)\right)=c_{r_n=j,r_{n+1}=k}\left(F_j\left(\mathbf{y}_n\right),F_k\left(\mathbf{y}_{n+1}\right)\right)$ with $F_{j,k}$, $F_j$, $C_{j,k}$ the

[3] the regime of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n,\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ is non-linear on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$.

associated cumulative functions, $j,\ k \in \mathbf{\Omega}$. And in (3.17) the abbreviation is like $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = \mathbf{A}_{r_n=j,r_{n+1}=k,}\left(\mathbf{y}_n^{n+1}\right)$, so as the other notations. The details of the form of all marginal distributions and copulas applied can be found in Appendix C. 1000 samples are simulated from GCOMSM under specific settings for restoration. All results presented are average of 100 independent experiments.

#### 3.2.3.1 Example 1 – Gaussian linear case

We set both the margins and copulas of the joint distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ be Gaussian, so that $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ is actually a two-dimensional Gaussian distribution. As we consider a stationary reversible case of $\left(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right)$, the distributions in (3.16) is then constructed by two different margins: $f_1\left(\mathbf{y}_n\right)$, $f_2\left(\mathbf{y}_n\right)$ and four copulas $c_{1,1}\left\{F_1\left(\mathbf{y}_n\right), F_1\left(\mathbf{y}_{n+1}\right)\right\}$, $c_{1,2}\left\{F_1\left(\mathbf{y}_n\right), F_2\left(\mathbf{y}_{n+1}\right)\right\} = c_{2,1}\left\{F_2\left(\mathbf{y}_n\right), F_1\left(\mathbf{y}_{n+1}\right)\right\}$, $c_{2,2}\left\{F_2\left(\mathbf{y}_n\right), F_1\left(\mathbf{y}_{n+1}\right)\right\}$, and the parameter sets are:

- Margins: $\theta_1 = \{loc_1 = 0.0, scale_1 = 2.0\}$, $\theta_2 = \{loc_2 = 1.0, scale_2 = 1.0\}$;

- Copulas: $\alpha_{1,1} = 0.7$, $\alpha_{1,2} = \alpha_{2,1} = 0.5$, $\alpha_{2,2} = 0.3$,

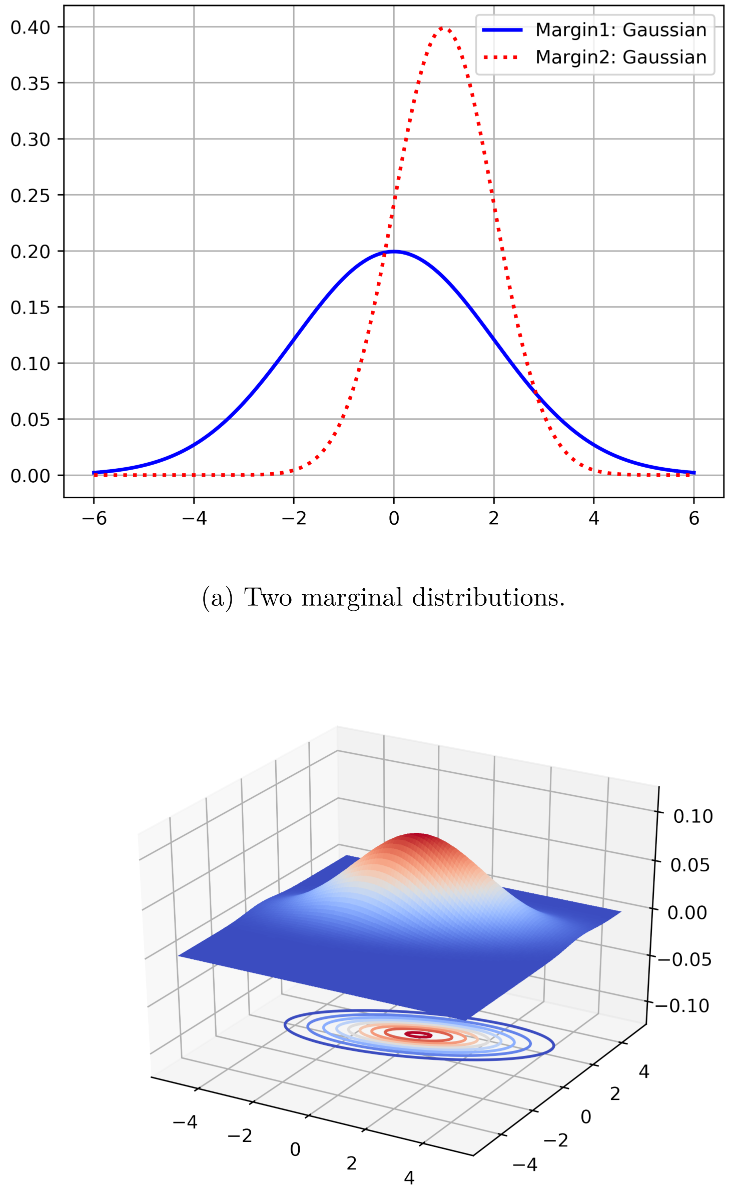
in which $loc_j$ and $scale_j$ represent the mean and standard deviation of Gaussian distribution of the margin $f_j\left(\mathbf{y}_n\right)$, while $\alpha_{j,k}$ denotes the only (linear correlation) parameter of Gaussian copula $c_{j,k}\left\{F_j\left(\mathbf{y}_n\right), F_k\left(\mathbf{y}_{n+1}\right)\right\}$ with $\forall j, k \in \mathbf{\Omega}$. See details of the Gaussian margin and copula in Appendix C.

Figure 3.1a shows the two Gaussian marginal distributions of $\left(\mathbf{y}_n | r_n = 0\right)$ and $\left(\mathbf{y}_n | r_n = 1\right)$. Figure 3.1b shows the joint Gaussian distribution of $\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = 0, r_{n+1} = 1\right)$ set above.

The parameters of $p\left(\mathbf{x}_{n+1}\left|\mathbf{r}_n^{n+1}, \mathbf{x}_n, \mathbf{y}_n^{n+1}\right.\right)$ defined in (3.17) are set to be more CGOMSM like, as defined in (2.7) (but with a reverse of places of $\mathbf{x}_n$ and $\mathbf{y}_{n+1}$), from $p\left(\mathbf{x}_n, \mathbf{x}_{n+1}\left|\mathbf{y}_n^{n+1}, r_n = j, r_{n+1} = k\right.\right) = \mathcal{N}\left\{\boldsymbol{\mu}_{j,k}, \boldsymbol{\sigma}_{j,k}^2\right\}$, parameterized with

$$\boldsymbol{\mu}_{j,k} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_j & \mathbf{0} \\ \boldsymbol{\mathcal{F}}_{k,j} & \boldsymbol{\mathcal{F}}_k \end{bmatrix} \begin{bmatrix} \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix}, \quad \boldsymbol{\sigma}_{j,k}^2 = \begin{bmatrix} \mathbf{\Gamma}_j & \left(\mathbf{\Sigma}_{k,j}\right)^\intercal \\ \mathbf{\Sigma}_{k,j} & \mathbf{\Gamma}_k \end{bmatrix}. \tag{3.19}$$

Similar to the zero set for $\boldsymbol{\mathcal{F}}_{n+1}^{\mathbf{yx}}(\mathbf{R}_n^{n+1})$ in (2.7), the zero set in the expression of



(a) Two marginal distributions.

(b) Joint distribution of $\big(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = 1, r_{n+1} = 2\big)$.

Figure 3.1: The distributions in Example 1.

$\boldsymbol{\mu}_{j,k}$ here leave out the direct relation between $\mathbf{y}_{n+1}$ and $\mathbf{x}_n$, which assures the Markovianity of the pair $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ in such a linear case. For this experimental example, $\boldsymbol{\mu}_{j,k}$ and $\boldsymbol{\sigma}_{j,k}^2$ are assigned by

- $\boldsymbol{\mu}_{j,k}$: $\boldsymbol{\mathcal{F}}_1 = 0.3$, $\boldsymbol{\mathcal{F}}_2 = 0.7$, $\boldsymbol{\mathcal{F}}_{1,2} = 0.2$, $\boldsymbol{\mathcal{F}}_{2,1} = 0.6$;

- $\boldsymbol{\sigma}_{j,k}^2$: $\boldsymbol{\Gamma}_1 = \boldsymbol{\Gamma}_2 = 1.0$, $\boldsymbol{\Sigma}_{1,1} = 0.3$, $\boldsymbol{\Sigma}_{2,2} = 0.7$, $\boldsymbol{\Sigma}_{1,2} = \boldsymbol{\Sigma}_{2,1} = 0.5$.

Then the equivalent parameters of $p\left(\mathbf{x}_{n+1} \left| \mathbf{r}_n^{n+1}, \mathbf{x}_n, \mathbf{y}_n^{n+1}\right.\right)$ are

$$\begin{aligned}
&\mathbf{A}_{j,k} = \boldsymbol{\Sigma}_{k,j}\left(\boldsymbol{\Gamma}_j\right)^{-1},\ \mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = \boldsymbol{\mathcal{F}}_k \mathbf{y}_{n+1} + \Big(\boldsymbol{\mathcal{F}}_{k,j} - \boldsymbol{\Sigma}_{k,j}\left(\boldsymbol{\Gamma}_j\right)^{-1}\boldsymbol{\mathcal{F}}_j\Big)\mathbf{y}_n,\\
&\boldsymbol{\mathcal{V}}_{j,k} = \boldsymbol{\Gamma}_k - \boldsymbol{\Sigma}_{k,j}\left(\boldsymbol{\Gamma}_j\right)^{-1}\boldsymbol{\Sigma}_{j,k}.
\end{aligned} \tag{3.20}$$

$\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ in the original form has becomes a constant so it turns to $\mathbf{A}_{j,k}$.

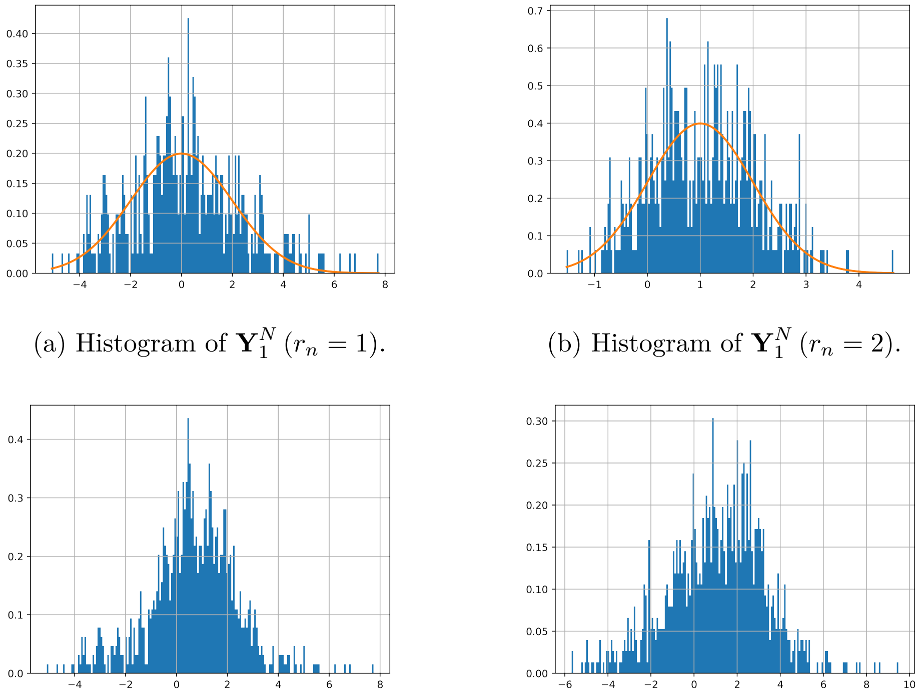
1000 samples are simulated according to the setting of this general GCOMSM. The histograms of the data are displayed in Figure 3.2, in which the sub-figures 3.2a and 3.2b show the histogram of $\mathbf{y}_1^N$ classified by two different values of $r_n$ (with orange lines indicating the distributions they follow). Sub-Figures 3.2c and 3.2d show respectively the histogram of the total simulated $\mathbf{y}_1^N$ and $\mathbf{x}_1^N$, they are both Gaussian mixtures.

Both optimal filtering and smoothing for GCOMSM are processed to restore the switches and hidden states from only observations. MPM criterion is applied on $p\left(r_n \left|\mathbf{y}_1^n\right.\right)$ and $p\left(r_n \left|\mathbf{y}_1^N\right.\right)$ for getting the filtering and smoothing estimation of $\mathbf{r}_1^N$.

All restoration results are listed in Table 3.1. An improvement of both the estimated $\mathbf{r}_1^N$ and $\mathbf{x}_1^N$ can be observed from the filtering to smoothing, but not too much regarding the MSE, since the $\mathbf{y}_{n+1}^N$ brings no more information for $\mathbb{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^N\right.\right]$ in smoothing comparing to $\mathbb{E}\left[\mathbf{X}_n \left|r_n, \mathbf{y}_1^n\right.\right]$ in the filtering.

Table 3.1: Restoration result of Example 1.

| Observation | Exact filtering | | Exact smoothing | |
|---|---|---|---|---|
| MSE | Error ratio | MSE | Error ratio | MSE |
| 2.328 | 0.261 | 1.086 | 0.229 | 1.079 |



(a) Histogram of $\mathbf{Y}_1^N$ ($r_n = 1$).

(b) Histogram of $\mathbf{Y}_1^N$ ($r_n = 2$).

(c) Histogram of $\mathbf{Y}_1^N$.

(d) Histogram of $\mathbf{X}_1^N$.

Figure 3.2: Histograms of simulated data of Example 1 (Gaussian linear case).

An instance of trajectories in this series is given in Figure 3.3 which shows the visual performance of these two exact restorations. The restoration of optimal filtering is actually quite close to the smoothing one that we can not distinguish them by naked eye.

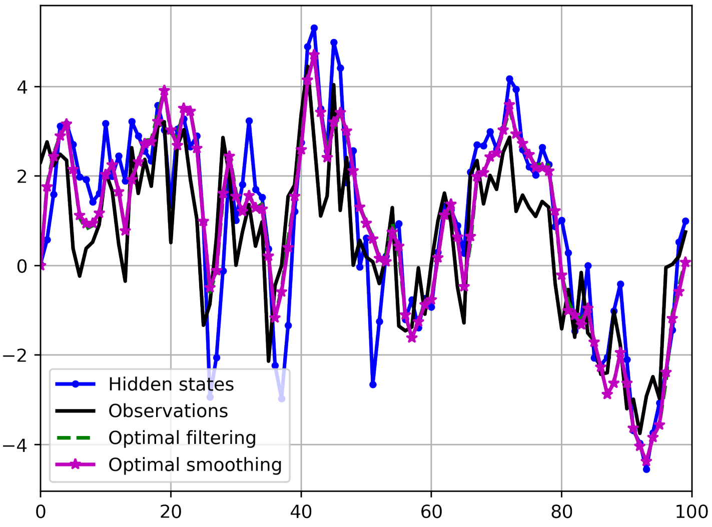


Figure 3.3: Trajectories of Example 1 (100 samples, Gaussian linear case).

#### 3.2.3.2 Example 2 – non-Gaussian non-linear case

Let us turn to an example of general GCOMSM which has non-Gaussian $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ and non-linear regime for $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$.

The parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$, which are supposed to be non-Gaussian, are set according to:

- Margins: $f_1\left(\mathbf{y}_n\right) = \text{Beta}^{4}\left\{\alpha_1 = 1, \beta_1 = 1, loc_1 = -3, scale_1 = 4\right\}$,

  $f_2\left(\mathbf{y}_n\right) = \text{Laplace}\left\{loc_2 = 0, scale_2 = 1\right\}$.

- Copulas: $c_{1,1}\left\{\cdot,\cdot\right\} = \text{Arch12}^{5}\left\{\cdot,\cdot|\alpha_{1,1} = 2\right\}$,

  $c_{2,2}\left\{\cdot,\cdot\right\} = \text{FGM}\left\{\cdot,\cdot|\alpha_{2,2} = 0.5\right\}$,

[4] With the setting $\alpha_1 = 1$, $\beta_1 = 1$, the Beta distribution is equal to a uniform distribution.

[5] Short for Archimiedean copula, order: 12.

$$c_{1,2}\left\{\cdot,\cdot\right\} = c_{2,1}\left\{\cdot,\cdot\right\} = \mathrm{Arch14}^{6}\left\{\cdot,\cdot|\alpha_{1,2}=3\right\}.$$

where $loc_j$ is short for location of $f_j\left(\mathbf{y}_n\right)$, and $scale_j$ represents its scale. How the parameters form this set of margins and copulas are detailed in Appendix C. We see the two marginal distributions of $p\left(\mathbf{y}_n\left|r_n\right.\right)$ and the joint distribution of $p\left(\mathbf{y}_n^{n+1}\left|r_n=1,r_{n+1}=2\right.\right)$ in Figure 3.4, they are far from Gaussian distributions and hard to be approximated by Gaussian mixture distribution with a small component number.

$p\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n,\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right) = \mathcal{N}\left\{\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)+\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right),\boldsymbol{\mathcal{V}}_{j,k}\right\}$ is set with $\mathbf{A}_{j,k} = a_{j,k}$, simple non-linear function on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$ that $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = b_{j,k}\sqrt{|\mathbf{y}_{n+1}|}$, and in which the parameters are assigned as

- $\mathbf{A}_{j,k}$: $\mathbf{A}_{1,1}=0.2$, $\mathbf{A}_{1,2}=0.4$, $\mathbf{A}_{2,1}=0.6$, $\mathbf{A}_{2,2}=0.8$,

- $b_{j,k}$: $b_{1,1}=0.7$, $b_{1,2}=0.5$, $b_{2,1}=0.6$, $b_{2,2}=0.9$,

- $\boldsymbol{\mathcal{V}}_{j,k}$: $\boldsymbol{\mathcal{V}}_{1,1}=\boldsymbol{\mathcal{V}}_{2,2}=1.0$, $\boldsymbol{\mathcal{V}}_{1,2}=\boldsymbol{\mathcal{V}}_{2,1}=0.8$.

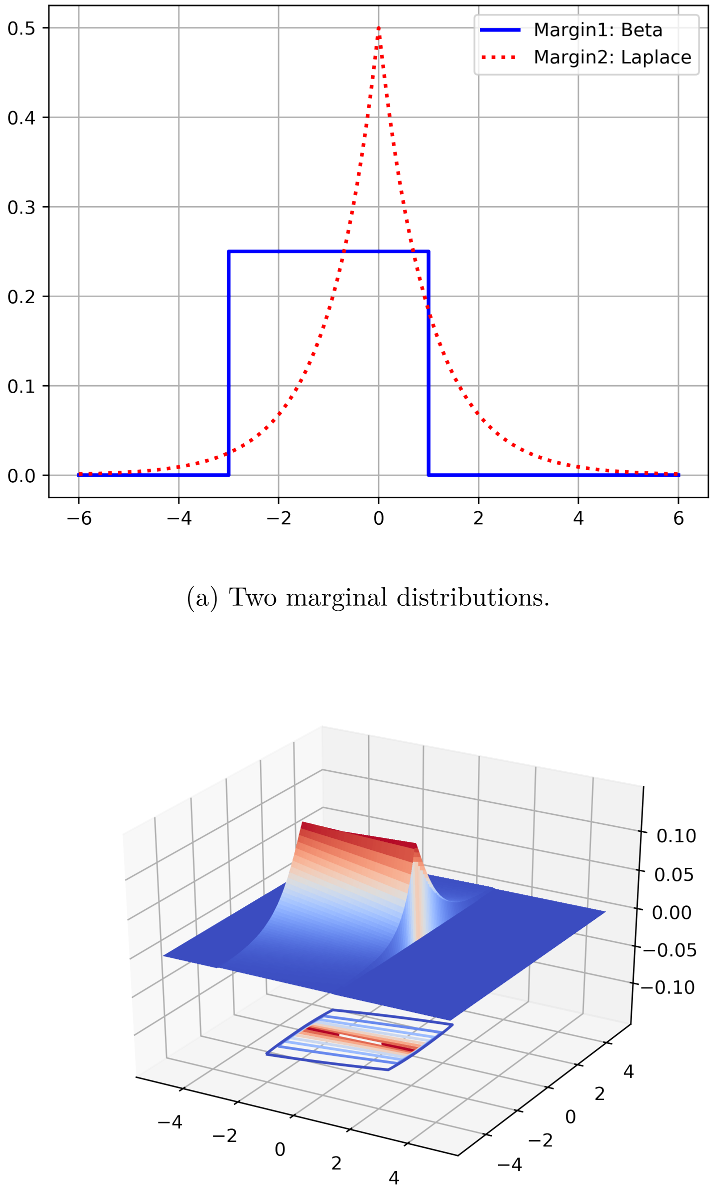
The histograms of the simulated data follows the general GCOMSM with all the setting above are given in Figure 3.5. We see the two non-Gaussian Margins makes the integral histogram of $\mathbf{y}_1^N$ an odd shape. The $\mathbf{x}_1^N$ is also non-Gaussian in spite of the conditional Gaussian assumption in GCOMSM that $p\left(\mathbf{x}_n^{n+1}\left|\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ should be Gaussian.

