Triplet Markov models and fast smoothing in switching systems

Wojciech Pieczynski, Stéphane Derrode, Ivan Gorynin, Emmanuel Monfrini

Triplet Markov models and fast smoothing in switching systems

Preprint, August, 30, 2018.

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

SAMOVAR, Telecom SudParis, CNRS, Université Paris-Saclay, 9, rue Charles Fourrier,
91000 Evry, France
wojciech.pieczynski@telecom-sudparis.eu

Abstract. The aim of the paper is twofold. The first aim is to present a mini tutorial on « pairwise Markov models » (PMMs) and "triplet Markov models" (TMMs) which extend the popular "hidden Markov models" (HMMs). The originality of these extensions is due to the fact that the hidden data does not need to be Markov. More precisely, for X hidden data and Y observed ones, the originality of PMMs is that X does not need to be Markov, and the originality of TMMs is that even (X, Y) does not need to be Markov. In spite of these lacks of Markovianity fast processing methods, similar to those applied in HMMs or their other extensions, remain workable. The second goal is to present an original switching model approximation allowing fast smoothing. The method we propose, called "double filtering based smoothing" (DFBS), uses a particular TMM in which the pair (X, R), where R models switches, is not Markov. It is based on two filters, and uses a class of models, known as conditionally Gaussian observed Markov switching models (CGOMSMs), where exact fast filtering is feasible. The original model is approximated by two CGOMSMs in order to process the past data and the future data in direct and reverse order, respectively. Then state estimates produced by these two models are fused to provide a smoothing estimate. The DFBS is insensitive to the dimensions of the hidden and observation space and appears as an alternative to the classic particle smoothing in the situations where the latter cannot be applied due to its high processing cost.

Keywords. Conditionally Gaussian linear state-space model, smoothing, conditionally Gaussian observed Markov switching model, double filtering based smoothing, Markov switching systems.

1. INTRODUCTION

Hidden Markov models (HMMs) are widely used in wide range of applications. For $Y_1^N = (Y_1,\ldots,Y_N)$ observed process and $X_1^N = (X_1,\ldots,X_N)$ hidden state process, in classic HMMs the distribution is given by Markov distribution of X_1^N and distributions of Y_1^N conditional on X_1^N. In spite of the fact that HMMs are among the simplest non i.i.d. models, they turn out to be very robust and can give excellent results in complex situations. Hundreds of papers are published on the subject each year so we only mention a few general papers or books. Hundreds of papers are published on the subject each year so that

we only cite some general papers or books [10, 13, 18, 31, 39, 50, 57]. HMMs have been extended in different directions leading to factorial models [22], hierarchical models [20], hidden bivariate Markov models [19], or still hidden semi-Markov models [9, 59, 60]. However, in all these extensions, hidden data - possibly containing additional latent data like in hidden semi-Markov models - remains Markov. This is not the case in "pairwise" and "triplet" Markov models (PMMs and TMMs, respectively). In PMMs one directly assumes that the pair (X_1^N, Y_1^N) is Markov. Then X_1^N is not necessary Markov and assuming it Markov can even appear as useless constraint. Important is that in PMM X_1^N is always Markov conditionally on Y_1^N, which allows processing similar to those in HMMs. Introduced in [47] for discrete finite data the idea was applied to continuous hidden data in [49]. PMMs have been successfully applied in different situations [25, 35] and some theoretical considerations can be seen in [16, 36]. In TMMs one adds a third process R_1^N and assumes the triplet (X_1^N, R_1^N, Y_1^N) Markov. Proposed in [46] for discrete finite (X_1^N, R_1^N) TMMs have been in particular successfully applied to non-stationary unsupervised processing [24, 34, 52]. Besides, links with Dempster-Shafer theory of evidence allowed to propose Dempster-Shafer fusion in Markovian context [11]. The idea has then been applied to continuous hidden (X_1^N, R_1^N) [3, 15]. As (X_1^N, Y_1^N) is not necessarily Markov in TMMs, they are more general than PMMs. For example, for Gaussian TMM (X_1^N, R_1^N, Y_1^N) there are cases in which Kalman filter is workable while (X_1^N, Y_1^N) is not Markov [15]. Crux difference between general TMMs [15, 46] and other previous TMMs is that, as in PMMs, the hidden process (X_1^N, R_1^N) is not necessarily Markov. The first part of this paper is devoted to a synthetic presentation of the PMMs and TMMs models.

The second aim is to introduce a new fast smoothing in Markov switching systems modeled by particular TMMs. Let us consider a TMM (X_1^N, R_1^N, Y_1^N) with X_1^N continuous and R_1^N discrete finite. R_1^N models stochastic switches and such a model is thus a switching one. The classic models to deal with switching systems are "switching linear dynamical systems" (SLDSs [4, 5, 7, 8, 13, 14, 17, 28, 29, 30, 38, 41, 43, 54, 56, 61, 62]). They are used in different fields, such as econometrics [29], finance [6], target tracking [7, 23, 58], speech recognition [40, 51], pattern recognition [44], among others. These systems are also known as jump Markov models (processes), switching conditional linear Gaussian state-space models, or still interacting multiple models. In SLDSs, the hidden variables take their values in a hybrid state space which includes continuous-valued and discrete-valued components. The idea is simple and seems natural: the hidden discrete switches are modeled by a Markov chain, and the hidden continuous states process is a classic Gaussian Markov chains conditionally on switches. In spite of the fact that a SDLS is a hidden Markov - since the hidden hybrid process (X_1^N, R_1^N) is Markov - there is no exact fast Bayesian filtering or smoothing algorithm tractable in the general SLDS context [13, 18, 37, 53, 54], so that different approximate methods have been used instead. In particular, smoothing, which is workable in systems without switches [12], is particularly difficult to perform in SLDSs. Previous research on smoothed inference in SLDSs includes Kim and Nelson's most popular method [30], simulation-based algorithms [17, 21, 44], smoothed inference by expectation correction [8], and various deterministic approximations [64]. The simulation-based methods intrinsically use Monte-Carlo integration in the state space and are asymptotically – when the number of particles used tends to infinity - optimal. Thus the accuracy of these approaches depends on the number of simulated particles and their great deal may be required to obtain an

efficient approximation. Conversely, if the number of simulated particles is insufficient for the state space dimension, these estimators would have high variance, while achieving an acceptable variance would mean for them a high processing cost. A common property of all these mentioned methods is that the hidden couple (X_1^N, R_1^N) is Markov. Relaxing the Markovianity of (X_1^N, R_1^N) leads to models allowing fast exact optimal filtering. First models allowing this, called "Conditionally Markov switching hidden linear models" (CMSHLMs) have been proposed in [48]. Then particular Gaussian switching CMSHLMs called "conditionally Gaussian observed switching Markov models" (CGOMSMs) have been subsequently proposed and different experiments attested of their interest with respect to particle-based methods [1, 2, 25, 26, 45, 63]. In particular, stationary CGOMSMs can be used to approximate any stationary Markov non-linear non-Gaussian system and then fast filtering and smoothing in approximating CGOMSMs are alternatives to particle methods [25, 26]. Another significant difference between CMSHLMs and classic SDLSs is that the pair formed by the discrete hidden variables and the observable ones (R_1^N, Y_1^N) is Markov in CMSHLMs, while it is not Markov in SLDSs.