The restoration efficiency of optimal filtering and smoothing on this general case of GCOMSM is proofed by the results listed in Table 3.2. As $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ is set in a non-linear form, observation is quite different from the hidden state, but the restorations still work well. One trajectories is given in Figure 3.6 as an example.

Table 3.2: Restoration result of example 2.

| Observation | Filtering | | Smoothing | |
|---|---|---|---|---|
| MSE | Error Ratio | MSE | Error Ratio | MSE |
| 11.496 | 0.224 | 2.144 | 0.193 | 2.093 |

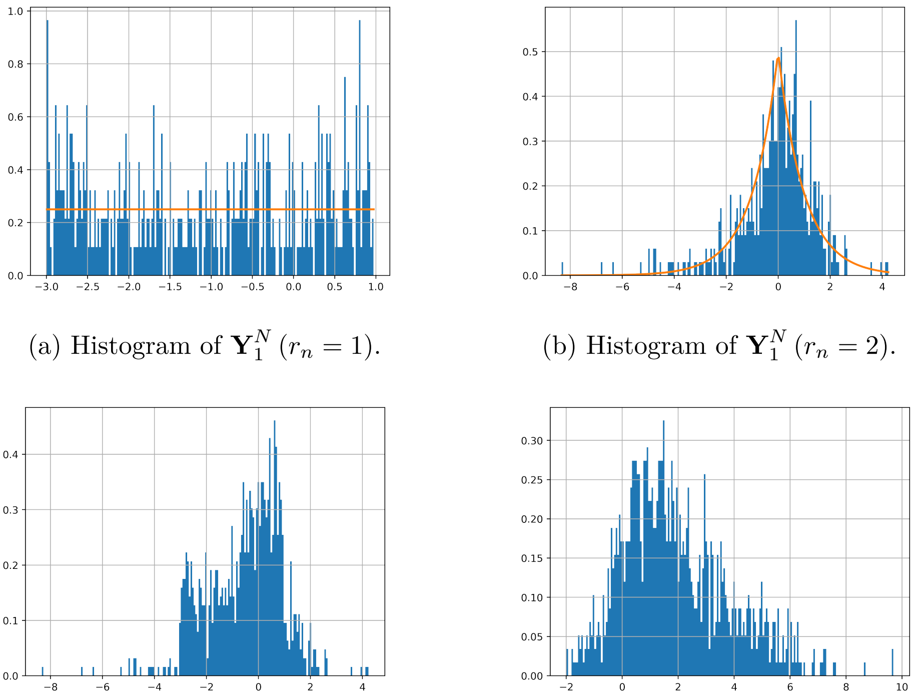
[6] Short for Archimiedean copula, order: 14.



(a) Two marginal distributions.

(b) Joint distribution of $\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = 1, r_{n+1} = 2\right)$.

Figure 3.4: The distributions in Example 2.



(a) Histogram of $\mathbf{Y}_1^N$ ($r_n = 1$).

(b) Histogram of $\mathbf{Y}_1^N$ ($r_n = 2$).

(c) Histogram of $\mathbf{Y}_1^N$.

(d) Histogram of $\mathbf{X}_1^N$.

Figure 3.5: Histograms of simulated data of Example 2 (non-Gaussian non-linear case).

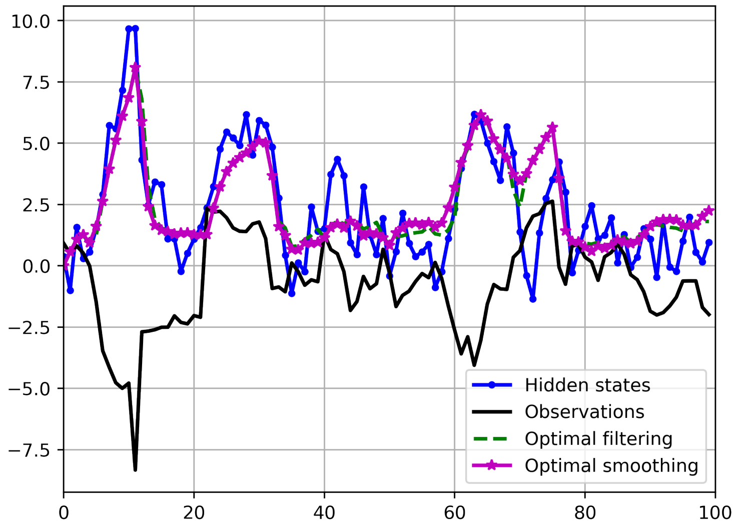


Figure 3.6: Trajectories of Example 2 (100 samples, non-Gaussian non-linear case).

## 3.3 Model identification

In the former Sections, we have defined the GCOMSM, and show how the optimal filtering and smoothing work. From this Section, we start to tackle how to approach a noised non-Gaussian non-linear system by the time independent GCOMSM with its learning sample set. The model identification problem we are facing is multi-folds:

1. what forms the distributions of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ are of;

2. once we have the distribution of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$, how to get the related parameters;

3. what the forms of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)$ are in the regime of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, r_n, \mathbf{y}_n^{n+1}\right.\right)$;

4. what the parameters of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)$ are once their forms are known[7].

We deal with the problems above simultaneously which will result in a general strategy for model identification and parameter estimation for GCOMSM.

### 3.3.1 Generalized iterative conditional estimation

Let us deal with the first two problems, which are how to identify the conditional distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ from the observation of the learning sample. As no confusion introduced, when we are discussing about the identification problem, $\left(\mathbf{x}_1^N, \mathbf{y}_1^N\right)$ represents the data of learning sample set. To solve the first two problem, we use an original variant of the "Generalized Iterative Conditional Estimation" (GICE) method, recently proposed in [38]. GICE is a generalization of "Iterative Conditional Estimation" (ICE) which has been introduced and applied in Chapter 1 as an alternative method to EM. ICE is an iterative method works on the parameter estimation of stationary Gaussian PMC, while the GICE is not limited for Gaussian

[7] $\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}$ is also a parameter, but if aiming at restoration, we can see from Chapter 3.2 that it is not necessary in neither the computation of filtering nor smoothing.

case. It searches the proper form of distributions from an expected form set for PMC and also give the estimated parameters.

For the stationary reversible case we are dealing with here, knowing $f_{j,k}\left(\mathbf{y}_1^2\right)$ is equivalent to knowing $f_j\left(\mathbf{y}_1\right)$, $f_k\left(\mathbf{y}_2\right)$ and $c_{j,k}\left(F_j\left(\mathbf{y}_1\right), F_k\left(\mathbf{y}_2\right)\right)$. For each pair $j,k \in \mathbf{\Omega} = \{1,\ldots,K\}$, the forms of $f_j(\cdot)$, $f_k(\cdot)$ are unknown, but we assume that they belong to a known set of possible forms $\mathbf{H} = \{H_1,\cdots,H_L\}$; besides, each form $H_l$, $l \in \{1,\cdots,L\}$ is a parametric set of probability distributions $H_l = \left\{f_{\theta(l)}\right\}_{\theta(l)\in\boldsymbol{\theta}(l)}$. Similarly, $c_{j,k}(\cdot,\cdot)$ is unknown, but it belongs to a known set of possible forms $\mathbf{G} = \{G_1,\cdots,G_M\}$ and each of them is a parametric set of copulas $G_m = \left\{c_{\alpha(m)}\right\}_{\alpha(m)\in\boldsymbol{\alpha}(m)}$, with $m \in \{1,\cdots,M\}$. Finally, for each $j,k \in \mathbf{\Omega}$, the two former problems we are tackling under all these assumptions is to find from $\mathbf{y}_1^N$:

1. The proper forms $H_l$ and $G_m$;

2. The proper parameters $\theta(l)$ and $\alpha(m)$.

To solve these questions, we need two families of estimators. Firstly, for each $l \in \{1,\ldots,L\}$, we assume that an estimator $\hat{\theta}(l)(\mathbf{y}_1^N)$ exists for giving $\theta(l)$ of the marginal distribution $p\left(\mathbf{y}_n\right)$ from $\mathbf{y}_1^N$, with the marginal distribution $p\left(\mathbf{y}_n\right)$ equals everywhere through $\mathbf{y}_1^N$ and belong to $H_l$. Secondly, for each $m \in \{1,\ldots,M\}$, another estimator $\hat{\alpha}(m)(\mathbf{y}_1^N)$ exists to estimate the parameter $\alpha(m)$ of the Copula $c(\mathbf{y}_n^{n+1})$ from $\mathbf{y}_1^N$, with $c(\mathbf{y}_n^{n+1})$ equal everywhere through all $\mathbf{y}_1^N$ and belong to $G_m$.

Having the parameters estimated for all possible margins and copulas, we need to decide the best fit distribution constructed by the best fit margins and copulas, which needs also two decision rules. For each $j,k \in \mathbf{\Omega}$, we note the two required "decision rules" by $\mathcal{D}^1$ and $\mathcal{D}^2$. They are applied on the observation sample $\mathbf{y}_1^N$. For any $f_{\theta(1)} \in H_1,\ldots,f_{\theta(L)} \in H_L$, $\mathcal{D}^1$ selects an unique element in the candidate margins forms $\left\{f_{\theta(1)},\ldots,f_{\theta(L)}\right\}$ corresponding to $\mathbf{y}_1^N$; and for any $c_{\alpha(1)} \in G_1,\ldots,c_{\alpha(M)} \in G_M$, $\mathcal{D}^2$ selects an unique element in the candidate copula forms $\left\{c_{\alpha(1)},\ldots,c_{\alpha(M)}\right\}$ corresponding to $\mathbf{y}_1^N$.

Dealing with all of these problems, the GICE is an iterative method who runs the following steps (with $i$ denotes the iterations):

1. Initialize GICE with $\left(p^0_{j,k}, f^0_j, c^0_{j,k}\right)$ (for each $j, k \in \mathbf{\Omega}$ ) found with a simple method.
2. Find $\left(p^{i+1}_{j,k}, f^{i+1}_j, c^{i+1}_{j,k}\right)$ from $\left(p^i_{j,k}, f^i_j, c^i_{j,k}\right)$ and $\mathbf{y}^N_1$.
   (a) set $p^{i+1}_{j,k} = \frac{1}{N-1}\sum_{n=1}^{N-1} p^i\left(r_n = j, r_{n+1} = k|\mathbf{y}^N_1\right)$ with $p^i\left(r_n = j, r_{n+1} = k|\mathbf{y}^N_1\right)$ computed from $\left(p^i_{j,k}, f^i_j, c^i_{j,k}\right)$ (the computation has been introduced in Chapter 1 from (1.7) to (1.14)):
   (b) sample $\left(\mathbf{r}^N_1\right)^{i+1}$ according to $p\left(r_{n+1}\left|r_n, \mathbf{y}^N_1\right.\right)$ based on parameters $\left(p^i_{j,k}, f^i_j, c^i_{j,k}\right)$;
   (c) for each $j, k \in \mathbf{\Omega}$, consider $\left(\mathbf{y}^N_1\right)^{i+1}_j$ the sub-sequence of $\mathbf{y}^N_1$ which corresponds to $r^{i+1}_n = j$, and $\left(\mathbf{y}^N_1\right)^{i+1}_{j,k}$ the sub-sequence of couples $\left(\mathbf{y}_n, \mathbf{y}_{n+1}\right)$ which corresponds to $\left(r^{i+1}_n = j, r^{i+1}_{n+1} = k\right)$;
   (d) for each $j, k \in \mathbf{\Omega}$, each $l \in \{1, \ldots, L\}$ and each $m \in \{1, \ldots, M\}$, calculate $\theta^{i+1}_j(l) = \hat{\theta}_j(l)\left[(\mathbf{y}^N_1)^{i+1}_j\right]$ and $\alpha^{i+1}_{j,k}(m) = \hat{\alpha}_{j,k}(m)\left[(\mathbf{y}^N_1)^{i+1}_{j,k}\right]$;
   (e) apply the decision rule $\mathcal{D}^1$ to chose uniquely the element $f^{q+1}_j$ in $\left\{f_{\theta^{q+1}_j(1)}, \ldots, f_{\theta^{q+1}_j(L)}\right\}$, and $\mathcal{D}^2$ to determine uniquely the $c^{q+1}_{j,k}$ in $\left\{c_{\alpha^{q+1}_{j,k}(1)}, \ldots, c_{\alpha^{q+1}_{j,k}(L)}\right\}$.
3. Stop according to some criterion.

As the GICE is a general frame for finding the distribution forms and their parameters, different possible ways can be included for parameter estimation, and for the decision rules. In this dissertation, when conducting the GICE principle, we adopt Kolmogorov distance [59] for the decision rule $\mathcal{D}^1$, while the original paper [38] is based on Pearson's system. Besides, all the $\hat{\theta}\,(l)$ and $\hat{\alpha}\,(m)$ are estimated through Maximum Likelihood (ML), while in [38] $\hat{\alpha}\,(m)$ are obtained with the empirical estimation of Kendall's tau.

### 3.3.2 Least-square parameter estimation for non-linear switching model

Assuming that the first two problems are figured out, which means that we know the distribution $p\left(\mathbf{y}^{n+1}_n\left|\mathbf{r}^{n+1}_n\right.\right)$. So, what we are facing now, are the remaining two problems which deal with the form and parameters of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)$. We are going to give a simple way to figure out only the parameters which are necessary for the CGOMSM restoration under a suitable assumption on their forms.

When $p\left(\mathbf{r}_1^N \left|\mathbf{y}_1^N\right.\right)$ is given, the parameter estimation of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ can be considered as estimation of a multi-regimes switching regression. Then normally, LS is an efficient method to figure out the parameters of this system in classical way. See the computation of $\mathbb{E}\left[\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_1^{n+1}\right.\right]$ in equation (3.9), only parameters in $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)$ are necessary for the exact restoration. So the simplest is to disregard the heteroscedasticity of the error terms in the switching regimes regressions. Interpreted in the GCOMSM, it is to neglect the variance items $\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}$. Then, the Ordinary Least-square (OLS) aims to minimize

$$\begin{aligned}\boldsymbol{e}^2 =& \frac{1}{N-1}\sum_{1}^{N-1}\left\{\mathbf{x}_{n+1} - \sum_{r_n}\sum_{r_{n+1}} p\left(\mathbf{r}_n^{n+1}\left|\mathbf{y}_1^N\right.\right)\right.\\ &\left.\left[\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)\mathbf{x}_n + \mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)\right]\right\}^2.\end{aligned} \tag{3.21}$$

In this way, we treat each $\mathbf{x}_{n+1}$ be the same informative about the underlying relationship of $\left(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right)$[8]. If not, we need to turn to Weighted Least-square (WLS) for solution, which considers $\mathbf{x}_{n+1}$ as more or less informative and gives more "weight" for the more informative ones while doing the minimization. The weight should be the reciprocal of the variance of $\left(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n^{n+1}\right)$ which can be computed by $\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}$ as $\left(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n^{n+1}\right)$ is taken as a Gaussian mixture here. However, the WLS regression is technically only valid if the weights are known a-priori, and a rule of thumb for OLS regression is that it isn't too impacted by heteroscedasticity as long as the maximum variance is not greater than 4 times the minimum variance. As $\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}$ is not necessary for restoration of GCOMSM, here we chose simply the OLS to recover the $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$, $\forall j, k \in \boldsymbol{\Omega} = \{1, \ldots, K\}$, which writes as (3.21).

Minimization of (3.21) takes derivatives with respect to each parameter in

[8] $\left(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right)$ notes that $\mathbf{x}_{n+1}$ conditional on $\mathbf{x}_n$, $\mathbf{y}_n^{n+1}$, $\mathbf{r}_n^{n+1}$

$\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$. If their forms are linear combination of the parameters, for example, if we have $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = \boldsymbol{a}_{j,k} g_a\left(\mathbf{y}_n^{n+1}\right)$, $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = \boldsymbol{b}_{j,k} g_b\left(\mathbf{y}_n^{n+1}\right)$, where $g_a\left(\mathbf{y}_n^{n+1}\right)$, $g_b\left(\mathbf{y}_n^{n+1}\right)$ are known functions forms, the only two sets of parameters are $\boldsymbol{a}_{j,k}$ and $\boldsymbol{b}_{j,k}$, $j, k \in \boldsymbol{\Omega}$. The parameters are then acquired by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{\mathcal{L}}^{\intercal}\boldsymbol{\mathcal{L}}\right)^{-1} \boldsymbol{\mathcal{L}}^{\intercal}\mathbf{X}, \tag{3.22}$$

in which $\hat{\boldsymbol{\beta}} = \left[\hat{\boldsymbol{a}}_{1,1}\hat{\boldsymbol{b}}_{1,1}\cdots\hat{\boldsymbol{a}}_{1,K}\hat{\boldsymbol{b}}_{1,K}\cdots\hat{\boldsymbol{a}}_{K,K}\hat{\boldsymbol{b}}_{K,K}\ \right]^{\intercal}$ are stack of all parameters corresponding to switches, $\mathbf{X} = [\mathbf{x}_2\cdots\mathbf{x}_N]^{\intercal}$, and the matrix

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} p_{1,1}^1 g_{\mathbf{a},\mathbf{b}}^1 & \cdots & p_{1,K}^1 g_{\mathbf{a},\mathbf{b}}^1 & \cdots & p_{K,K}^1 g_{\mathbf{a},\mathbf{b}}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{1,1}^{N-1} g_{\mathbf{a},\mathbf{b}}^{N-1} & \cdots & p_{1,K}^{N-1} g_{\mathbf{a},\mathbf{b}}^{N-1} & \cdots & p_{K,K}^{N-1} g_{\mathbf{a},\mathbf{b}}^{N-1}, \end{bmatrix} \tag{3.23}$$

where $p_{j,k}^n = p\left(r_n = j, r_{n+1} = k\left|\mathbf{y}_1^N\right.\right)$ with $j, k \in \boldsymbol{\Omega}$, and $g_{\mathbf{a},\mathbf{b}}^n = \left[g_a\left(\mathbf{y}_n^{n+1}\right)\mathbf{x}_n \quad g_b\left(\mathbf{y}_n^{n+1}\right)\right]$.

When it comes to the case that $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$, $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_1^{n+1}\right)$ are non-linear on parameters, we can turn to various of numerical algorithms for minimizing the error, for example, the basic Gauss-Newton method with linear approximation of the functions, the Powell's Dog Leg method with a control of trust region, and some hybrid methods introduced in [16], [91], [16], [23]. In practice of the experiments illustrated in following Chapters, we tackle the non-linear least square problem with Levenberg–Marquardt (LM) algorithm which is a Damped Gauss-Newton method as proposed in [88] and completed in [94], [107] and [72].

### 3.3.3 The overall GICE-LS identification algorithm

Combining the GICE and LS methods for GCOMSM explained in previous Sections, Figure 3.7 gives the scheme of the entire GICE-LS identification strategy.

In this strategy, we assume that proper forms of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ are guessed. If not, similar to the choosing program in GICE, we can take several initial guess of the form of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$, applying LS on each of

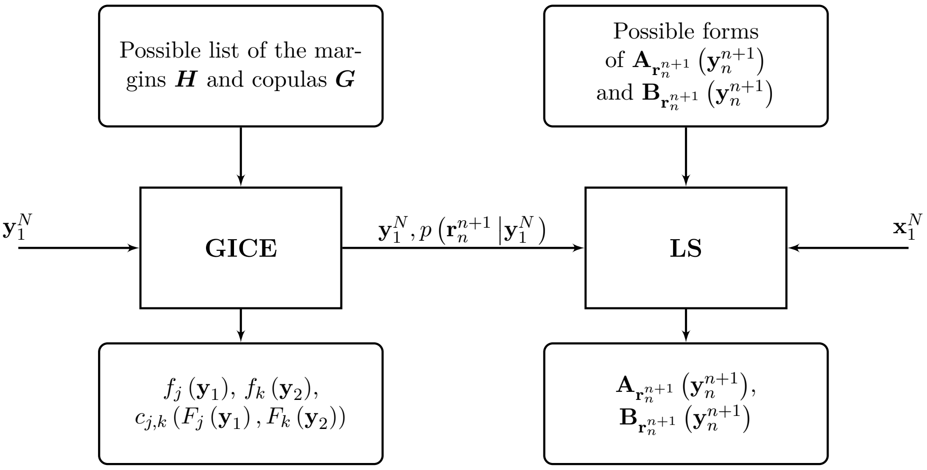

Possible list of the margins $\boldsymbol{H}$ and copulas $\boldsymbol{G}$
Possible forms of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$
$\mathbf{y}_1^N$
GICE
$\mathbf{y}_1^N, p\left(\mathbf{r}_n^{n+1}\left|\mathbf{y}_1^N\right.\right)$
LS
$\mathbf{x}_1^N$
$f_j\left(\mathbf{y}_1\right)$, $f_k\left(\mathbf{y}_2\right)$, $c_{j,k}\left(F_j\left(\mathbf{y}_1\right), F_k\left(\mathbf{y}_2\right)\right)$
$\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$, $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$

Figure 3.7: GICE-LS scheme.

them, then decide the best fit one by a decision rule $\mathcal{D}^3$, which choses the one who has minimum residual error of LS or the one who have minimal restoration MSE of sample set *etc.*

## 3.4 Performance and application of the GICE-LS identification algorithm

In this Section, we study how time independent GCOMSM identified by GICE-LS performs on non-Gaussian non-linear data.