After having recalled CMSHLMs, CGOMSMs, and related fast filtering in sections 3 and 4 we propose an original approximated fast smoothing valid in general "conditionally Gaussian pairwise Markov switching models" (CGPMSMs, [1, 2]) which simultaneously extends CGOMSMs and Gaussian SLDSs. Therefore a given CGPMSM can be approximated by a CGOMSM and then the fast smoothing performed in this approximating CGOMSM can be seen as an approximated solution of smoothing problem in CGPMSM. Such a method has been proposed and studied in [25], and parameter estimation method, resulting in unsupervised smoothing, has been recently proposed in [63]. However, smoothing in CGOMSMs suffers from the following dissymmetry. For $Y_1^N = (Y_1,\ldots,Y_N)$ observed process, $R_1^N = (R_1,\ldots,R_N)$ hidden switches process, and $X_1^N = (X_1,\ldots,X_N)$ hidden state process, $p(x_n|y_1,\ldots,y_n)$ in approximating CGOMSM is close to the similar distribution in CGPMSM, and thus $(y_1,\ldots,y_n)$ contains comparable information about x_n in both CGOMSM and CGPMSM. However, approximating true $p(x_n|y_n,\ldots,y_N)$ in CGPMSM with the same distribution in CGOMSM is less efficient. More precisely, $p(x_n|y_1,r_1,\ldots,y_n,r_n)$ depend on all $(y_1,\ldots,y_n)$ while $p(x_n|y_n,r_n,\ldots,y_N,r_N) = p(x_n|y_n,r_n,r_{n+1}\ldots,r_N)$, and thus only depends on y_n. As $p(x_n|r_n,y_n\ldots,y_N) = \sum_{(r_{n+1},\ldots,r_N)} p(r_{n+1},\ldots,r_N|y_{n+1},\ldots,y_N)p(x_n|y_n,\, r_{n+1}\ldots,r_N)$, $p(x_n|r_n,y_n\ldots,y_N)$ also depend on all $(y_n,\ldots,y_N)$; however, the dependence is obtained in somewhat indirect way and thus should be, *a priori*, less close to the true CGPMSM's $p(x_n|r_n,y_n\ldots,y_N)$.

Then the idea of the new method proposed in this paper is to approximate a given CGPMSM by CGOMSMs varying with $n = 1$, …, N. For given n the proposed "n-CGOMSM" approximation is the classic "left to right" CGOMSM for $Y_1^n = (Y_1,\ldots,Y_n)$, $R_1^n = (R_1,\ldots,R_n)$, $X_1^n = (X_1,\ldots,X_n)$, and a "right to left" CGOMSM obtained by inverting time for $X_n^N = (X_n,\ldots,X_N)$, $R_n^N = (R_n,\ldots,R_N)$, $Y_n^N = (Y_n,\ldots,Y_N)$. Such a possibility will be showed to exist in particular "stationary in law" family of CGPMSMs, in which $p(x_n, y_n|r_n)$ are Gaussians - with parameters possibly varying with n – for each $n = 1$, …, N. Let us also mention another smoothing approximated algorithm based on two classic

approximating filters called "switching Kalman filters" (SKF) filters proposed in [27]. The basic idea consisting on two "direct" and "reverse" filters is similar; however, the filters are quite different from those used in this paper and the fusion of information, called "Bayesian assimilation", they provide is quite different as well. In particular, Proposition 4.1 in section 4 is new.

Let us stress on great generality of TMMs. CMSHLMs and CGOMSMs have been obtained by relaxing the classic "hierarchical" way of defying the switching system distributions. In SLDSs hidden data are assumed Markov and then one adds the distributions $p(y_1^N | r_1^N, x_1^N)$, which model the "noise" and which are almost systematically very simple. General TMMs do not make any difference among X_1^N, R_1^N, and Y_1^N. Therefore (X_1^N, R_1^N, Y_1^N) is Markov but each of the six sequences X_1^N, R_1^N, Y_1^N, (X_1^N, R_1^N), (R_1^N, Y_1^N), (X_1^N, Y_1^N) can be Markov or not. CMSHLMs and SCGOMSMs are then particular TMMs in which (R_1^N, Y_1^N) is Markov and (X_1^N, R_1^N) is not, while the converse is true in SLDSs.

The organization of the paper is the following.

General pairwise and triplet Markov models are recalled and discussed in section 2. Section 3 is devoted to fast filtering and smoothing in CMSHLMs and CGOMSMs. The new smoothing method is proposed in Section 4, and section 5 contains some experiments. Concluding remarks and perspectives are proposed in the last section 6.