We will firstly test the ability of the regimes and parameter recognition of GICE-LS on simulated GCOMSM data. Two cases are considered here. One Gaussian linear case which is the case degenerated to CGOMSM, and one general non-Gaussian non-linear case. For comparison, we display also the result of other two identification restoration algorithms. One replaces the GICE with ICE and assuming the distribution form of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ are all Gaussian, another one is the “CGOMSM Approximation Based Filter” (CGOMSM-ABF) [61], [62], which is an identification method for CGOMSM, takes entirely $p\left(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ as Gaussian. In addition, result of optimal restoration using the true parameters is given as a reference. Secondly, for further investigating the adaptability of the proposed GCOMSM, we apply the restoration of GCOMSM on other non-Gaussian non-linear generable models identified by GICE-LS, with comparison to the result of CGOMSM-ABF and the supervised particle filter (since no optimal filter exists for these non-Gaussian non-linear models) [9], [60], [75].

### 3.4.1 Performance on simulated GCOMSM data

Two series of Monte-Carlo experiments on simulated GCOMSM are provided here for verifying the ability of GICE-LS on the issue of the identification of time independent GCOMSM. All experiments based on simulation are under assumption that the Markov chain $\mathbf{R}_1^N$ has $K = 2$ states, and joint probabilities of switches $p_{j,k}$: $p_{1,1} = p_{2,2} = 0.45$, $p_{1,2} = p_{2,1} = 0.05$. Both hidden states and observations are assumed to be scalar. A set of 5000 simulated $\left(\mathbf{x}_1^N, \mathbf{y}_1^N\right)$ is taken as learning sample used for the model identification of both $p\left(\mathbf{y}_n^{n+1} \left| \mathbf{r}_n^{n+1}\right.\right)$ and parameter estimation of $\mathcal{G}\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$, assuming knowing the true regimes by GICE-LS. Another set of 1000 simulated data is taken for testing the restoration with the identified parameters. Meanwhile, replacing the GICE with the classic ICE which leads to the identification method ICE-LS is also applied on the same data set for comparison. This is of interest, because ICE (knowing all distributions are Gaussian) can be considered as a particular case of GICE which is sufficient for the identification of Gaussian $p\left(\mathbf{y}_n^{n+1} \left| \mathbf{r}_n^{n+1}\right.\right)$ (as in CGOMSM). The CGOMSM-ABF which will also be conducted for comparison is a newly proposed identification method based on EM for the CGOMSM. It is interesting to see what will happen when the CGOMSM-ABF is applied to the data which follows GCOMSM but no more its special Gaussian linear case.

#### 3.4.1.1 Gaussian linear case

In this series, $p\left(\mathbf{y}_n^{n+1} \left| \mathbf{r}_n^{n+1}\right.\right)$ is assumed to be Gaussian. The parameters of its Gaussian margins and copulas are set as

- Margins: $\theta_1 = \{loc_1 = 0.0, scale_1 = 1.5\}$, $\theta_2 = \{loc_2 = 1.0, scale_2 = 0.8\}$.

- Copulas: $\alpha_{1,1} = 0.8$, $\alpha_{1,2} = \alpha_{2,1} = 0.45$, $\alpha_{2,2} = 0.2$.

$loc_j$ and $cale_j$ denote the mean and standard deviation of the Gaussian margins respectively. $\alpha_{j,k}$ represents the single parameter of Gaussian copula, $j,k \in \{1,2\}$. Figure 3.8 shows the PDF of the margins and the joint distribution $p\left(\mathbf{y}_n, \mathbf{y}_{n+1} \left| r_n = 1, r_{n+1} = 2\right.\right)$.

The Gaussian distribution $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, r_n = j, r_{n+1} = k\right.\right) = \mathcal{N}\left(\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) + \mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right), \boldsymbol{\mathcal{V}}_{j,k}\right)$ is assumed with all linear forms conditionally on $\mathbf{x}_n$, $\mathbf{y}_n$, $\mathbf{y}_{n+1}$. Particularly, here we set $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = a_{j,k}$ and $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = b_{j,k}\mathbf{y}_n + c_{j,k}\mathbf{y}_{n+1} + d_{j,k}$. The parameters are assigned as:

- $a_{j,k}$: $a_{1,1} = 0.3$, $a_{1,2} = a_{2,1} = 0.5$, $a_{2,2} = 0.7$,

- $b_{j,k}$: $b_{1,1} = 0.61$, $b_{1,2} = 0.05$, $b_{2,1} = 0.25$, $b_{2,2} = -0.19$,

- $c_{j,k}$: $c_{1,1} = c_{2,1} = 0.30$, $c_{1,2} = c_{2,2} = 0.70$,

- $\boldsymbol{\mathcal{V}}_{j,k}$: $\boldsymbol{\mathcal{V}}_{1,1} = 0.91$, $\boldsymbol{\mathcal{V}}_{1,2} = \boldsymbol{\mathcal{V}}_{2,1} = 0.75$, $\boldsymbol{\mathcal{V}}_{2,2} = 0.51$,

and the constant $d_{j,k}$ is set to be 0 with $\forall j,k \in \{1,2\}$. Specially, for the identification with GICE, we assume that there are six candidate margin forms $\mathbf{H} = \{H_1, \cdots, H_6\}$, and seven candidate copula forms $\mathbf{G} = \{G_1, \cdots, G_7\}$ (all of them are one-parameter copula families with details in Table C.2 in Appendix C).

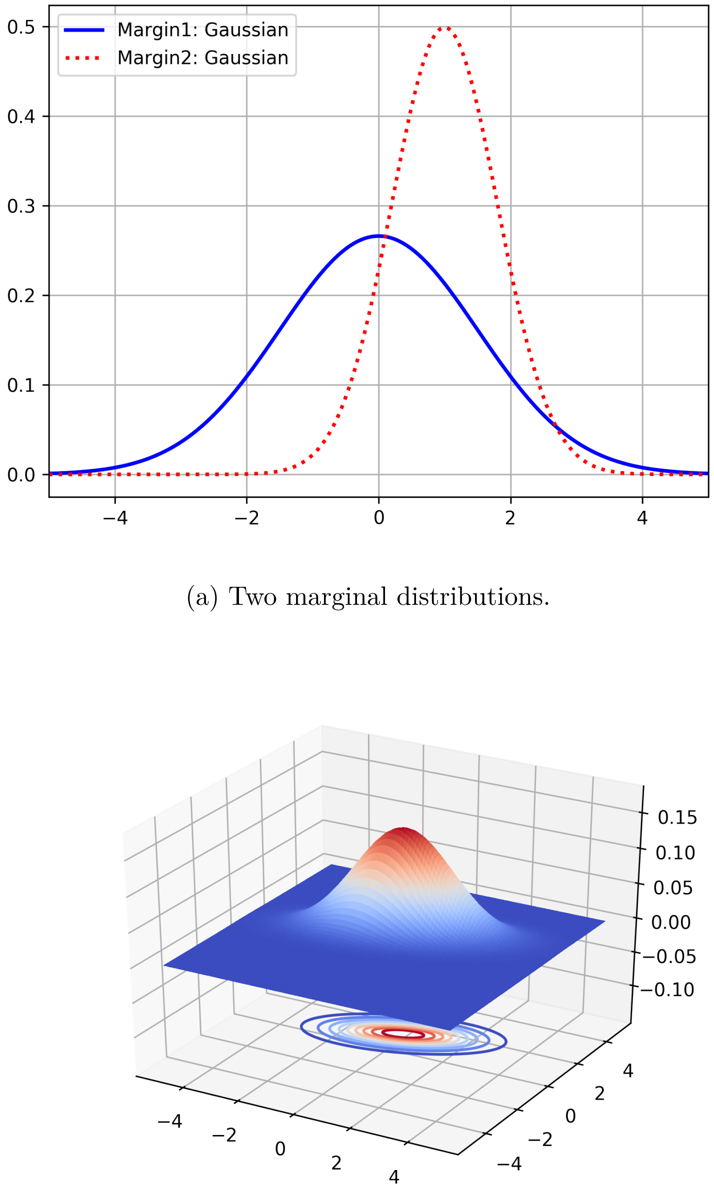
- $\{H_1, \cdots, H_6\}$: {Gamma, Fisk, Gaussian, Laplace, Beta, Beta prime}.

- $\{G_1, \cdots, G_7\}$: {Gumble, Gaussian, Clayton, FGM, Arch12, Arch14, Product}.

In each iteration of GICE, parameters of all the candidate margins and copulas are estimated by ML here. When estimating the parameters $\alpha_{j,k}$ of copulas, we use the semi-parametric method [76], [137], which calculates the ML with

$$\hat{\alpha}_{j,k} = \underset{\alpha_{j,k}}{\arg\max} \sum_{n=1}^{N} log\ c\left(\hat{F}_j\left(\mathbf{y}_n\right), \hat{F}_k\left(\mathbf{y}_{n+1}\right) \left| \hat{\alpha}_{j,k}\right.\right), \tag{3.24}$$

where $\left(\hat{F}_j\left(\mathbf{y}_n\right), \hat{F}_k\left(\mathbf{y}_{n+1}\right) \left| \hat{\alpha}_{j,k}\right.\right)$ is the empirical CDF of the pair $\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = j, r_{n+1} = k\right)$.



(a) Two marginal distributions.

(b) Joint distribution of $(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = 1, r_{n+1} = 2)$.

Figure 3.8: The distributions of series 1 (Gaussian linear).

In fact, we can have variate alternative methods for estimating the parameters. For margins, one could use also the moments method, while for copulas, a popular way is to estimate their Kendall's tau $\tau$ [74], which is equivalent to estimating $\alpha$ since they are linked by specific relations following individual copula forms. Moreover, one can also use parametric or non-parametric methods to replace the semi-parametric estimation in (3.24) [70].

The decision rules $\mathcal{D}^1$ for deciding the best fit marginal distribution we apply here, is the minimization of the "Kolmogorov Distance" denoted by "$d$" between the distribution specified by estimated parameters and the empirical distribution of $p\left(y_n \left| r_n = j\right.\right)$. It makes decision by computing

$$\mathcal{D}^1\left(\left(\mathbf{y}_1^N\right)_j\right) = \underset{F_l \in \{F_1, \cdots, F_L\}}{\arg\inf} \left[d\left(F_l, F_e\right)\right]. \tag{3.25}$$

Paying attention that $\left(\mathbf{y}_1^N\right)_j$ here refers to the data which is considered belonging to the same candidate distribution. For each iteration of GCOMSM, it is $\left(\mathbf{y}_1^N\right)_j^{i+1}$, the sub-sequence of $\mathbf{y}_1^N$ which corresponds to $r_n^{i+1} = j$. The empirical CDF $F_e\left(y\right) = \frac{1}{N}\sum_{n=1}^N 1_{[y_n<y]}$, and the related CDF are $F_1(y), \cdots, F_L(y)$. The Kolmogorov Distance $d$ between two CDFs is given by $d\left(F, F'\right) = sup\ _{y\in\mathbb{R}}\left|F\left(y\right) - F'\left(y\right)\right|$. As an alternative method, one can also use the "Bayesian Copula Selection" proposed in [68].

The decision rule $\mathcal{D}^2$ adopted for choosing the best fit copula is called "Pseudo-Likelihood Maximization" (PLM) [38] [76], whose decision is made by

$$\mathcal{D}^2\left(\left(\mathbf{y}_1^N\right)_{j,k}\right) = \underset{c_m \in \{c_1, \cdots, c_M\}}{\arg\sup} \prod_{n=2}^N c_m\left(F_{n-1}\left(y_{n-1}\right), F_n\left(y_n\right)\right), \tag{3.26}$$

in which $\left(\mathbf{y}_1^N\right)_{j,k} = \left(y_1, y_2\right)_{j,k}, \left(y_2, y_3\right)_{j,k}, \cdots, \left(y_{N-1}, y_N\right)_{j,k}$ refers to the data pairs which are considered belonging to the same candidate copulas. For each iteration of GCOMSM, it is $\left(\mathbf{y}_1^N\right)_{j,k}^{i+1}$, the sub-sequence of couples $\left(\mathbf{y}_n, \mathbf{y}_{n+1}\right)$ which corresponds to $\left(r_n^{i+1}, r_{n+1}^{i+1}\right) = (j,k)$. The two associated marginal CDF, $F_{n-1}\left(y_{n-1}\right)$ and $F_n\left(y_n\right)$ are already decided by $\mathcal{D}^1$.

In this experiment, the linear form of $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$, $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ are assumed known. Applying the three identification methods GICE-LS, ICE-LS, CGOMSM-ABF on the sample data set, 100 iterations are set for all GICE, ICE and CGOMSM-ABF to converge. Then we use the learned parameters to restore the testing set, the result are reported in Table 3.3.

Table 3.3: Restoration results of series 1 (Gaussian linear).

| MSE of observation: 1.726 | | Optimal | GICE-LS | ICE-LS | CGOMSM-ABF |
|---|---|---|---|---|---|
| Filtering | Error ratio | 0.245 | 0.289 | 0.249 | 0.247 |
| | MSE | 1.037 | 1.047 | 1.044 | 1.044 |
| Smoothing | Error ratio | 0.211 | 0.261 | 0.215 | 0.213 |
| | MSE | 1.032 | 1.044 | 1.039 | 1.040 |

From Table 3.3, we see that GCOMSM still works on Gaussian case as CGOMSM does. Under the worst condition that both distribution forms and parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{y}_n^{n+1}\right.\right)$ are unclear, the filtering and smoothing with the parameters identified through GICE-LS gets competitive result not far from the filtering and smoothing result identified thorough ICE-LS and CGOMSM-ABF which assumes knowing all the shapes to be Gaussian. However, it still needs to notice that GICE not always find Gaussian as the "best fitted" distribution, it may sometimes converge to the others with similar PDF, the identification result of the form of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ on the learning sample set is listed in Table 3.4 and Table 3.5.

Table 3.4: Margin selection result of GICE in series 1.

| Form | Gamma | Fisk | Gaussian | Laplace | Beta | Beta prime |
|---|---|---|---|---|---|---|
| $f_1$ | 2% | 1% | 86% | 0% | 0% | 0% |
| $f_2$ | 5% | 3% | 54% | 1% | 0% | 37% |

Table 3.5: Copula selection result of GICE in series 1.

| Form | Gumbel | Gaussian | Clayton | FGM | Arch12 | Arch14 | Product |
|---|---|---|---|---|---|---|---|
| $c_{1,1}$ | 1% | 43% | 2% | 0% | 3% | 51% | 0% |
| $c_{1,2} = c_{2,1}$ | 32% | 52% | 10% | 4% | 0% | 2% | 0% |
| $c_{2,2}$ | 14% | 60% | 4% | 19% | 0% | 3% | 0% |

Actually, this fact affects not too much in the final restoration, since with specific parameter settings, different distributions can have very similar PDF. The same phenomenon will also be reported in next experimental series with details. Table

3.6 shows the average of estimated parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ from the instances that GICE converges to the true Gaussian forms. The estimated parameters related to $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}\mathbf{r}_n^{n+1}\right.\right)$ are reported in Table 3.7. The estimated parameters are all not far away from the true ones. Estimated switching joint probabilities from GICE are $p_{1,1} = 0.485$, $p_{1,2} = p_{2,1} = 0.047$, $p_{2,2} = 0.421$; while from ICE are $p_{1,1} = 0.489$, $p_{1,2} = p_{2,1} = 0.046$, $p_{2,2} = 0.419$. In addition, the identification tested is quite efficient in this series, since all the three methods get similar results as the supervised optimal one. ICE-LS could be taken as an alternative method to CGOMSM-ABF, as they both work on Gaussian case, and get very close results. An example of trajectories (smoothing) is given in Figure 3.9 which shows intuitively very similar performance of all the three identification methods and also the optimal restoration.

Table 3.6: Estimated parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ in series 1.

| | Margins | | | | Copulas | | |
|---|---|---|---|---|---|---|---|
| | $f_1$ (Gaussian) | | $f_2$ (Gaussian) | | $c_{1,1}$<br>(Gaussian) | $c_{1,2}/c_{2,1}$<br>(Gaussian) | $c_{2,2}$<br>(Gaussian) |
| | $loc_1$ | $scale_1$ | $loc_2$ | $scale_2$ | $\alpha_{1,1}$ | $\alpha_{1,2}/\alpha_{2,1}$ | $\alpha_{2,2}$ |
| Estimates | -0.04 | 1.47 | 0.99 | 0.82 | 0.78 | 0.49 | 0.23 |
| True values | 0.00 | 1.50 | 1.00 | 0.80 | 0.80 | 0.45 | 0.20 |

Table 3.7: Estimated parameters of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ in series 1.

| | Estimates | | | | True values | | | |
|---|---|---|---|---|---|---|---|---|
| $(j,k)$ | (1,1) | (1,2) | (2,1) | (2,2) | (1,1) | (1,2) | (2,1) | (2,2) |
| $a_{j,k}$ | 0.34 | 0.56 | 0.47 | 0.67 | 0.30 | 0.50 | 0.50 | 0.70 |
| $b_{j,k}$ | 0.50 | 0.05 | 0.20 | -0.11 | 0.61 | 0.05 | 0.25 | -0.19 |
| $c_{j,k}$ | 0.39 | 0.78 | 0.27 | 0.64 | 0.30 | 0.70 | 0.30 | 0.70 |
| $d_{j,k}$ | 0.01 | 0.07 | 0.01 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 |

#### 3.4.1.2 Non-Gaussian non-linear case

As last series shows the efficiency of all the three identification method on Gaussian linear case of GCOMSM, this series is designed to test their performance on the general non-Gaussian non-linear case of GCOMSM.

The parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ which are supposed to be non-Gaussian in

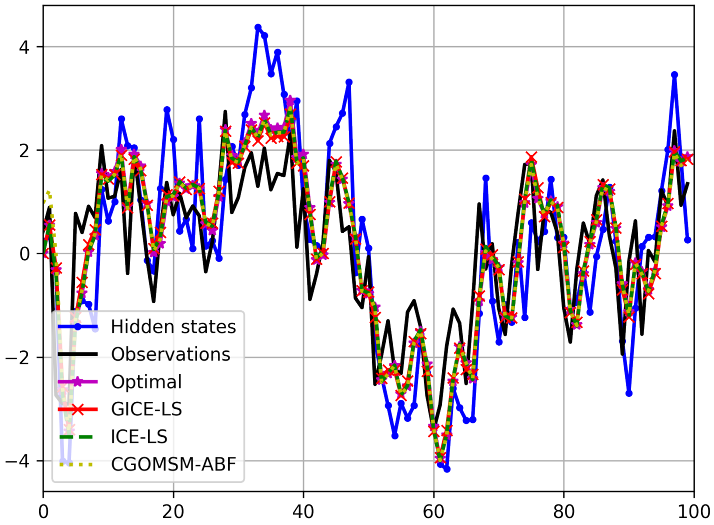


Figure 3.9: Trajectory example in series 1 (100 samples, smoothing).

this experiment are set as:

- Margins: $f_1(\mathbf{y}_n) = \text{Gamma}\{\theta_1 = 16, loc_1 = -5, scale_1 = 0.25\}$,
  $f_2(\mathbf{y}_n) = \text{Fisk}\{\theta_2 = 4, loc_2 = -2.67, scale_2 = 2.4\}$.

- Copulas: $c_{1,1}\{\cdot,\cdot\} = \text{Gumbel}\{\cdot,\cdot|\alpha_{1,1} = 1.1\}$,
  $c_{2,2}\{\cdot,\cdot\} = \text{Clayton}\{\cdot,\cdot|\alpha_{2,2} = 4.67\}$,
  $c_{1,2}\{\cdot,\cdot\} = c_{2,1}\{\cdot,\cdot\} = \text{Gaussian}\{\cdot,\cdot|\alpha_{1,2} = 0.45\}$.

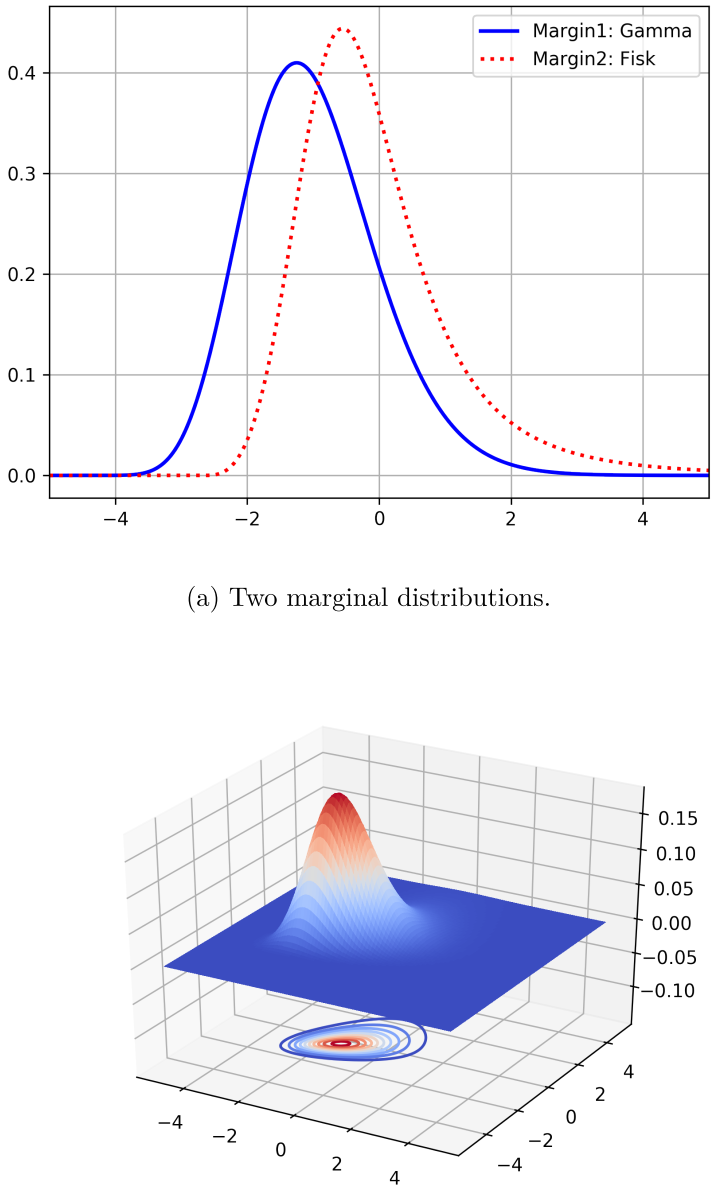
See the detail of the parameterization of all margins and copulas in Table C.1 and Table C.2 in Appendix C. $p\left(\mathbf{x}_{n+1}\middle|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right) = \mathcal{N}\left(a_{j,k}\mathbf{x}_n + \mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right), \boldsymbol{\mathcal{V}}_{j,k}\right)$ is set in this series, which completes the relation in $p\left(\mathbf{t}_{n+1}|\mathbf{t}_n\right)$. $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = b_{j,k}\mathbf{y}_n\mathbf{y}_{n+1} + d_{j,k}$ is defined non-linear on $\mathbf{y}_n$, $\mathbf{y}_{n+1}$, with simply one parameter $b_{j,k}$ and all the parameters are assigned as:

- $a_{j,k}$: $a_{1,1} = 0.2$, $a_{1,2} = 0.4$, $a_{2,1} = 0.6$, $a_{2,2} = 0.8$,

- $b_{j,k}$: $b_{1,1} = 0.7$, $b_{1,2} = 0.5$, $b_{2,1} = 0.6$, $b_{2,2} = 0.9$,

- $\boldsymbol{\mathcal{V}}_{j,k}$: $\boldsymbol{\mathcal{V}}_{1,1} = \boldsymbol{\mathcal{V}}_{2,2} = 1.0$, $\boldsymbol{\mathcal{V}}_{1,2} = \boldsymbol{\mathcal{V}}_{2,1} = 0.8$.

The constants $d_{j,k}$ with $\forall j, k \in \{1, 2\}$ are all set to be zero. The two margins and a joint distribution, which has Gamma and Fisk as marginal distributions and Gaussian as copula are displayed in Figure 3.10.



(a) Two marginal distributions.

(b) Joint distribution of $\left(\mathbf{y}_n, \mathbf{y}_{n+1} | r_n = 1, r_{n+1} = 2\right)$.

Figure 3.10: The distributions of series 2 (non-Gaussian non-linear).

The same settings (iteration, candidate forms) for GICE-LS, ICE-LS and CGOMSM-ABF in series 1 are adopted in this series, restoration results are reported in Table 3.8.

Table 3.8: Restoration results of series 2 (non-Gaussian non-linear).

| MSE of observation: 27.123 | | Optimal | GICE-LS | ICE-LS | CGOMSM-ABF |
|---|---|---|---|---|---|
| Filtering | Error ratio | 0.139 | 0.156 | 0.404 | 0.462 |
| | MSE | 2.380 | 2.771 | 5.762 | 9.353 |
| Smoothing | Error ratio | 0.084 | 0.103 | 0.378 | 0.456 |
| | MSE | 2.290 | 2.631 | 5.750 | 9.273 |

Clearly from Table 3.8, when the case comes to non-Gaussian, non-linear, GICE-LS is the most suitable method for identifying the GCOMSM comparing to the other two methods. CGOMSM-ABF performs the worst, since it assumes Gaussian in both $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ and $p\left(\mathbf{x}_n^{n+1}\left|\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ (the regime of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n,\mathbf{y}_n^{n+1},\mathbf{r}_n^{n+1}\right.\right)$ is linear on $\mathbf{y}_n^{n+1}$), while ICE-LS can better take into account the consideration of the non-linear form of $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$. The very difference between CGOMSM-ABF and the other LS based methods is that CGOMSM-ABF takes $\left(\mathbf{R}_1^N,\mathbf{XY}_1^N\right)$ as a PMC where $\mathbf{XY}_n = [\mathbf{X}_n^\intercal\ \mathbf{Y}_n^\intercal]^\intercal$ and $\mathbf{XY}_1^N = \{\mathbf{XY}_1,\cdots,\mathbf{XY}_N\}$. This assumption can be very sensible and practical in real applications. But under GCOMSM, the case is $\left(\mathbf{R}_1^N,\mathbf{Y}_1^N\right)$ is a PMC while $\left(\mathbf{R}_1^N,\mathbf{XY}_1^N\right)$ can be not a PMC (for example under the setting in this experimental series). This explains why ICE-LS gets better error ratio of estimated $\mathbf{r}_1^N$ than CGOMSM-ABF.

Inevitably, like briefly illustrated in the Gaussian linear series, the automatic selection process of GICE sometimes may choose other distributions but not the optimal one. As listed in Table 3.9, the selection percentage of margins, and Table 3.10, the selection percentage of copulas in this series, Fisk is selected as the optimal shape with 8% rate which originally should be Gamma distribution. But according to the result, GICE still converges when the "wrong" shape is selected, since with specific parameter estimated, the "wrong" shape may also fits the data well. This situation exists in the copula selection too. An instance of this similarity is reported in Figure 3.11 from one "wrong" estimated form case in the 100 Monte-Carlo experiments, it has a very close PDF shape compared to true one illustrated in Figure

3.10b. Nevertheless, the original margins and copulas are selected by GICE most frequently in this series.

Table 3.9: Margin selection result of GICE in series 2.

| Form | Gamma | Fisk | Gaussian | Laplace | Beta | Beta prime |
|---|---|---|---|---|---|---|
| $f_1$ | 87% | 12% | 0% | 1% | 0% | 0% |
| $f_2$ | 1% | 99% | 0% | 0% | 0% | 0% |

Table 3.10: Copula selection result of GICE in series 2.

| Form | Gumbel | Gaussian | Clayton | FGM | Arch12 | Arch14 | Product |
|---|---|---|---|---|---|---|---|
| $c_{1,1}$ | 96% | 2% | 1% | 0% | 0% | 1% | 0% |
| $c_{1,2} = c_{2,1}$ | 34% | 58% | 4% | 0% | 0% | 0% | 0% |
| $c_{2,2}$ | 2% | 0% | 96% | 1% | 1% | 0% | 0% |

An example of error ratio tendencies with GICE and ICE iterations of estimated $\mathbf{R}_1^N$ (using MPM criterion) in identification set is given in Figure 3.12. It shows that GICE and ICE both converge after around 40 iterations, but ICE is not able to well approximate $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$.

The parameters estimated by GICE-LS are quite near to the true ones as listed in Table 3.11 (average of the instances where the forms of the distribution are exactly estimated) and Table 3.12. Estimated switch joint probabilities from GICE are $p_{1,1} = 0.474$, $p_{1,2} = p_{2,1} = 0.040$, $p_{2,2} = 0.445$.

Table 3.11: Estimated parameters of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ in series 2.

| | Margins | | | | | | Copulas | | |
|---|---|---|---|---|---|---|---|---|---|
| | $f_1$ (Gamma) | | | $f_2$ (Fisk) | | | $c_{1,1}$<br>(Gumbel) | $c_{1,2}/c_{2,1}$<br>(Gaussian) | $c_{2,2}$<br>(Clayton) |
| | $\theta_1$ | $loc_1$ | $scale_1$ | $\theta_2$ | $loc_2$ | $scale_2$ | $\alpha_{1,1}$ | $\alpha_{1,2}/\alpha_{2,1}$ | $\alpha_{2,2}$ |
| Estimates | 13.72 | -4.75 | 0.29 | 3.93 | -2.60 | 2.30 | 1.15 | 0.46 | 4.46 |
| True value | 16.00 | -5.00 | 0.25 | 4.00 | -2.67 | 2.40 | 1.10 | 0.45 | 4.67 |

Table 3.12: Estimated parameters of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ in series 2.

| | Estimates | | | | True values | | | |
|---|---|---|---|---|---|---|---|---|
| $(j, k)$ | (1,1) | (1,2) | (2,1) | (2,2) | (1,1) | (1,2) | (2,1) | (2,2) |
| $a_{j,k}$ | 0.27 | 0.41 | 0.69 | 0.81 | 0.20 | 0.40 | 0.60 | 0.80 |
| $b_{j,k}$ | 0.69 | 0.56 | 0.63 | 0.90 | 0.70 | 0.50 | 0.60 | 0.90 |
| $d_{j,k}$ | 0.00 | -0.01 | -0.13 | -0.01 | 0.00 | 0.00 | 0.00 | 0.00 |

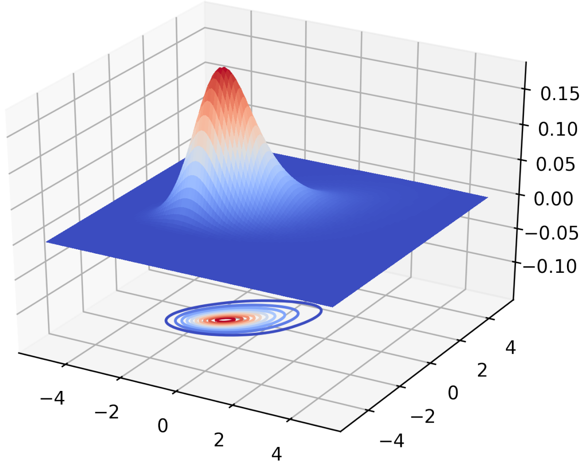


Figure 3.11: “Wrong” estimated joint distribution with (Margins: Fisk, Fisk; Copula: Gumbel).

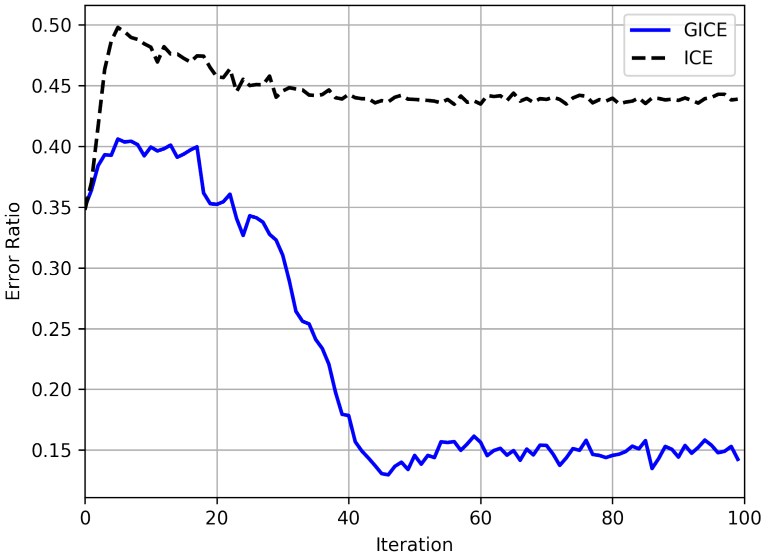


Figure 3.12: Error ratio tendency of estimated $\mathbf{R}_1^N$ with GICE and ICE iterations within same individual experiment in series 2.

Finally, we display a trajectory example of $\left(\mathbf{x}_1^N, \mathbf{y}_1^N\right)$, with all estimated hidden states restored by the parameters identified through all the three identification methods in Figure 3.13. From this Figure, the superiority of GICE-LS over the other methods on general GCOMSM data is clearly visible.

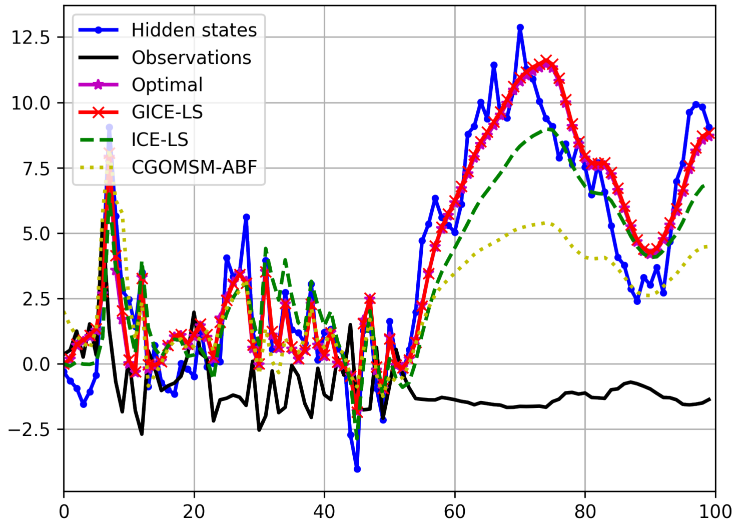


Figure 3.13: Trajectory example in series 2 (100 samples, smoothing).

## 3.4.2 Application of GICE-LS to non-Gaussian non-linear models

The efficiency of GICE-LS on identification of GCOMSM has been proved in previews Sections. We would like to see how it performs on other non-Gaussian non-linear data. This application means to approximate a non-Gaussian non-linear system by GCOMSM, with parameter identified through GICE-LS, and also restored by the optimal restoration method of the approximated GCOMSM.

### 3.4.2.1 On stochastic volatility data

Stochastic volatility model is a family of models used in the field of mathematical finance. It models the volatility as a stochastic process and is widely used as an approach to solve the shortcoming of the Black–Scholes model, in which the underlying volatility is always constant and unaffected by the changes, and it explains the "volatility smile" in a self-consistent way that the volatility has its realistic dynamics [57], [58]. There are stochastic volatility models which formulate the dynamic volatility in different ways, for example Heston model [66], CEV model [48], GARCH model [22] *etc.*

The article which proposes the CGOMSM-ABF [62] shows that switching Gaussian Markov model can well approximate the non-linear non-Gaussian stochastic volatility models. The associated optimal restoration to the approximated CGOMSM for the stochastic volatility models can reach a very close performance as Particle Filter [44], [30] on same stochastic volatility model but with much less time consumption.

In this Section, we apply the identification method GICE-LS to approach the stochastic volatility with the general GCOMSM and see the performance comparing to CGOMSM-ABF of CGOMSM and the Particle filter. Two stochastic volatility models are considered in this experimental series, one is the standard stochastic volatility (SV) model [78], [69], [131], which is defined as

$$\begin{aligned} \mathbf{X}_1 &= \mu + \mathbf{U}_1 \\ \mathbf{X}_{n+1} &= \mu + \phi\left(\mathbf{X}_n - \mu\right) + \sigma \mathbf{U}_{n+1}, \\ \mathbf{Y}_n &= \beta\ exp\left(\frac{\mathbf{X}_n}{2}\right)\mathbf{V}_n \end{aligned} \tag{3.27}$$

in which, the hidden state $\mathbf{X}_1^N$ is normally taken as log-volatility and observations $\mathbf{Y}_1^N$ is the so called mean corrected return. $\mathbf{U}_1^N$, $\mathbf{V}_1^N$ are independent standard normal white noises. $\mu$. $\phi$, $\sigma$ represent the mean, persistence, and the volatility of this hidden log-volatility process. The parameter $\beta$ is the constant scaling factor. A second stochastic volatility model which is extended from the canonical one, is the asymmetric stochastic volatility (ASV) [65], [105], [106], [130], defined as

$$\begin{aligned} \mathbf{X}_1 &= \mu + \mathbf{U}_1 \\ \mathbf{X}_{n+1} &= \mu + \phi\left(\mathbf{X}_n - \mu\right) + \sigma\left(\frac{\rho \mathbf{Y}_n}{\beta\ exp\left(\frac{\mathbf{X}_n}{2}\right)}\lambda \mathbf{U}_{n+1}\right). \\ \mathbf{Y}_n &= \beta\ exp\left(\frac{\mathbf{X}_n}{2}\right)\mathbf{V}_n \end{aligned} \tag{3.28}$$

These two stochastic volatility models are both generable HMM models.

***SV model***

The parameters in SV model in this experiment is set as $\mu = \beta = \phi = 0.5$, and $\sigma$ is got by $\sqrt{1-\phi^2}$ to ensure the stationarity (both mean and variance are stationary) of $\mathbf{X}_1^N$. We test the CGOMSM-ABF, ICE-LS, GICE-LS and the Particle Filter on the same observations generated from this SV model. When carrying out the three identification methods, we try different state numbers $K$ of the switches. The size of learning sample set for identification is 20000, while the testing data set is of 1000 samples. Regarding especially GICE which assumes the distributions of $\left(\mathbf{y}_n^{n+1} | \mathbf{r}_n^{n+1}\right)$ is unknown, we prepare six candidate margin shapes $\{H_1, \cdots, H_6\}$ and seven candidate copula shapes $\{G_1, \cdots, G_7\}$ as in previous experiments. All candidate forms are listed bellow.

- $\{H_1, \cdots, H_6\}$: {Gamma, Fisk, Gaussian, Laplace, Beta, Beta prime}.
- $\{G_1, \cdots, G_7\}$: {Gumble, Gaussian, Clayton, FGM, Arch12, Arch14, Product}.

The regime $\mathcal{G}\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, r_n = j, r_{n+1} = k\right.\right)$ where $j, k \in \mathbf{\Omega}$ assumed for the approximated GCOMSM is with the form $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)\mathbf{x}_n + \mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$, which has both $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right)$ linear forms conditionally on $\mathbf{x}_n$, $\mathbf{y}_n$, $\mathbf{y}_{n+1}$ for GICE. Which means that $\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = a_{j,k}$ and $\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = b_{j,k}\mathbf{y}_n + c_{j,k}\mathbf{y}_{n+1} + d_{j,k}$ with $a_{j,k}$, $b_{j,k}$, $c_{j,k}$, $d_{j,k}$ the parameters needed to be estimated. 100 iterations is set for EM in CGOMSM-ABF, ICE and GICE.

As there is no exact filtering for SV model, the result of Particle Filter can be a reference to see if the switching models fit for the SV model or not. 1500 particles[9] are used for Particle Filter in the result reported in this experiment since empirically we found tiny difference between the performances of Particle Filter with more particles. The MSE results of all the methods are reported in Table 3.13.

***Asymmetric SV model***

[9] PF behaves asymptotically for this particle number in this experimental series.