2. TRIPLET MARKOV MODELS

2.1 Hidden Markov models

Classic hidden Markov model (HMM) is a couple of stochastic sequences (X_1^N, Y_1^N), with $X_1^N = (X_1, \ldots, X_N)$, $Y_1^N = (Y_1, \ldots, Y_N)$, verifying hypotheses (H1), (H2), (H3) below:

(H1) $X_1^N = (X_1, \ldots, X_N)$ is Markov;

(H2) Y_1, …, Y_N are independent conditionally on $(X_1, \ldots, X_N)$;

(H3) distribution of each Y_n conditional on $(X_1, \ldots, X_N)$ is equal to its distribution conditional on Y_n.

Equivalently, one can say that (X_1^N, Y_1^N) is a HMM if and only if its distribution is written:

$$p(x_1^N, y_1^N) = p(x_1) p(y_1 | x_1) \prod_{n=2}^{N} p(x_n | x_{n-1}) p(y_n | x_n) . \tag{2.1.1}$$

Then (H1) is equivalent to

$$p(x_1^N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_{n-1}) , \tag{2.1.2}$$

(H2) is equivalent to

$$p(y_1^N \big| x_1^N) = \prod_{n=1}^{N} p(y_n \big| x_1^N) \,, \tag{2.1.3}$$

and (H3) is equivalent to

$$\text{for } n = 1 \,, \ldots, \, N \,, \; p(y_n \big| x_1^N) = p(y_n | x_n) \,. \tag{2.1.4}$$

HMMs have huge and various applications, due to the fact that when Y_1^N is observed and X_1^N is searched for, it can be estimated - in several particular models - by fast Bayesian methods. This is the case in at least two situations, especially dealt with in this paper:

Case (i): X_1^N is discrete finite and Y_1^N is either discrete finite or continuous (real or multidimensional);
Case (ii): (X_1^N, Y_1^N) is multidimensional Gaussian.

The fast processing possibilities stem from the fact that X_1^N is Markov conditionally on Y_1^N .
HMM defined with (H1)-(H3), or equivalently with (2.1.1), will be called in the following HMM "with independent noise" (HMM-IN).

Let as assume that we are either in Case (i) or in Case (ii). The core point of developments in sub-section 2.2 below is to note that none of hypotheses (H1)-(H3) is a necessary one to make possible Bayesian processing of interest. Indeed, one can see that HMM-IN considered as a couple $Z_1^N = (X_1^N, Y_1^N) = (Z_1, \ldots, Z_N)$, with $Z_n = (X_n, Y_n)$ for $n = 1, \ldots, N$, is a Markov process, with transitions $p(x_n, y_n | x_{n-1}, y_{n-1}) = p(x_n | x_{n-1}) p(y_n | x_n)$. Then the same Bayesian processing are workable in models verifying Markovianity of $Z_1^N = (Z_1, \ldots, Z_N)$ as the only condition. In other words, once Markovianity of $Z_1^N = (Z_1, \ldots, Z_N)$ has been admitted, hypotheses (H1)-(H3) appear as useless constraints. In particular, X_1^N may be not Markov. However, in such cases a non Markov X_1^N becomes Markov after having been conditioned upon Y_1^N .

2.2 Pairwise Markov models

Markov $Z_1^N = (Z_1,\ldots,Z_N)$ is called "pairwise Markov model" (PMM).

Thus relaxing one, or two, or even three, hypotheses among (H1)-(H3) and keeping Markovianity of Z_1^N we get different extensions of HMM-INs. Equivalently, considering a PMM Z_1^N and adding some hypotheses among (H1)-(H3) we obtain different particular PMMs.

Let Z_1^N be a PMM. The general form of its distribution is

$$p(x_1^N, y_1^N) = p(x_1, y_1) \prod_{n=2}^{N} p(x_n, y_n | x_{n-1}, y_{n-1}) = \tag{2.2.1}$$

$$p(x_1, y_1) \prod_{n=2}^{N} p(x_n | x_{n-1}, y_{n-1}) p(y_n | x_n, x_{n-1}, y_{n-1}).$$

Let us specify three particular cases of PMMs, recently studied in [24].