Table 3.13: MSE results of four methods on SV model (PF represents the Particle Filter).

| | $K$ | 2 | 3 | 4 | 5 | PF |
|---|---|---|---|---|---|---|
| Filtering | CGOMSM-ABF | 0.71 | 0.69 | 0.70 | 0.70 | 0.70 |
| | ICE-LS | 0.73 | 0.70 | 0.70 | 0.70 | |
| | GICE-LS | 0.79 | 0.70 | 0.70 | 0.69 | |
| Smoothing | CGOMSM-ABF | 0.69 | 0.67 | 0.67 | 0.67 | 0.67 |
| | ICE-LS | 0.71 | 0.68 | 0.67 | 0.67 | |
| | GICE-LS | 0.79 | 0.69 | 0.69 | 0.69 | |

We take the same parameter setting of $\mu$, $\beta$, $\phi$, $\sigma$ as SV for ASV model, the extra parameters are assigned by $\rho = -0.5$ and $\lambda = \sqrt{1-\rho^2}$ (to ensure the stationarity of $\mathbf{X}_1^N$). All the conditions set for identification, restoration, and sample size are the same as the experiment on SV model. Results of the four methods applied on the simulated data following this ASV are reported in Table 3.14.

Table 3.14: MSE results of four methods on ASV model.

| | $K$ | 2 | 3 | 4 | 5 | 7 | PF |
|---|---|---|---|---|---|---|---|
| Filtering | CGOMSM-ABF | 0.60 | 0.59 | 0.58 | 0.58 | 0.58 | 0.57 |
| | ICE-LS | 0.60 | 0.61 | 0.60 | 0.58 | 0.58 | |
| | GICE-LS | 0.66 | 0.59 | 0.59 | 0.59 | 0.58 | |
| Smoothing | CGOMSM-ABF | 0.57 | 0.56 | 0.54 | 0.54 | 0.54 | 0.54 |
| | ICE-LS | 0.58 | 0.59 | 0.58 | 0.56 | 0.55 | |
| | GICE-LS | 0.66 | 0.58 | 0.58 | 0.57 | 0.56 | |

From the result of these two experiments on SV and ASV models, we can see that switching Markov models works well on approximating the stationary stochastic volatility models. Their exact filtering or smoothing results are quite close to Particle Filter. Still, we see the differences of the performance between different identification methods. Before explaining their performance, let us recall simultaneously the characteristics of the three identification methods. CGOMSM-ABF is specified for CGOMSM which is a Gaussian linear GCOMSM, ICE-LS is for Gaussian GCOMSM which can be non-linear, and GICE-LS is for GCOMSM which can be non-Gaussian non-linear. No matter what the value $K$ is, the results in Table 3.13 and 3.14 show that CGOMSM-ABF is always the most efficient identification method on these two stochastic volatility models. It indicates that Gaussian mixture is a very suitable approximation of $p\left(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1}\right)$ for SV models under the settings in this experiment. As $\mathcal{G}\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right)$ is set to be linear on $\mathbf{x}_n$, both CGOMSM-ABF and ICE-LS identification methods serve for CGOMSM. But partial general consideration of dependence that $\left(\mathbf{R}_1^N, \mathbf{XY}_1^N\right)$ in the identification procedure may lead to the better performance of CGOMSM-ABF comparing to ICE-LS. Regarding the identification performance between ICE-LS and GICE-LS, when the Gaussian linear case fits well for the true system, ICE-LS works better than GICE-LS which lacks knowledge of the shape of $p\left(\mathbf{y}_n^{n+1} \left| \mathbf{r}_n^{n+1}\right.\right)$. Implemented by C programming language, the smoothing for CGOMSM in CGOMSM-ABF takes 0.0038 seconds on a 3.7GHz CPU, while Particle Filter takes 0.56s seconds [10]. The smoothing for GCOMSM costs around 0.40 seconds implemented by python 3.6. Though different programming language based implementations make the time consumption incomparable at present, both as exact restoration, the smoothing for GCOMSM should consume time not far (could be a little more due to the calculation of copulas) from CGOMSM-ABF if implemented in the same programming language. To conclude, we sum up some interesting points from these two experimental series that

1. Switching Markov model can be a good approach for stationary stochastic volatility models, the advantage of this approach is that exact restoration can be derived which is normally less time consuming than MCMC based restoration methods.

2. Working on CGOMSM, CGOMSM-ABF and ICE-LS are alternative identification methods of each other, but since CGOMSM-ABF has more general assumption in its partial process (which is actually of the property of CGPMSM), in practice, it may work better than ICE-LS if $\mathcal{G}\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ is linear on $\mathbf{y}_n^{n+1}$.

3. With less knowledge, GICE-LS could be the least efficient method when it comes to the case that linear Gaussian well fits distributions. But it still gets the result not too far away from the other identification methods.

[10] The original program is provided by the author of [62]

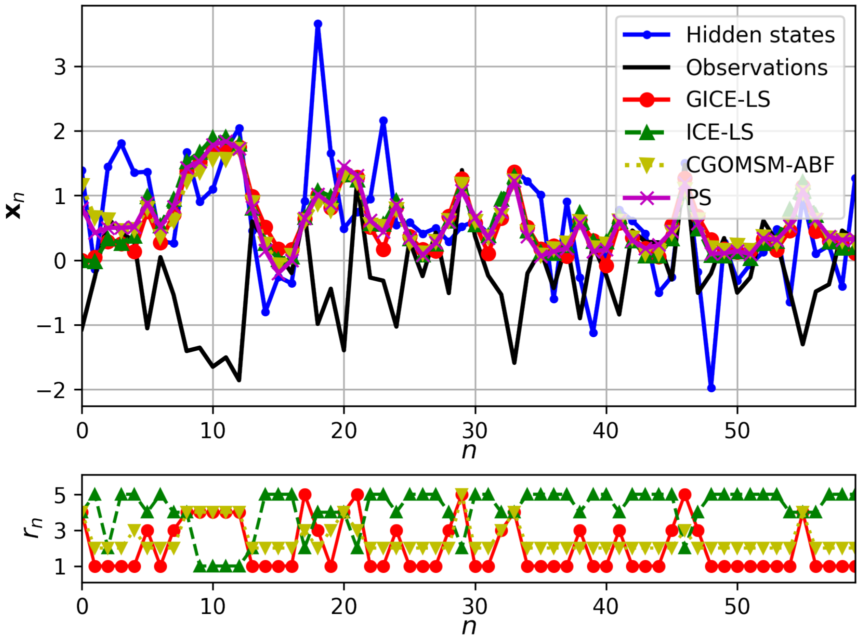


Figure 3.14: Trajectory example of SV model (60 samples, $K$=5).

Two trajectory instances corresponding to these two SV models are displayed in Figure 3.14 and 3.15. In the experiment of Figure 3.14, for $K = 5$ different margins, GICE-LS chose { Gaussian, Beta prime, Gaussian, Laplace, Beta prime }. In Figure 3.15, the $K = 7$ different chosen margins by GICE-LS are { Beta prime, Gamma, Laplace, Gaussian, Gamma, Gaussian, Gaussian, }. The chosen copulas are too many to list, but they are also not all Gaussian. We see in both figures that the assistant artificial switches estimated from CGOMSM-ABF, ICE-LS and GICE-LS can be still very close to each other[11] though the estimated distributions of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ are quite different between GICE-LS and the other two identification methods.

[11] The classes of switches are randomly distributed from K-means, so most of the time, the class labels of two individual experiments are different.

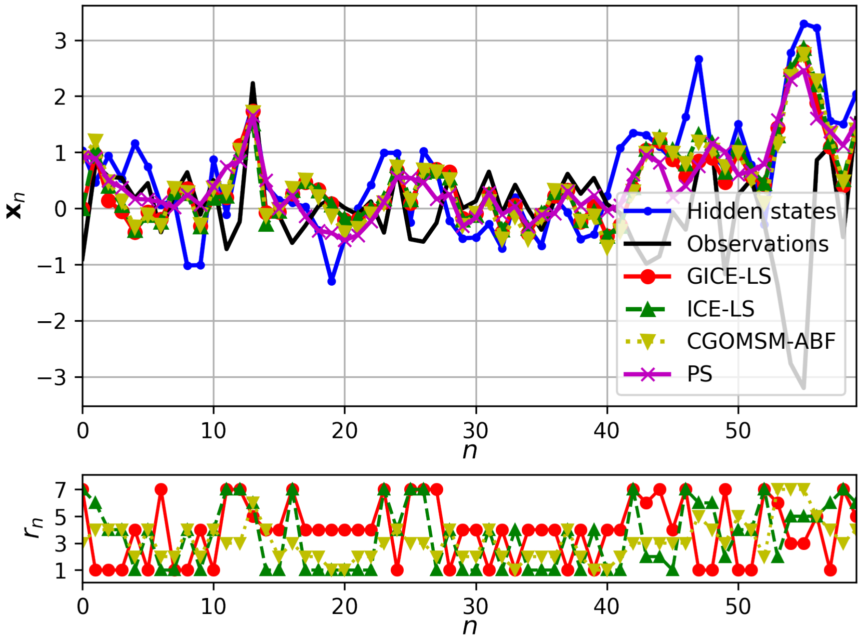


Figure 3.15: Trajectory example of ASV model (60 samples, $K$=7).

#### 3.4.2.2 On Kitagawa data

To better understand these methods and their properties, we also test all of the methods on the non-Gaussian non-linear model originally used in [103], and has been reconsidered by [80] and [79] for testing the performance of MCMC based filter. Here we call this model "Kitagawa model" (KTGW). In addition, we test the methods on the transformed semi-linear case of KTGW, called "Kitagawa semi-linear model" (KTGWSL) later which has been studied in [39] as supplementary.

KTGW model is defined as

$$\begin{aligned} \mathbf{X}_{n+1} &= 0.5\mathbf{X}_n + \frac{25\mathbf{X}_n}{1+\mathbf{X}_n^2} + 8\cos\left(1.2n+1\right) + \mathbf{V}_{n+1} \\ \mathbf{Y}_{n+1} &= \frac{\mathbf{X}_{n+1}^2}{20} + \mathbf{U}_{n+1} \end{aligned}, \tag{3.29}$$

where $\mathbf{V}_{n+1}$ and $\mathbf{U}_{n+1}$ are Gaussian white noise sequences, and the KTGWSL model is defined just with a change of non-linear measurement to a linear one.

$$\begin{aligned} \mathbf{X}_{n+1} &= 0.5\mathbf{X}_n + \frac{25\mathbf{X}_n}{1+\mathbf{X}_n^2} + 8\cos\left(1.2n+1\right) + \mathbf{V}_{n+1} \\ \mathbf{Y}_{n+1} &= 0.5\mathbf{X}_{n+1} + \mathbf{U}_{n+1} \end{aligned} . \tag{3.30}$$

They are both non-stationary models.

For the experimental series on Kitagawa models, learning sample set for identification is of 20000 points and testing sample set is of 1000 samples. All settings of the identification methods and Particle Filter are the same as previous experimental series on stochastic volatility models if no specification declared.

***KTGW model***

We assign the parameters of KTGW with $\mathbf{X}_1 \sim \mathcal{N}\{0,1\}$, $\mathbf{V}_{n+1} \sim \mathcal{N}\{0,0.5\}$ and $\mathbf{U}_{n+1} \sim \mathcal{N}\{0,2\}$. Regarding the GCOMSM approximation here (in the identification of GICE-LS and ICE-LS), we consider a non-linear form on $\mathbf{y}_n$, $\mathbf{y}_{n+1}$ which is a bit similar to the regime of KTGW that

$$\begin{aligned} \mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) &= a_{j,k}\mathbf{y}_n + b_{j,k}; \\ \mathbf{B}_{j,k,n}{}^{12}\left(\mathbf{y}_n^{n+1}\right) &= c_{j,k}\sqrt{\left|\mathbf{y}_{n+1}\right|} + d_{j,k}\cos\left(1.2\left(n+1\right)+e_{j,k}\right) + f_{j,k}. \end{aligned} \tag{3.31}$$

The exact restoration results of all applied identification methods are reported in Table 3.15 with the restorations of Particle Filter in the rightmost column of the Table. We see that the Markov switching model is less efficient for approaching KTGW model than SV models, as KTGW is non-stationary. Moreover, the regu-

Table 3.15: MSE results of four methods on KTGW model.

| | $K$ | 2 | 3 | 4 | 5 | 7 | PF |
|---|---|---|---|---|---|---|---|
| Filtering | CGOMSM-ABF | 105.43 | 98.96 | 99.23 | 98.98 | 99.31 | 8.43 |
| | ICE-LS | 41.04 | 25.96 | 22.72 | 20.09 | 19.84 | |
| | GICE-LS | 39.65 | 34.96 | 22.45 | 19.77 | 19.25 | |
| Smoothing | CGOMSM-ABF | 110.87 | 98.06 | 99.92 | 99.22 | 100.31 | 0.83 |
| | ICE-LS | 41.19 | 22.97 | 4.55 | 7.54 | 6.58 | |
| | GICE-LS | 39.95 | 34.75 | 4.59 | 7.81 | 7.55 | |

[12] Here, $B_{j,k,n}$ is not time independent, but the parameters which need to be estimated are still time-independent.

larity that larger $K$ set gets better restoration is no more held under non-stationary model. Actually, from Table 3.15, we find better result of both ICE-LS and GICE-LS when $K = 4$ than $K = 5$ or $K = 7$.

Assuming both stationary and linear, CGOMSM-ABF turns out to be noneffective for KTGW model, while ICE-LS and GICE-LS can still work for restoration although can not be as efficient as the supervised Particle Filter. This implies the significance of the generalization from CGOMSM to GCOMSM, especially the extension of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ which becomes much more flexible in the proposed GCOMSM. A trajectory example is illustrated in Figure 3.16, which shows the performances of all methods comparing to the true hidden states and observations.

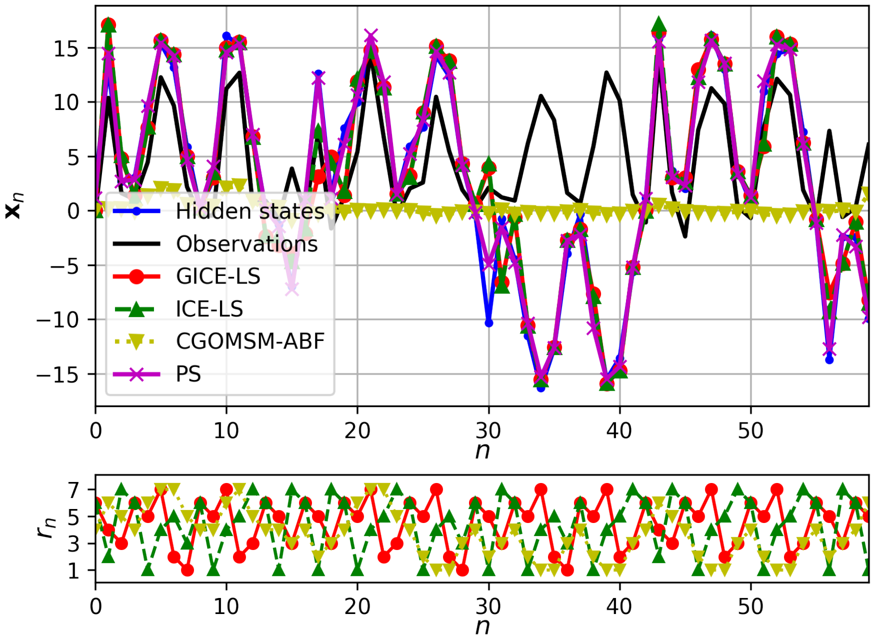


Figure 3.16: Trajectory example of KTGW model (60 samples, $K$=7).

### *KTGWSL model*

Regarding the KTGWSL model, we set $\mathbf{X}_1$, $\mathbf{V}_{n+1}$ and $\mathbf{U}_{n+1}$ follow the same distribution as the settings for KTGW model. For model identification, we consider two forms of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ for comparison. One is the linear form that has been applied already on SV models, which is defined by

$$\begin{aligned}
&\mathbf{A}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = a_{j,k} \\
&\mathbf{B}_{j,k}\left(\mathbf{y}_n^{n+1}\right) = b_{j,k}\mathbf{y}_n + c_{j,k}\mathbf{y}_{n+1} + d_{j,k}
\end{aligned} \tag{3.32}$$

The other one is the non-linear form as defined in (3.31) for KTGW experiment.

Results of ICE-LS and GICE-LS applied are reported in Table 3.16, while the performance of CGOMSM-ABF and Particle Filter are reported in Table 3.17.

Comparing the two subtables in Table 3.16, non-linear assumption of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ gets better restoration than linear assumption. This verifies again the importance of the appropriate chosen form of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$, which can be varied in GCOMSM but not CGOMSM. In addition, the best result got in both Table 3.16a and Table 3.16b are thorough GICE-LS. This could be cause by the generality of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ considered in the identification of GICE. It can be also inferred by comparing the best restoration of GICE-LS in Table 3.16a and the best restoration of CGOMSM-ABF in Table 3.17. Although not so significant, the flexibility of the distribution of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ assumed in GCOMSM make improvement of the fitness when approximate the non-stationary KTGWSL models by switching Markov model. A trajectory example of this experimental series is given in Figure 3.17.

In summary, the result of these two experimental series on KTGW models show that

1. Switching Markov models could be less efficient when approaching the non-stationary non-Gaussian non-linear system than a stationary one. Under non-stationary case, more switching classes can not always leads to better approximation. So, when we chose $K$, it is not the larger the better.

2. The flexible consideration of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ and $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ in the proposed GCOMSM contribute to the improvement from CGOMSM of the fitness to the non-Gaussian non-linear model. In practice, the non-linear extension from CGOMSM to GCOMSM could be more significant than the

Table 3.16: MSE results of ICE-LS and GICE-LS on KTGWSL model.

(a) Linear $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ assumption.

| | $K$ | 2 | 3 | 4 | 5 | 7 |
|---|---|---|---|---|---|---|
| Filtering | ICE-LS | 6.81 | 6.13 | 6.10 | 4.83 | 4.70 |
| | GICE-LS | 6.79 | 6.25 | 5.80 | 4.80 | 4.63 |
| Smoothing | ICE-LS | 6.80 | 6.96 | 5.68 | 4.15 | 3.72 |
| | GICE-LS | 6.79 | 6.05 | 5.46 | 4.02 | 3.61 |

(b) Non-linear $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ assumption.

| | $K$ | 2 | 3 | 4 | 5 | 7 |
|---|---|---|---|---|---|---|
| Filtering | ICE-LS | 6.77 | 2.93 | 3.32 | 2.87 | 2.85 |
| | GICE-LS | 6.51 | 4.00 | 3.00 | 2.93 | 2.75 |
| Smoothing | ICE-LS | 5.85 | 2.71 | 3.05 | 2.32 | 2.43 |
| | GICE-LS | 5.76 | 3.56 | 2.76 | 2.37 | 2.26 |

Table 3.17: MSE results of CGOMSM-ABF on KTGWSL model.

| $K$ | 2 | 3 | 4 | 5 | 7 | PF |
|---|---|---|---|---|---|---|
| Filtering | 5.98 | 5.62 | 5.36 | 4.72 | 4.90 | 1.69 |
| Smoothing | 5.37 | 5.29 | 4.56 | 3.71 | 4.03 | 1.32 |

non-Gaussian extension of $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$.

3. CGOMSM is a special case of GCOMSM, but ICE-LS and CGOMSM-ABF is more accurate identification method than GICE-LS as they have less consideration of distribution shapes. One may need to consider the balance of applying a suitable model and keep accuracy of the identification when dealing with a practical issue.

When wondering how to chose the identification methods among CGOMSM-ABF, ICE-LS and GICE-LS, a simple way is to observe the restoration MSE of learning sample set and chose the method who gets the minimum MSE. Normally, the MSE of learning sample set is close to the result got from the testing set.

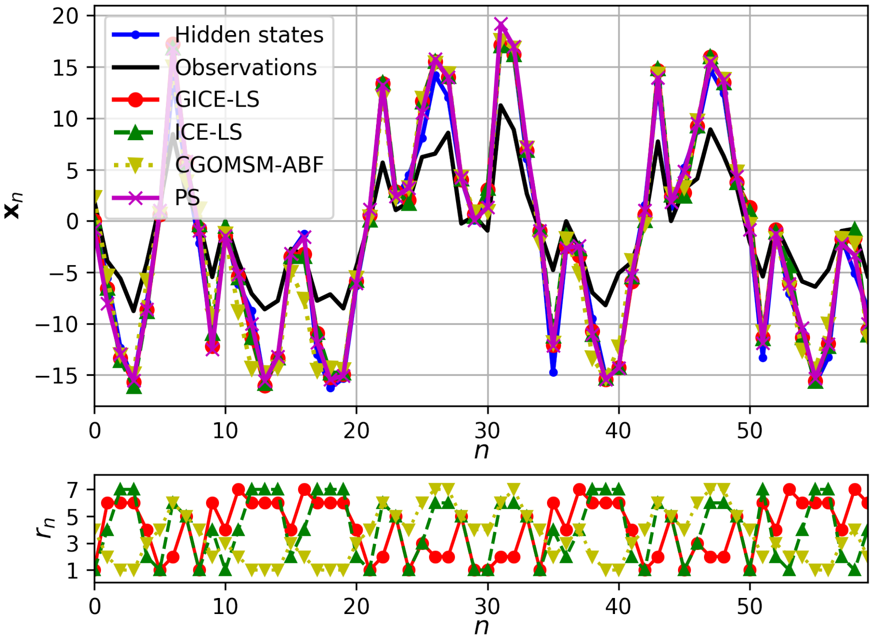


Figure 3.17: Trajectory example of KTGWSL model (60 samples, $K$=7).

## 3.5 Conclusion

The CGOMSM model introduced in Chapter 2 is extended to a more general switching model called "Generalized Conditionally Observed Markov Switching Model" (GCOMSM). GCOMSM can incorporate any distribution of $p\left(\mathbf{y}_{n+1}\left|\mathbf{y}_n, \mathbf{r}_n^{n+1}\right.\right)$ and includes non-linear formation of the regime of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right.\right)$ on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$, while still keeping the advantage of CGOMSM that the optimal restorations are feasible.

This Chapter defines the new general GCOMSM, gives the model simulation method, and derives the associated optimal restorations (filtering and smoothing). Two different examples of data simulation and optimal restorations of GCOMSM are given. One is with special Gaussian linear settings which degenerates the model to the CGOMSM, the other is with general non-Gaussian non-linear settings to show the interest of the extension in GCOMSM. Moreover, a GCOMSM identification method based on the recent "Generalized Iterative Conditional Estimation"

(GICE) and the Least-square (LS) principles from sample data set is proposed, called “GICE-LS”. The identification ability of GICE-LS for GCOMSM is verified by also two experiments on Gaussian linear and non-Gaussian non-linear GCOMSM data respectively with comparison to a variant identification method called “ICE-LS” which combines the classic “Iterative Conditional Estimation” (ICE) and LS principles assuming that $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ is Gaussian; and also to the identification method of CGOMSM called CGOMSM-ABF. Results show that the identification and restoration methods which suit for Gaussian linear switching model are no more valid for approximating the general GCOMSM and getting its appropriate restoration. Finally, experiments on the restorations for GCOMSM approximation identified by GICE-LS and ICE-LS for other generable non-Gaussian non-linear systems (stochastic volatility models and Kitagawa models) are conducted, with comparison to the restoration for CGOMSM approximation identified by CGOMSM-ABF, and by Particle Filter. The results show that GCOMSM can perform better when approximating a non-stationary non-Gaussian non-linear system than CGOMSM. Approaching an unknown non-Gaussian non-linear system with GCOMSM by GICE-LS, then restoring by the optimal restorations of the approximated GCOMSM can be an alternative of MCMC based methods under high dimension state-space condition, since MCMC based methods will become much more time consuming when large amount of particle is required.

Chapter 4

# Conclusion and perspectives

Switching Markov models are widely used in many fields to describe the dynamic state-space systems. When applying switching Markov models on imitating real systems, the issues of learning their suitable parameters and data restoration (filtering and smoothing) are indispensable. This dissertation focuses on finding solutions for these two problems of recent switching Markov models without taking use of the Markov Chain Monte-Carlo (MCMC) method which is the generic train of thought when dealing with these problems.

The main contribution of this dissertation is two folds:

1. An unsupervised parameter estimation method named “Double EM” is proposed for the recent Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM) which is based on two successive Expectation-Maximization (EM) algorithms:

    a) EM for discrete state-space Pairwise Markov Chain (PMC), with a mild approximation that the pair of switches and observations, $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$, has the Markov property in CGPMSM.

    b) An extension of the EM algorithm for constant parameter Pairwise Gaussian Markov Model (GPMM) to switching case, under condition that the switches are known, called “Switching EM”.

    Besides, two restoration approaches were proposed for CGPMSM:

    a) one is based on parameter modification to a sub-model known as Conditionally Gaussian Observation Markov Switching Model (CGOMSM), called “CGO-Appro”.

b) A second one is based on EM algorithm assuming that $\left(\mathbf{R}_1^N, \mathbf{Y}_1^N\right)$ is a PMC, called “EM-Appro”.

Simulations were conducted to evaluate all proposed methods. Results show that Switching EM can furnish good estimation of parameters for Gaussian switching case. The two restoration approaches are superior to other parameter modification based restorations and can get competitive results *w.r.t.* Particle Filter. Integrally, the Double EM algorithm combined with the EM-Appro works well on solving the unsupervised restoration problem of CGPMSM. Its performance even has great chance to surpass the other sub-optimal supervised restoration methods. In addition, the effects of observation means and noise level defined by covariances on restoration are studied in the meanwhile.

2. Copulas are introduced in the CGOMSM and fused to a more general one, called “Generalized Conditionally Observed Markov Switching Model” (GCOMSM). The main advantage of this general switching Markov model is that, it incorporates more flexible distributions and regimes while still allows the fast optimal restoration as CGOMSM does. The extensions are on two aspects that

   a) Introduction of copulas in the distribution of observations conditionally on switches, $p\left(\mathbf{y}_n^{n+1} \left| \mathbf{r}_n^{n+1}\right.\right)$, enriches the distributions in GCOMSM which are always assumed Gaussian or Gaussian mixtures in the classic CGOMSM.

   b) The auto-regressive function $\mathcal{G}\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$ is linear on $\mathbf{x}_n$, but can be of any form on $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$ in GCOMSM, whereas they are all linear defined in the classic CGOMSM.

   Moreover, an identification method called “GICE-LS” is proposed to learn the distributions and parameters of time-independent GCOMSM from its sample data set $\left(\mathbf{x}_1^N, \mathbf{y}_1^N\right)$. GICE-LS is based on two principles:

   a) The “Generalized Iterative Conditional Estimation” principle (GICE) for identifying $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$ from candidate forms and estimating the associated parameters.

b) The Least-Square (LS) principle for estimating the parameters of the supposed regime form of $\mathcal{G}\left(\mathbf{x}_{n+1}\left|\mathbf{x}_n, \mathbf{y}_n^{n+1}, \mathbf{r}_n^{n+1}\right.\right)$.

Experiments verify the capability of GCOMSM to work on data under flexible distributions and non-linear regimes. The GICE-LS can get proper distributions and parameters of GCOMSM and the associated optimal restorations work well on the data simulated from GCOMSM model, while the methods of identification and restorations for the inchoate CGOMSM turn out to be an improper choice for the data following this more general system settings. The "identification-restoration" method combining GICE-LS and optimal restoration of GCOMSM is also tested on other generable non-Gaussian non-linear systems (the Stochastic Volatility and Kitagawa models), results show the merits of the extensions embedded in GCOMSM comparing to CGOMSM.

Due to the limitation of time, the efficiency of proposed methods in this dissertation has not been evaluated by real data applications. Also, the proposed methods may still have some inadequacies and maybe some unnecessary assumptions. Considering the current state of the methods, the future work may include:

1. For the unsupervised restoration of CGPMSM

a) The performance of Double EM is dependent on the accuracy of the realization of $\mathbf{R}_1^N$ from the first EM principle applied, which can be further replaced by the probability of $p\left(r_n\left|\mathbf{y}_1^N\right.\right)$, to reduce the influence brought by arbitrariness of the MPM criterion on Switching EM.

b) In this work, the parameter initialization of Switching EM is assumed to be not very far away from the true one. An initialization method will be incorporated or developed later for completing the Double EM algorithm for suiting the real issues.

c) The proposed Double EM can not work on the CGOMSM, for which, we may find another parameter estimation method other than applying the

"EM" principle.

2. For the identification of newly proposed GCOMSM model

   a) Only parameter estimations of $\mathbf{A}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ and $\mathbf{B}_{\mathbf{r}_n^{n+1}}\left(\mathbf{y}_n^{n+1}\right)$ in equation (3.18) are considered in GICE-LS, as they are already sufficient for the restoration under a GCOMSM model. Further, we can include the estimation of $\boldsymbol{\mathcal{V}}_{\mathbf{r}_n^{n+1}}$, since the fully consideration of the parameterization of $\left(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}\right)$ may also influent the estimation of each individual parameter.

   b) In the implementation of GICE, the Maximum-Likelihood (ML) principle is used for all parameter estimation of the stationary distribution $p\left(\mathbf{y}_n^{n+1}\left|\mathbf{r}_n^{n+1}\right.\right)$. In practice, it can be replaced by other alternative methods. For example, to get the parameters of some marginal distributions, the moments method can be considered; and to get the estimation of copulas, the empirical calculation of Kendall's tau, $\tau$ can replace the calculation of $\alpha$ [81]. They are all worth a try for comparison. In addition, we use the semi-parametric method to estimate the parameters of copulas. It might be also interesting to try the other copula estimation methods, such as non-parametric methods to further improve the GICE efficiency [10], [73].

   c) The model and methods proposed in this dissertation are easy to extend to higher dimensional state-space, at least when parameters are known. Their interest with respect to MCMC based methods could increase when the state-space dimension grows, since under high dimension circumstance, much more particles will be required by MCMC methods.

Appendix A

# Maximization of the likelihood function in Switching EM

The likelihood (2.47) we want to maximize in the Switching EM concerns the parameter $\boldsymbol{\Theta}_4$ as

$$L(\boldsymbol{\Theta}_4) = \sum_{n=1}^{N-1} L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right), \tag{A.1}$$

with

$$L_n\left(\boldsymbol{\Theta}_4(\mathbf{r}_n^{n+1})\right) = \mathbb{E}\left[\ln p\left(\mathbf{z}'_{n+1}\left|\mathbf{z}'_n\right.\right)\right], \tag{A.2}$$

where $\mathbf{z}'_{n+1} = \mathbf{z}_{n+1} - \mathbf{M}^{\mathbf{z}}(r_{n+1})$ and $\mathbf{z}'_n = \mathbf{z}_n - \mathbf{M}^{\mathbf{z}}(r_n)$. $p\left(\mathbf{z}'_{n+1}\left|\mathbf{z}'_n\right.\right)$ is Gaussian. Specifically, when $r_n = j$, $r_{n+1} = k$:

$$\begin{aligned}
&L_n\left(\boldsymbol{\Theta}_4(r_n = j, r_{n+1} = k)\right)\\
=&\mathbb{E}\left\{\ln\left[\frac{1}{\sqrt{(2\pi)^q|\boldsymbol{\mathcal{Q}}_{j,k}|}}exp\left(-\frac{1}{2}\left(\mathbf{z}'_{n+1} - \boldsymbol{\mathcal{F}}_{j,k}\mathbf{z}'_n\right)^{\intercal}\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\left(\mathbf{z}'_{n+1} - \boldsymbol{\mathcal{F}}_{j,k}\mathbf{z}'_n\right)\right)\right]\right\}\\
=&-\frac{1}{2}\left\{q\ln(2\pi) + \ln|\boldsymbol{\mathcal{Q}}_{j,k}| + \operatorname{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\mathbb{E}\left(\left(\mathbf{z}'_{n+1} - \boldsymbol{\mathcal{F}}_{j,k}\mathbf{z}'_n\right)\left(\mathbf{z}'_{n+1} - \boldsymbol{\mathcal{F}}_{j,k}\mathbf{z}'_n\right)^{\intercal}\right)\right]\right\}\\
=&-\frac{1}{2}\left\{q\ln(2\pi) + \ln|\boldsymbol{\mathcal{Q}}_{j,k}| + \operatorname{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}}\right] - \operatorname{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\intercal}\right]\right.\\
&\left.-\operatorname{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\boldsymbol{\mathcal{F}}_{j,k}{}^{\intercal}\right] + \operatorname{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}\boldsymbol{\mathcal{F}}_{j,k}{}^{\intercal}\right]\right\},
\end{aligned} \tag{A.3}$$

in which

$$\begin{aligned}
\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n} &= \mathbb{E}\left[\mathbf{z}'_n\mathbf{z}'^t_n|\mathbf{y}_1^N\right]\\
&= \begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n)\\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n)\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_n)\\ \mathbf{y}_n - \mathbf{M}^{\mathbf{y}}(r_n)\end{bmatrix}^t + \begin{bmatrix}\boldsymbol{P}_{n|N} & 0\\ 0 & 0\end{bmatrix},
\end{aligned} \tag{A.4}$$

$$\begin{aligned}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n} &= \mathbb{E}\left[\mathbf{z}'_{n+1}\mathbf{z}'^{t}_{n}|\mathbf{y}_1^N\right] \\ &= \begin{bmatrix}\hat{\mathbf{x}}_{n+1|N} - \mathbf{M}^{\mathbf{x}}(r_{n+1}) \\ \mathbf{y}_{n+1} - \mathbf{M}^{\mathbf{y}}(r_{n+1})\end{bmatrix} \begin{bmatrix}\hat{\mathbf{x}}_{n|N} - \mathbf{M}^{\mathbf{x}}(r_{n}) \\ \mathbf{y}_{n} - \mathbf{M}^{\mathbf{y}}(r_{n})\end{bmatrix}^t \\ &+ \begin{bmatrix}\boldsymbol{C}_{n+1,n|N} & 0 \\ 0 & 0\end{bmatrix},\end{aligned} \tag{A.5}$$

as defined in (2.51) and (2.52).

Taking partial derivative of the likelihood function with respect to $\boldsymbol{\mathcal{F}}_{j,k}$, we get (2.53), and make it equal to zero we have

$$\begin{aligned}&\sum_{n=1}^{N-1}\delta_n\left(j,k\right)\left\{-\partial\mathrm{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\mathsf{T}}\right] - \partial\mathrm{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\right.\right. \\ &\left.\left.\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\boldsymbol{\mathcal{F}}_{j,k}{}^{\mathsf{T}}\right] + \partial\mathrm{tr}\left[\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}\boldsymbol{\mathcal{F}}_{j,k}{}^{\mathsf{T}}\right]\right\}\Big/\partial\boldsymbol{\mathcal{F}}_{j,k} \\ =&\sum_{n=1}^{N-1}\delta_n\left(j,k\right)\left\{-\left(\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\mathsf{T}}\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\right)^{\mathsf{T}} - \boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right. \\ &\left.+\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n} + \boldsymbol{\mathcal{Q}}_{j,k}{}^{-t}\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}\right)^{\mathsf{T}}\right\} \\ =&\sum_{n=1}^{N-1}\delta_n\left(j,k\right)\left\{-2\boldsymbol{\mathcal{Q}}_{j,k}{}^{-1}\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n} - \boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}\right)\right\} \\ =&0.\end{aligned} \tag{A.6}$$

So we get $\hat{\boldsymbol{\mathcal{F}}}_{j,k} = \widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_n,\mathbf{z}'_n}\right)^{-1}$, with $\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_n,\mathbf{z}'_n}$ and $\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}$ where

$$\begin{aligned}\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_n,\mathbf{z}'_n} &= \sum_{n=1}^{N-1}\delta_n\left(j,k\right)\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}; \\ \widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n} &= \sum_{n=1}^{N-1}\delta_n\left(j,k\right)\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}; \\ \widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}} &= \sum_{n=1}^{N-1}\delta_n\left(j,k\right)\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}},\end{aligned} \tag{A.7}$$

as defined in (2.56).

Also, taking partial derivative of the likelihood function (A.3) with respect to

$\boldsymbol{\mathcal{Q}}_{j,k}$ and making it equal to zero we have

$$\begin{aligned}
&\sum_{n=1}^{N-1} \delta_n\left(j,k\right)\Bigg\{\partial\ln|\boldsymbol{\mathcal{Q}}_{j,k}| + \partial\mathrm{tr}\left[{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}}\right] - \partial\mathrm{tr}\Bigg[\\
&{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}_{\mathbf{z}_{n+1},\mathbf{z}_n}\right)^{\mathsf{T}}\Bigg] - \partial\mathrm{tr}\left[{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}}\right] +\\
&\partial\mathrm{tr}\left[{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}}\right]\Bigg\}\Bigg/\partial\boldsymbol{\mathcal{Q}}_{j,k}\\
=&\boldsymbol{Card}\left(j,k\right){\boldsymbol{\mathcal{Q}}_{j,k}}^{-1} + \sum_{n=1}^{N-1}\delta_n\left(j,k\right)\Big\{-{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}}\right)^{\mathsf{T}}\\
&{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1} + {\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}}{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1} + {\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\\
&\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\mathsf{T}}{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1} - {\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}\right)^{\mathsf{T}}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}}{\boldsymbol{\mathcal{Q}}_{j,k}}^{-1}\Big\}\\
=&0.
\end{aligned}\tag{A.8}$$

After simplification, we have

$$\begin{aligned}
&\boldsymbol{Card}\left(j,k\right)\boldsymbol{\mathcal{Q}}_{j,k} + \sum_{n=1}^{N-1}\delta_n\left(j,k\right)\Big\{\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}} +\\
&\boldsymbol{\mathcal{F}}_{j,k}\left(\boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{\mathsf{T}} - \boldsymbol{C}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}} - \boldsymbol{\mathcal{F}}_{j,k}\boldsymbol{C}^{\mathbf{z}'_n,\mathbf{z}'_n}{\boldsymbol{\mathcal{F}}_{j,k}}^{\mathsf{T}}\Big\} = 0.
\end{aligned}\tag{A.9}$$

Bringing $\hat{\boldsymbol{\mathcal{F}}}_{j,k}$ into expression, we get

$$\hat{\boldsymbol{\mathcal{Q}}}_{j,k} = \frac{1}{\boldsymbol{Card}\left(j,k\right)}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_{n+1}} - \hat{\boldsymbol{\mathcal{F}}}_{j,k}\left(\widetilde{\boldsymbol{C}}_{j,k}^{\mathbf{z}'_{n+1},\mathbf{z}'_n}\right)^{t}\right). \tag{A.10}$$

Appendix B

# Particle filter for CGPMSM

Given the observations and parameters of CGPMSM, we are interested in two optimal problems:

1. **Filtering**: Obtain the filtering distribution $p\left(r_n, \mathbf{x}_n \left| \mathbf{y}_1^n \right.\right)$, and get the state estimation $\mathbb{E}\left[\mathbf{x}_n \left| \mathbf{y}_1^n \right.\right]$.
2. **Smoothing**: Obtain the smoothing (fixed interval) distribution $p\left(r_n, \mathbf{x}_n \left| \mathbf{y}_1^N \right.\right)$, and get the state estimation $\mathbb{E}\left[\mathbf{x}_n \left| \mathbf{y}_1^N \right.\right]$.

If we are able to sample $M$ random samples called particles $\left\{(\mathbf{r}_1^{n(m)}, \mathbf{x}_1^{n(m)}); m = 1, \ldots, M\right\}$ according to $p\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right)$, then an empirical estimation of this distribution would be given by

$$\overline{p_M}\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right) = \frac{1}{M}\sum_{m=1}^{M} \delta_{\left(\mathbf{r}_1^{n(m)}, \mathbf{x}_1^{n(m)}\right)}\left(d\mathbf{r}_1^n, d\mathbf{x}_1^n\right), \tag{B.1}$$

and also a corollary, one can easily estimate the mean of function $f\left(r_n, \mathbf{x}_n \left| \mathbf{y}_1^n \right.\right)$, noted by $I\left(f_{n|n}\right)$

$$\begin{aligned} \overline{I_M}\left(f_{n|n}\right) &= \int f\left(r_n, \mathbf{x}_n \left| \mathbf{y}_1^n \right.\right) \overline{p_M}\left(r_n, \mathbf{x}_n \left| \mathbf{y}_1^n \right.\right) dr_n d\mathbf{x}_n \\ &= \frac{1}{M}\sum_{m=1}^{M} f\left(r_n^{(m)}, \mathbf{x}_n^{(m)} \left| \mathbf{y}_1^n \right.\right). \end{aligned} \tag{B.2}$$

This estimate is unbiased and from strong law of large numbers (SLLN), $\overline{I_M}\left(f_{n|n}\right)$ converges almost surely toward $I\left(f_{n|n}\right)$ as $M \to +\infty$.

Solution to estimate the $p\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right)$ and to get $\overline{I_M}\left(f_{n|n}\right)$ consist of using the well-know importance sampling method. Let us introduce an arbitrary importance distribution $\pi\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right)$ and $p\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right) > 0$ implies $\pi\left(\mathbf{r}_1^n, \mathbf{x}_1^n \left| \mathbf{y}_1^n \right.\right) > 0$. Then

$$I\left(f_{n|n}\right) = \frac{\mathrm{E}_{\pi\left(\mathbf{r}_1^n,\mathbf{x}_1^n\left|\mathbf{y}_1^n\right.\right)p}\left[f\left(r_n,\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right),\omega(\mathbf{r}_1^n,\mathbf{x}_1^n)\right]}{\mathrm{E}_{\pi\left(\mathbf{r}_1^n,\mathbf{x}_1^n\left|\mathbf{y}_1^n\right.\right)}\left[\omega(\mathbf{r}_1^n,\mathbf{x}_1^n)\right]} \tag{B.3}$$

where the importance weight is equal to

$$\omega(\mathbf{r}_1^n,\mathbf{x}_1^n) = \frac{p\left(\mathbf{r}_1^n,\mathbf{x}_1^n\left|\mathbf{y}_1^n\right.\right)}{\pi\left(\mathbf{r}_1^n,\mathbf{x}_1^n\left|\mathbf{y}_1^n\right.\right)}, \tag{B.4}$$

If we have $M$ random samples $\left\{(\mathbf{r}_1^{n^{(m)}},\mathbf{x}_1^{n^{(m)}}); m = 1,\ldots,M\right\}$ distributed according to $\pi\left(\mathbf{r}_1^n,\mathbf{x}_1^n\left|\mathbf{y}_1^n\right.\right)$, then a Monte Carlo estimate of $I\left(f_{n|n}\right)$ is given by:

$$\begin{aligned}\overline{I_M}\left(f_{n|n}\right) &= \frac{\sum_{m=1}^{M} f\left(r_n^{(m)},\mathbf{x}_n^{(m)}\left|\mathbf{y}_1^n\right.\right)\omega\left(\mathbf{r}_1^{n^{(m)}},\mathbf{x}_1^{n^{(m)}}\right)}{\sum_{m=1}^{M}\omega\left(\mathbf{r}_1^{n^{(m)}},\mathbf{x}_1^{n^{(m)}}\right)} \\ &= \sum_{m=1}^{M}\tilde{\omega}_1^{n^{((m))}} f\left(r_n^{(m)},\mathbf{x}_n^{(m)}\left|\mathbf{y}_1^n\right.\right),\end{aligned} \tag{B.5}$$

where the normalized importance weights $\tilde{\omega}_1^{n^{((m))}}$ are equal to

$$\tilde{\omega}_1^{n^{((m))}} = \frac{\omega\left(\mathbf{r}_1^{n^{(m)}},\mathbf{x}_1^{n^{(m)}}\right)}{\sum_{m=1}^{M}\omega\left(\mathbf{r}_1^{n^{(m)}},\mathbf{x}_1^{n^{(m)}}\right)}. \tag{B.6}$$

## B.1 Particle Filter

Under CGPMSM, it is possible to reduce the problem of estimating $p\left(r_n,\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right)$ to sampling from $p\left(\mathbf{r}_1^n\left|\mathbf{y}_1^n\right.\right)$, since $p\left(\mathbf{r}_1^n,\mathbf{x}_n\left|\mathbf{y}_1^n\right.\right) = p\left(\mathbf{r}_1^n\left|\mathbf{y}_1^n\right.\right)p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,\mathbf{r}_1^n\right.\right)$, where $p\left(\mathbf{x}_n\left|\mathbf{y}_1^n,\mathbf{r}_1^n\right.\right)$ is Gaussian, and can be evaluated by Kalman filter. This simplification is the so called variance reduction in [43].

So, the estimation of $I\left(f_{n|n}\right)$ can be simplified to

$$\overline{I_M}\left(f_{n|n}\right) = \frac{\sum_{m=1}^{M} \mathrm{E}_{p\left(\mathbf{x}_n \left|\mathbf{y}_1^n, \mathbf{r}_1^{n(m)}\right.\right)}\left[f\left(r_n^{(m)}, \mathbf{x}_n \left|\mathbf{y}_1^n\right.\right)\right] \omega\left(\mathbf{r}_1^{n(m)}\right)}{\sum_{m=1}^{M} \omega\left(\mathbf{r}_1^{n(m)}\right)} \tag{B.7}$$

where

$$\omega(\mathbf{r}_1^n) = \frac{p\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right)}{\pi\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right)}. \tag{B.8}$$

### B.1.1 Sequential Importance Sampling

According to the structure of CGPMSM, we can rewrite the importance function at time $n$ as follows:

$$\begin{aligned}\pi\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right) &= \pi\left(r_1 \left|\mathbf{y}_1\right.\right) \frac{\pi\left(r_2, \mathbf{y}_2 \left|r_1, \mathbf{y}_1\right.\right)}{\pi\left(\mathbf{y}_2 \left|\mathbf{y}_1\right.\right)} \cdots \frac{\pi\left(r_n, \mathbf{y}_n \left|\mathbf{r}_1^{n-1}, \mathbf{y}_1^{n-1}\right.\right)}{\pi\left(\mathbf{y}_n \left|\mathbf{y}_1^{n-1}\right.\right)} \\ &= \pi\left(r_1 \left|\mathbf{y}_1\right.\right) \prod_{k=2}^{n} \frac{\pi\left(r_k, \mathbf{y}_k \left|\mathbf{r}_1^{k-1}, \mathbf{y}_1^{k-1}\right.\right)}{\pi\left(\mathbf{y}_k \left|\mathbf{y}_1^{k-1}\right.\right)},\end{aligned} \tag{B.9}$$

so that $\pi\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right)$ admits $\pi\left(\mathbf{r}_1^{n-1} \left|\mathbf{y}_1^{n-1}\right.\right)$ as marginal distribution at time $n-1$. We can propagate the estimated distribution of $p\left(\mathbf{r}_1^{n-1} \left|\mathbf{y}_1^{n-1}\right.\right)$ in time without modification, and so as the simulated particles $\left\{\mathbf{r}_1^{n-1(m)}; m = 1, \dots, M\right\}$. Such an importance function allows us to compute the importance weight recursively with $\omega(\mathbf{r}_1^n) = \omega\left(\mathbf{r}_1^{n-1}\right)\omega_n$, where the incremental weight $\omega_n$ is given by

$$\begin{aligned}\omega_n &= \frac{p\left(r_n, \mathbf{y}_n \left|\mathbf{r}_1^{n-1}, \mathbf{y}_1^{n-1}\right.\right)}{p\left(\mathbf{y}_n \left|\mathbf{y}_1^{n-1}\right.\right) \pi\left(r_n \left|\mathbf{r}_1^{n-1}, \mathbf{y}_1^n\right.\right)} \\ &\propto \frac{p\left(r_n, \mathbf{y}_n \left|\mathbf{r}_1^{n-1}, \mathbf{y}_1^{n-1}\right.\right)}{\pi\left(r_n \left|\mathbf{r}_1^{n-1}, \mathbf{y}_1^n\right.\right)}.\end{aligned} \tag{B.10}$$

### B.1.2 Importance distribution and weight

There are infinite possible choices for $\pi\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right)$, the only condition is that it should include the one of $p\left(\mathbf{r}_1^n \left|\mathbf{y}_1^n\right.\right)$, that is the support of $p\left(\mathbf{r}_1^n\right)$. To choose a proposal that minimizes the variance of the importance weights at time $n$, given $\mathbf{r}_1^{n-1}$ and $\mathbf{y}_1^n$

as the importance weight, the optimal importance distribution is $p\left(r_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^n\right.\right)$. It can be computed with

$$p\left(r_n=k\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^n\right.\right)=\frac{p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},r_n=k,\mathbf{y}_1^{n-1}\right.\right)p\left(r_n=m\left|r_{n-1}\right.\right)}{p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^{n-1}\right.\right)}. \tag{B.11}$$

The associate importance weight computed following (B.10) is

$$\omega_n\propto\frac{p\left(r_n,\mathbf{y}_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^{n-1}\right.\right)}{p\left(r_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^n\right.\right)}\propto p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^{n-1}\right.\right). \tag{B.12}$$

As $\mathbf{R}_1^N$ is Markov chain, we have

$$p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^{n-1}\right.\right)=\sum_{k=1}^{K}p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},r_n=k,\mathbf{y}_1^n\right.\right)p\left(r_n=k\left|r_{n-1}\right.\right). \tag{B.13}$$

We can also choose prior distribution $p\left(r_n\left|r_{n-1}\right.\right)$ as importance distribution, then the associate importance weight is

$$\begin{aligned}\omega_n&\propto\frac{p\left(r_n,\mathbf{y}_n\left|\mathbf{r}_1^{n-1},\mathbf{y}_1^{n-1}\right.\right)}{p\left(r_n\left|\mathbf{r}_1^{n-1}\right.\right)}\\&\propto p\left(\mathbf{y}_n\left|\mathbf{r}_1^{n-1},r_n=k,\mathbf{y}_1^{n-1}\right.\right).\end{aligned} \tag{B.14}$$

### B.1.3 Sampling importance resampling (SIR)

Assuming that before we have weighted distribution $\widetilde{p_M}\left(\mathbf{r}_1^n|\mathbf{y}_1^n\right)=\sum_{m=1}^{M}\tilde{\omega}_n^{(m)}\delta_{\tilde{\mathbf{r}}_1^{n(m)}}\left(d\mathbf{r}_1^n\right)$ with particles $\tilde{\mathbf{r}}_1^{n(m)}$, and after sampling from $\widetilde{p_M}\left(\mathbf{r}_1^n|\mathbf{y}_1^n\right)$ $M$ times, we get new particles $\mathbf{r}_1^{n(m)}$, and have all weights become 1, the estimated distribution becomes

$$\widehat{p_M}\left(\mathbf{r}_1^n|\mathbf{y}_1^n\right)=M^{-1}\sum_{m=1}^{M}\delta_{\mathbf{r}_1^{n(m)}}\left(d\mathbf{r}_1^n\right). \tag{B.15}$$

Resampling allows reallocating particles from low-density regions into high-density ones making thus a more optimal use of available articles.

## B.2 Particle Smoother

The simulation based filter can be straightforwardly extended to smoothing. As we have the Monte Carlo approximation of $p\left(\mathbf{r}_1^N \left|\mathbf{y}_1^N\right.\right)$ that

$$\widehat{p_M}\left(\mathbf{r}_1^N|\mathbf{y}_1^N\right) = M^{-1}\sum_{m=1}^{M}\delta_{\mathbf{r}_1^{N(m)}}\left(d\mathbf{r}_1^N\right). \tag{B.16}$$

Therefore, the estimation of the marginal distribution is

$$\widehat{p_M}\left(\mathbf{r}_1^n|\mathbf{y}_1^N\right) = M^{-1}\sum_{m=1}^{M}\delta_{\mathbf{r}_1^{n(m)}}\left(d\mathbf{r}_1^n\right). \tag{B.17}$$

However, this direct extension suffers from the so-called sample depletion problem, which means that the trajectories have been resampled $N-n$ times and it causes a loss of diversity of particles [45].

Appendix C

# Margins and copulas used in this dissertation

The standard form of marginal distribution studied in this dissertation are listed in Table C.1. Parameter set of present distributions are denoted by $\{\theta, loc, scale\}$, where "$loc$","$scale$" represent the location and scale of the distribution respectively from its standard form. In detail, $loc$ and $scale$ transform a distribution from standard one by $y = (y - loc)/scale$ and the pdf $f = f/scale$. "$\theta$" denotes the parameters other than $loc$ and $scale$, here can be absent, when the distribution is only defined by $loc$ and $scale$; $\theta^1$, if only one parameter besides $loc$ and $scale$ presents (maybe replaced by $\theta$); or $\left\{\theta^1, \theta^2\right\}$ if two parameters present.

Table C.1: Marginal distributions studied in this dissertation.

| Name | cdf $F$ | pdf $f$ | parameter $\theta$ |
|---|---|---|---|
| Gamma[1] | $F = \frac{\gamma\left(\theta^1, y\right)}{\Gamma\left(\theta^1\right)}$ | $f = \frac{y^{\theta^1-1} exp(-y)}{\Gamma\left(\theta^1\right)}$ | $\theta^1 > 0$ |
| Fisk[2] | $F = \frac{1}{1+y^{-\theta^1}}$ | $f = \frac{\theta^1 y^{\theta^1-1}}{\left(1+y^{\theta^1}\right)^2}$ | $\theta^1 > 0$ |
| Gaussian[3] | $F = \frac{1}{2}\left(1 + erf\left(\frac{y}{\sqrt{2}}\right)\right)$ | $f = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{y^2}{2}\right)$ | - |
| Laplace[4] | $F = \begin{cases} \frac{1}{2} exp\left(y\right) & if\ y < 0 \\ 1 - \frac{1}{2} exp\left(-y\right) & if\ y \geq 0 \end{cases}$ | $f = \frac{1}{2} exp\left(-\lvert x \rvert\right)$ | - |
| Beta[5] | $F = \frac{I_y\left(\theta^1, \theta^2\right)}{B\left(\theta^1, \theta^2\right)}$ | $f = \frac{\Gamma\left(\theta^1+\theta^2\right) y^{\theta^1-1} (1-y)^{\theta^2-1}}{\Gamma\left(\theta^1\right)\Gamma\left(\theta^2\right)}$ | $\theta^1 > 0, \theta^2 > 0$ |
| Beta prime[6] | $F = I_{\frac{y}{1+y}}\left(\theta^1, \theta^2\right)$ | $f = \frac{y^{\theta^1-1}(1+y)^{-\theta^1-\theta^2}}{B\left(\theta^1, \theta^2\right)}$ | $\theta^1 > 0, \theta^2 > 0$ |

[1] $\gamma\left(x, =\right) \int_0^\infty t^{x-1} exp\left(-t\right) dt$ is a complete Gamma function and $\gamma\left(s, x\right) = \int_0^x t^{s-1} exp\left(-t\right) dt$ represents the lower incomplete gamma function.

[2] Also known as log-logistic distribution.

[3] $erf\left(x\right) = \frac{2}{\sqrt{\pi}} \int_0^x exp\left(-t^2\right) dt$ represents the error function.

[4] Sometimes called double exponential distribution.

[5] $B\left(x, s\right) = \int_0^1 t^{x-1}\left(1-t\right)^{s-1} dt$ is the Beta function and $I_x\left(a, b\right) = \int_0^x t^{a-1}\left(1-t\right)^{b-1} dt$ is the incomplete Beta function.

[6] Also called beta distribution of the second kind or inverted beta distribution.

Table C.2: Copulas studied in this dissertation (all are one parameterized named $\alpha$).

| Name | cdf $C$ | pdf $c$ | $[\alpha_{min}, \alpha_{max}]$ |
|---|---|---|---|
| Gumbel[7] | $C = exp\left(-\left(U_1+U_2\right)^{\frac{1}{\alpha}}\right)$ | $c = \frac{U_1}{u_1 ln(u_1)} \frac{U_2}{u_2 ln(u_2)} \left(\alpha - 1 + U_1 + U_2\right)^{\frac{1}{\alpha}} \left(U_1+U_2\right)^{\frac{1}{\alpha}-2} exp\left(-\left(U_1+U_2\right)^{\frac{1}{\alpha}}\right)$ | $[1, +\infty)$ |
| Gauss[8] | $C = \int_0^{u_1} \phi\left(\frac{\phi^{-1}(u_2)-\rho\phi^{-1}(u)}{\sqrt{1-\rho^2}} du\right)$ | $c = \frac{1}{1-\alpha^2} exp\left(1\frac{1}{2}\xi^{\intercal}(\rho - \boldsymbol{I})\xi\right)$ | $[-1, 1]$ |
| Clayton | $C = \left(u_1^{-\alpha} + u_2^{-\alpha} - 1\right)^{\frac{1}{\alpha}}$ | $c = (1+\alpha) u_1^{-1-\alpha} u_2^{-1-\alpha} \left(-1 + u_1^{-\alpha} + u_2^{-\alpha}\right)^{-\frac{1}{\alpha}-2}$ | $[0, +\infty)$ |
| FGM | $C = u_1 u_2 \left(1 + \alpha\left(1-u_1\right)\left(1-u_2\right)\right)$ | $c = 1 + \alpha\left(1-2u_1\right)\left(1-2u_2\right)$ | $[-1, 1]$ |
| Arch12[9] | $C = \left(1 + \left(U_1+U_2\right)^{\frac{1}{\alpha}}\right)^{-1}$ | $c = \frac{U_1}{u_1(u_1-1)} \frac{U_2}{u_2(u_2-1)} \left(\alpha - 1 + (\alpha+1)\left(U_1+U_2\right)^{\frac{1}{\alpha}}\right) \frac{(U_1+U_2)^{\frac{1}{\alpha}-2}}{\left(1+(U_1+U_2)^{\frac{1}{\alpha}}\right)^3}$ | $[1, +\infty)$ |
| Arch14[10] | $C = \left(1 + \left(U_1+U_2\right)^{\frac{1}{\alpha}}\right)^{-\alpha}$ | $c = U_1 U_2 \left(U_1+U_2\right)^{\frac{1}{\alpha}-2} \left(1 + \left(U_1+U_2\right)^{\frac{1}{\alpha}}\right)^{-2-\alpha} \frac{\alpha-1+2\alpha(U_1+U_2)^{\frac{1}{\alpha}}}{\alpha u_1 u_2 \left(u_1^{\frac{1}{\theta}}-1\right)\left(u_2^{\frac{1}{\theta}}-1\right)}$ | $[1, +\infty)$ |
| Product | $C = u_1 u_2$ | $c = 1$ | - |

[7] Familly of Archimedean copulas, $U_1 = \left(-ln\left(u_1\right)\right)^{\alpha}$ and $U_2 = \left(-ln\left(u_2\right)\right)^{\alpha}$.

[8] Familly of elliptical copulas, $\xi_i = \phi^{-1}(u_i)$ where $\phi$ here represents the standard normal distribution, $\rho = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix}$ is the $2 \times 2$ correlation and $\boldsymbol{I}$ is the identity matrix.

[9] $U_1 = \left(\frac{1}{u_1} - 1\right)^{\alpha}$ and $U_2 = \left(\frac{1}{u_2} - 1\right)^{\alpha}$

[10] $U_1 = \left(u_1^{-\frac{1}{\alpha}}\right)^{\alpha}$ and $U_2 = \left(u_2^{-\frac{1}{\alpha}}\right)^{\alpha}$.

All the copulas studied in this article are one parameter copulas, listed in Table C.2. The solutions of the maximum of $\underset{u_2 \in [0,1]}{max} c_{n+1}\left(u_1, u_2 | \mathbf{r}_n^{n+1}\right)$ for copulas which can be closed-form are listed in Table C.3.

Table C.3: Closed-form solutions for $u_2 = \arg\max u_2 \in [0,1] c\left(u_1, u_2\right)$ and $max\left(c\left(u_1, u_2\right)\right)$ of several copulas ($u_1 \in [0,1]$).

| Name | $u_2$ | $max\left(c\left(u_1, u_2\right)\right)$ |
|---|---|---|
| Gaussian | $\phi\left(\frac{\phi^{-1}(u_1)}{\alpha}\right)$ | $\frac{1}{\sqrt{1-\alpha^2}} exp\left(\frac{1}{2}\left[\phi^{-1}\left(u_1\right)\right]^2\right)$ |
| FGM | $\begin{cases} 0 & \text{if } \left(1-2u_1\right)\alpha > 0 \\ 1 & \text{if } \left(1-2u_1\right)\alpha < 0 \\ - & \text{else} \end{cases}$ | $1 + max\left(\left(1-2u_1\right)\alpha, -\left(1-2u_1\right)\alpha\right)$ |
| Clayton | $min\left(1, \left(\frac{\alpha+1}{\alpha}\left(u_1^{-\alpha}-1\right)\right)^{-\frac{1}{\alpha}}\right)$ | $\begin{cases} \left(1+\alpha\right)u_1^{\alpha} & \text{if } u_2 = 1 \\ \left(\frac{\alpha+1}{2\alpha+1}^{\frac{2\alpha+1}{\alpha}}\right)\frac{\alpha}{u_1\left(1-u_1^{\alpha}\right)} & \text{else} \end{cases}$ |
| Product | - | 1 |

# Bibliography

[1] Noufel Abbassi, Dalila Benboudjema, Stéphane Derrode, and Wojciech Pieczynski. Optimal filter approximations in conditionally Gaussian pairwise Markov switching models. *IEEE Transactions on Automatic Control*, 60(4):1104–1109, 2015. xiv, xv, 13, 18, 20, 21, 53, 70

[2] Noufel Abbassi, Dalila Benboudjema, and Wojciech Pieczynski. Kalman filtering approximations in triplet Markov Gaussian switching models. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 77–80. IEEE, 2011. 21

[3] Guy A Ackerson and King-Sun Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1):10–17, 1970. xiv

[4] Phillip L Ainsleigh, Nasser Kehtarnavaz, and Roy L Streit. Hidden Gauss-Markov models for signal classification. *IEEE Transactions on Signal Processing*, 50(6):1355–1367, 2002. 13

[5] Boujemaa Ait-El-Fquih and François Desbouvries. Unsupervised signal restoration in partially observed Markov chains. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–III. IEEE, 2006. 15, 36, 39

[6] Boujemaa Ait-El-Fquih and François Desbouvries. Fixed-interval Kalman smoothing algorithms in singular state–space systems. *Journal of Signal Processing Systems*, 65(3):469–478, Dec 2011. 18

[7] Christophe Andrieu, Manuel Davy, and Arnaud Doucet. Efficient particle filtering for jump Markov systems. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–1625–II–1628, May 2002. 18

[8] Christophe Andrieu, Manuel Davy, and Arnaud Doucet. Efficient particle filtering for jump Markov systems. application to time-varying autoregressions. *IEEE Transactions on signal processing*, 51(7):1762–1770, 2003. xiv

[9] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002. 13, 93

[10] Armand Kodjo Atiampo and Georges Laussane Loum. Unsupervised image segmentation with pairwise Markov chains based on nonparametric estimation of copula using orthogonal polynomials. *International Journal of Image and Graphics*, 16(04):1650020, 2016. 120

[11] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs London, UK:, 1995. xiv

[12] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970. xiv, 5

[13] André Berchtold. The double chain Markov model. *Communications in Statistics - Theory and Methods*, 28(11):2569–2589, 1999. 4

[14] André Berchtold. High-order extensions of the double chain Markov model. *Stochastic Models*, 18(2):193–227, 2002. 4

[15] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and Data Analysis*, 41(3):561 – 575, 2003. Recent Developments in Mixture Model. 9

[16] Ake Björck. *Numerical methods for least squares problems.* SIAM, 1996. 91

[17] El-Kébir Boukas. *Stochastic switching systems: analysis and design.* Springer Science & Business Media, 2007. xiv

[18] Eric Bouyé, Valdo Durrleman, Ashkan Nikeghbali, Gaël Riboulet, and Thierry Roncalli. Copulas for finance – A reading guide and some applications. 2000. xv

[19] Bjørn Braathen, Wojciech Pieczynski, and Pascal Masson. Global and local methods of unsupervised Bayesian segmentation of images. *Machine Graphics and Vision*, 2(1):39–52, 1993. 9

[20] Matthew Brand. *Coupled hidden Markov models for modeling interacting processes.* MIT Media Lab Perceptual Computing/Learning and Common Sense Technical Report 405 (Revised), 1997. 4

[21] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden Markov models for complex action recognition. In *Computer vision and pattern recognition, 1997. proceedings., 1997 ieee computer society conference on*, pages 994–999. IEEE, 1997. 4

[22] Chris Brooks. *Introductory Econometrics for Finance.* Cambridge University Press, 2 edition, 2008. 105

[23] Kenneth M Brown and John E Dennis. A new algorithm for nonlinear least-Squares curve fitting. *Mathematical Software*, page 391, 2014. 91

[24] Nicolas Brunel and Wojciech Pieczynski. Unsupervised signal restoration using hidden Markov chains with copulas. *Signal processing*, 85(12):2304–2315, 2005. xv, 72

[25] Olivier Cappé. Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749, 2011. 7

[26] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 18

[27] Gilles Celeux and Jean Diebolt. A stochastic approximation type EM algorithm for the mixture problem. *Stochastics: An International Journal of Probability and Stochastic Processes*, 41(1-2):119–134, 1992. 10

[28] Chaw-Bing Chang and Michael Athans. State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems*, (3):418–425, 1978. xiv

[29] Barbara Choroś, Rustam Ibragimov, and Elena Permiakova. Copula estimation. *Copula theory and its applications*, pages 77–91, 2010. 6

[30] Drew Creal. A survey of sequential Monte Carlo methods for economics and finance. *Econometric reviews*, 31(3):245–296, 2012. 105

[31] Yves Delignon, Abdelwaheb Marzouki, and Wojciech Pieczynski. Estimation of generalized mixtures and its application in image segmentation. *IEEE Transactions on image processing*, 6(10):1364–1375, 1997. 9

[32] Jean Pierre Delmas. An equivalence of the EM and ICE algorithm for exponential family. *IEEE transactions on signal processing*, 45(10):2613–2615, 1997. 10

[33] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 7

[34] Stéphane Derrode, Cyril Carincotte, and Salah Bourennane. Unsupervised image segmentation based on high-order hidden Markov chains [radar imaging examples]. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*, volume 5, pages V–769. IEEE, 2004. 1

[35] Stéphane Derrode and Wojciech Pieczynski. Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9):2477–2489, 2004. 1, 5

[36] Stéphane Derrode and Wojciech Pieczynski. Exact fast computation of optimal filter in Gaussian switching linear systems. *IEEE Signal Processing Letters*, 20(7):701–704, 2013. 48

[37] Stéphane Derrode and Wojciech Pieczynski. Unsupervised data classification using pairwise Markov chains with automatic copulas selection. *Computational Statistics & Data Analysis*, 63:81–98, 2013. xv, 72

[38] Stéphane Derrode and Wojciech Pieczynski. Unsupervised classification using hidden Markov chain with unknown noise copulas and margins. *Signal Processing*, 128:8–17, 2016. xv, 72, 87, 89, 96

[39] François Desbouvries, Yohan Petetin, and Boujemaa Ait-El-Fquih. Direct, prediction-and smoothing-based Kalman and particle filter algorithms. *Signal Processing*, 91(8):2064–2077, 2011. 110

[40] Pierre A Devijver. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3:369–373, 1985. 5

[41] José G Dias, Jeroen K Vermunt, and Sofia Ramos. Mixture hidden Markov models in finance research. In *Advances in data analysis, data handling and business intelligence*, pages 451–459. Springer, 2009. xiv

[42] Arnaud Doucet and Christophe Andrieu. Iterative algorithms for state estimation of jump Markov linear systems. *IEEE transactions on signal processing*, 49(6):1216–1227, 2001. xiv, 18

[43] Arnaud Doucet, Neil J Gordon, and Vikram Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on signal processing*, 49(3):613–624, 2001. 18, 126

[44] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009. 105

[45] Arnaud Doucet, Andrew Logothetis, and Vikram Krishnamurthy. Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Transactions on Automatic Control*, 45(2):188–202, 2000. 129

[46] Baum Leonard E and Petrie Ted. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966. xiii

[47] Mahmoud Elmezain, Ayoub Al-Hamadi, Jorg Appenrodt, and Bernd Michaelis. A hidden Markov model-based continuous gesture recognition system for hand motion trajectory. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008. xiii

[48] David C Emanuel and James D MacBeth. Further results on the constant elasticity of variance call option pricing model. *Journal of Financial and Quantitative Analysis*, 17(4):533–554, 1982. 105

[49] Paul Embrechts. Copulas: A personal view. *Journal of Risk and Insurance*, 76(3):639–650, 2009. xv

[50] Yariv Ephraim and Brian L Mark. Causal recursive parameter estimation for discrete-time hidden bivariate Markov chains. *IEEE Trans. Signal Processing*, 63(8):2108–2117, 2015. 4

[51] Yariv Ephraim, Brian L Mark, et al. Bivariate Markov processes and their estimation. *Foundations and Trends® in Signal Processing*, 6(1):1–95, 2013. 4

[52] Yariv Ephraim and Neri Merhav. Hidden Markov processes. *IEEE Transactions on information theory*, 48(6):1518–1569, 2002. 4

[53] Dean Fantazzini. Dynamic copula modelling for value at risk. *Frontiers in Finance and Economics, Forthcoming*, 2008. 71

[54] Carsten Fritsche, Emre Özkan, and Fredrik Gustafsson. Online EM algorithm for jump Markov systems. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1941–1946. IEEE, 2012. 18

[55] Mark Gales and Steve Young. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008. xiii

[56] René Garcia and Georges Tsafack. Dependence structure and extreme comovements in international equity and bond markets. *Journal of Banking & Finance*, 35(8):1954–1970, 2011. 71

[57] Jim Gatheral. *The volatility surface: a practitioner's guide*, volume 357. John Wiley & Sons, 2011. 105

[58] Eric Ghysels, Andrew C Harvey, and Eric Renault. Stochastic volatility. *Handbook of statistics*, 14:119–191, 1996. 105

[59] Nathalie Giordana and Wojciech Pieczynski. Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):465–475, 1997. 89

[60] Neil Gordon, Branko Ristic, and Sanjeev Arulampalam. Beyond the Kalman filter: Particle filters for tracking applications. *Artech House, London*, 830, 2004. 93

[61] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini, and Wojciech Pieczynski. Exact fast smoothing in switching models with application to stochastic volatility. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 924–928. IEEE, 2015. xv, 92

[62] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini, and Wojciech Pieczynski. Fast filtering in switching approximations of nonlinear Markov systems with applications to stochastic volatility. *IEEE Transactions on Automatic Control*, 62(2):853–862, 2017. xv, 17, 92, 105, 108

[63] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994. 28

[64] Mark S Handcock, Garry Robins, Tom Snijders, Jim Moody, and Julian Besag. Assessing degeneracy in statistical models of social networks. Technical report, Citeseer, 2003. 18

[65] Andrew C Harvey and Neil Shephard. Estimation of an asymmetric stochastic volatility model for asset returns. *Journal of Business & Economic Statistics*, 14(4):429–434, 1996. 105

[66] Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993. 105

[67] Xuedong D Huang, Yasuo Ariki, and Mervyn A Jack. *Hidden Markov models for speech recognition*, volume 2004. Edinburgh university press Edinburgh, 1990. xiii

[68] David Huard, Guillaume Évin, and Anne-Catherine Favre. Bayesian copula selection. *Computational Statistics & Data Analysis*, 51(2):809–822, 2006. 96

[69] Eric Jacquier, Nicholas G Polson, and Peter E Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 20(1):69–87, 2002. 105

[70] Piotr Jaworski, Fabrizio Durante, Wolfgang Karl Hardle, and Tomasz Rychlik. *Copula theory and its applications*. Springer, 2010. 96

[71] Kengo Kamatani. Local degeneracy of Markov chain Monte Carlo methods. *ESAIM: Probability and Statistics*, 18:713–725, 2014. 18

[72] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima. Withdrawn: Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics*, 173(2):321–343, 2005. 91

[73] Göran Kauermann, Christian Schellhase, and David Ruppert. Flexible copula density estimation with penalized hierarchical b-splines. *Scandinavian Journal of Statistics*, 40(4):685–705, 2013. 120

[74] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 96

[75] Chang-Jin Kim, Charles R Nelson, et al. State-space models with regime switching: classical and Gibbs-sampling approaches with applications. *MIT Press Books*, 1, 1999. 93

[76] Gunky Kim, Mervyn J Silvapulle, and Paramsothy Silvapulle. Comparison of semiparametric and parametric methods for estimating copulas. *Computational Statistics and Data Analysis*, 51(6):2836–2850, 2007. 94, 96

[77] Nam Soo Kim, Tae Gyoon Kang, Shin Jae Kang, Chang Woo Han, and Doo Hwa Hong. Speech feature mapping based on switching linear dynamic system. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):620–631, Feb 2012. 13, 18

[78] Sangjoon Kim, Neil Shephard, and Siddhartha Chib. Stochastic volatility: likelihood inference and comparison with ARCH models. *The review of economic studies*, 65(3):361–393, 1998. 105

[79] Genshiro Kitagawa. Non-Gaussian state — space modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041, 1987. 110

[80] Genshiro Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. 110

[81] William R. Knight. A computer method for calculating Kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439, 1966. 120

[82] Piere Lanchantin. *Chaînes de Markov triplets et segmentation non supervisée de signaux.* PhD thesis, Institut National des Télécommunications, Evry, France, 2006. 7

[83] Pierre Lanchantin, Jérôme Lapuyade-Lahorgue, and Wojciech Pieczynski. Unsupervised segmentation of triplet Markov chains hidden with long-memory noise. *Signal Processing*, 88(5):1134–1151, 2008. 9

[84] Pierre Lanchantin, Jérôme Lapuyade-Lahorgue, and Wojciech Pieczynski. Unsupervised segmentation of randomly switching data hidden with non-gaussian correlated noise. *Signal Processing*, 91(2):163–175, 2011. 4, 14

[85] Pierre Lanchantin and Wojciech Pieczynski. Unsupervised non stationary image segmentation using triplet Markov chains. *Advanced Concepts for Intelligent Vision Systems (ACVIS 04)*, 2004. 10

[86] Pierre Lanchantin and Wojciech Pieczynski. Unsupervised restoration of hidden nonstationary Markov chains using evidential priors. *IEEE Transactions on Signal processing*, 53(8):3091–3098, 2005. 5

[87] Steven Le Cam, Christophe Collet, and Fabien Salzenstein. Acoustical respiratory signal analysis and phase detection. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3629–3632. IEEE, 2008. 1

[88] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 91

[89] Fuchun Li. Identifying asymmetric comovements of international stock market returns. *Journal of Financial Econometrics*, 12(3):507–543, 2014. 71

[90] Andrew Logothetis and Vikram Krishnamurthy. Expectation maximization algorithms for MAP estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 47(8):2139–2156, 1999. xiv

[91] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. *Methods for Non-Linear Least Squares Problems (2nd ed.)*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004. 91

[92] Rogemar S Mamon and Robert J Elliott. *Hidden markov models in finance*, volume 4. Springer, 2007. xiv

[93] Morelande Marc R and Moran Bill. An unscented transformation for conditionally linear models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 3, pages III–1417–III–1420, April 2007. 18

[94] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 91

[95] Pascale Masson and Wojciech Pieczynski. SEM algorithm and unsupervised statistical segmentation of satellite images. *IEEE transactions on geoscience and remote sensing*, 31(3):618–633, 1993. 10

[96] Efim Mazor, Amir Averbuch, Yakov Bar-Shalom, and Joshua Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on aerospace and electronic systems*, 34(1):103–123, 1998. xiv

[97] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007. 7

[98] Alexander J McNeil and Johanna Nešlehová. Multivariate Archimedean copulas, d-monotone functions and l1-norm symmetric distributions. *The Annals of Statistics*, pages 3059–3097, 2009. 6

[99] Gabriele Moser, Josiane Zerubia, and Sebastiano B Serpico. Dictionary-based stochastic Expectation-Maximization for SAR amplitude probability density function estimation. *IEEE Transactions on Geoscience and Remote Sensing*, 44(1):188–200, 2006. 10

[100] Roger B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 6

[101] Valérian Némesin and Stéphane Derrode. Robust blind pairwise Kalman algorithms using QR decompositions. *IEEE Transactions on Signal Processing*, 61(1):5–9, 2013. 15, 39

[102] Valérian Némesin and Stéphane Derrode. Robust partial-learning in linear Gaussian systems. *IEEE Transactions on Automatic Control*, 60(9):2518–2523, 2015. 15

[103] M. Netto, L. Gimeno, and M. Mendes. On the optimal and suboptimal nonlinear filtering problem for discrete-time systems. *IEEE Transactions on Automatic Control*, 23(6):1062–1067, Dec 1978. 110

[104] Nam Thanh Nguyen, Dinh Q Phung, Svetha Venkatesh, and Hung Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 955–960. IEEE, 2005. xiii

[105] Nikolay Y Nikolaev, Lilian M Menezes, and Evgueni Smirnov. Nonlinear filtering of asymmetric stochastic volatility models and value-at-risk estimation. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2104 IEEE Conference on*, pages 310–317. IEEE, 2014. 105

[106] Yasuhiro Omori and Toshiaki Watanabe. Block sampler and posterior mode estimation for asymmetric stochastic volatility models. *Computational Statistics & Data Analysis*, 52(6):2892–2910, 2008. 105

[107] MR Osborne. Nonlinear least squares - the Levenberg algorithm revisited. *The ANZIAM Journal*, 19(3):343–357, 1976. 91

[108] Emre Özkan, Fredrik Lindsten, Carsten Fritsche, and Fredrik Gustafsson. Recursive maximum likelihood identification of jump Markov nonlinear systems. *IEEE Transactions on Signal Processing*, 63(3):754–765, 2015. 18

[109] I Papila and O Ersoy. Multiscale segmentation of remotely sensed images using pairwise Markov chains. In *Antennas and Propagation Society International Symposium, 2004. IEEE*, volume 2, pages 2123–2126. IEEE, 2004. xiv, 1

[110] Andrew J Patton. Modelling asymmetric exchange rate dependence. *International economic review*, 47(2):527–556, 2006. 71

[111] Anrong Peng and Wojciech Pieczynski. Adaptive mixture estimation and unsupervised local Bayesian image segmentation. *Graphical Models and image processing*, 57(5):389–399, 1995. 9

[112] Yohan Petetin and François Desbouvries. A class of fast exact Bayesian filters in dynamical models with jumps. *IEEE Transactions on Signal Processing*, 62(14):3643–3653, 2014. 48

[113] Wojciech Pieczynski. Statistical image segmentation. *Machine graphics and vision*, 1(1/2):261–268, 1992. 9

[114] Wojciech Pieczynski. Pairwise markov chains. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):634–639, 2003. xiv, 1, 4

[115] Wojciech Pieczynski. Exact filtering in conditionally markov switching hidden linear models. *Comptes Rendus Mathematique*, 349(9-10):587–590, 2011. 18, 48

[116] Wojciech Pieczynski and François Desbouvries. Kalman filtering using pairwise Gaussian models. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 6, pages VI–57. IEEE, 2003. 13

[117] Wojciech Pieczynski, Cedric Hulard, and Thomas Veit. Triplet Markov chains in hidden signal restoration, 2003. 15

[118] James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random structures and Algorithms*, 9(1-2):223–252, 1996. 4

[119] Lawrence Rabiner and B Juang. An introduction to hidden Markov models. *ieee assp magazine*, 3(1):4–16, 1986. xiii

[120] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. xiii

[121] Selwa Rafi, Marc Castella, and Wojciech Pieczynski. Pairwise Markov model applied to unsupervised image separation. In *SPPRA'11: The Eighth IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*. Acta Press, 2011. 7

[122] Luis Rodríguez and Inés Torres. Comparative study of the Baum-Welch and Viterbi training algorithms applied to read and spontaneous speech recognition. *Pattern Recognition and Image Analysis*, pages 847–857, 2003. xiii

[123] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982. 7

[124] Abe Sklar. *Fonctions de Répartition À N Dimensions Et Leurs Marges*. Université Paris 8, 1959. 6, 71

[125] Jason F. Smith, Ajay Pillai, Kewei Chen, and Barry Horwitz. Identification and validation of effective connectivity networks in functional magnetic resonance imaging using switching linear dynamic systems. *NeuroImage*, 52(3):1027–1040, 2010. Computational Models of the Brain. 18

[126] J Michael Steele. *Non-Uniform Random Variate Generation (Luc Devroye)*. Society for Industrial and Applied Mathematics, 1987. 75

[127] Rolf Sundberg. Maximum likelihood theory and applications for distributions generated when observing a function of an exponential family variable. dissertation. 1971. 7

[128] Rolf Sundberg. Maximum likelihood theory for incomplete data from an exponential family. *Scandinavian Journal of Statistics*, 1(2):49–58, 1974. 7

[129] Rolf Sundberg. An iterative method for solution of the likelihood equations for incomplete data from exponential families. *Communication in Statistics-Simulation and Computation*, 5(1):55–64, 1976. 7

[130] Makoto Takahashi, Yasuhiro Omori, and Toshiaki Watanabe. Estimating stochastic volatility models using daily returns and realized volatility simultaneously. *Computational Statistics & Data Analysis*, 53(6):2404–2426, 2009. 105

[131] Stephen J Taylor. *Modelling financial time series.* world scientific, 2008. 105

[132] Pravin K Trivedi, David M Zimmer, et al. Copula modeling: an introduction for practitioners. *Foundations and Trends® in Econometrics*, 1(1):1–111, 2007. xv

[133] Jitendra Tugnait. Adaptive estimation and identification for discrete systems with Markov jump parameters. *IEEE Transactions on Automatic control*, 27(5):1054–1065, 1982. 18

[134] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967. xiii

[135] Chien-Fu Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103, 1983. 7

[136] Meriem Yahiaoui, Emmanuel Monfrini, and Bernadette Dorizzi. Implementation of unsupervised statistical methods for low-quality iris segmentation. In *SITIS 2014 : 10th International Conference on Signal-Image Technology and Internet-Based Systems*, pages 566 – 573, Marrakech, Morocco, November 2014. IEEE. xiv, 1

[137] Bingduo Yang, Yuhua Li, and Peiqin Zhang. Semiparametric es timation for index copula models. *SSRN Electronic Journal*, Jan 2016. 94