(a) Adding (H1)-(H3) to obtain a HMM-IN is equivalent to assume that

$$p(x_n | x_{n-1}, y_{n-1}) = p(x_n | x_{n-1}); \tag{2.2.2}$$

and

$$p(y_n | x_n, x_{n-1}, y_{n-1}) = p(y_n | x_n). \tag{2.2.3}$$

(b) Adding just (H1) we obtain a HMM with "correlated noise" (HMM-CN). An example of an HMM-CN is obtained considering (2.2.1) and (2.2.2). However, let us notice that (2.2.2) is not a necessary condition for a PMM to be a HMM-CN in general. As we will see in the following, (2.2.3) will be equivalent to (H1) in particular stationary and reversible PMMs;

(c) Adding just (H2) we obtain a PMM with "independent noise" (PMM-IN). An example of a PMM-IN is obtained considering (2.2.1) and (2.2.3). However, (2.2.3) is not a necessary condition for a PMM to be a PMM-IN in general. We will see that a stationary and reversible PMM is a PMM-IN if and only if $p(y_n | x_n, x_{n-1}, y_{n-1}) = p(y_n | x_n, x_{n-1})$.

Remark 2.1

One can notice that in PMMs both hidden and observed process play symmetrical roles. In particular, likely to the observed Y_1^N, the hidden X_1^N is not Markov in general. This can appear as somewhat disturbing as the distribution of X_1^N is not known explicitly in general. However, X_1^N is Markov conditionally on Y_1^N, which allows same Bayesian processing as in classic HMM-INs. Besides,

symmetric roles of X_1^N and Y_1^N can sometimes be justified in real situations. For example, if X_1^N is inflation rate and Y_1^N is interest rate, each of them can be seen as a noisy version of the other. Each of them can be considered as hidden and searched from the other, considered as observed.

We said above that assuming a hypothesis among (H1)-(H3) can be seen as useless constraint. Let us show that in some situations assuming X_1^N Markov is even quite detrimental. We have the following

Proposition 2.1

Let $Z_1^N = (Z_1,...,Z_N)$, with $Z_n = (X_n, Y_n)$ for $n = 1, \ldots, N$, be a stationary reversible PMM (which means that $p(x_n, y_n, y_{n+1}, x_{n+1})$ is independent from $n = 1, \ldots, N-1$, and distributions $p(y_{n+1}, x_{n+1} | x_n, y_n)$ are equal to distributions $p(y_n, x_n | x_{n+1}, y_{n+1})$). Then the following conditions are equivalent:

(C1) X_1^N is Markov;

(C2) $p(y_2 | x_1, x_2) = p(y_2 | x_2)$ (equivalent to $p(y_1 | x_1, x_2) = p(y_1 | x_1)$;

(C3) $p(y_n | x_1^N) = p(y_n | x_n)$ for each $n = 1, \ldots, N$.

The proof can be seen in [34]. We see that in Proposition 2.1 frame assuming X_1^N Markov is equivalent to simplifying the noise distribution when, in general PMMs, $p(y_n | x_1^N)$ depends on all $x_1, \ldots, x_N$. Besides, in the same frame, we see that (H1) and (H3) are equivalent. Finally, let us notice that this result allows for easily constructing PMMs without markovianity of the hidden process.

Example 2.1

Let us consider the case of stationary reversible PMM with X_1^N discrete finite and Y_1^N continuous real. The PMM distribution is given by $p(x_1, y_1, x_2, y_2) = p(x_1, x_2) p(y_1, y_2 | x_1, x_2)$. Let us assume the distributions $p(y_1, y_2 | x_1, x_2)$ Gaussian. If the mean or/and variance of Gaussian $p(y_2 | x_1, x_2)$ depend on x_1, we can state, by virtue of the Proposition, that X_1^N is not Markov. Besides, the PMM transitions are

$$p(y_{n+1}, x_{n+1} | x_n, y_n) = \frac{p(x_n, x_{n+1}) p(y_n, y_{n+1} | x_n, x_{n+1})}{\sum_{x_{n+1}} p(x_n, x_{n+1}) p(y_n | x_n, x_{n+1})}, \tag{2.2.4}$$

and thus they are not Gaussian. As a consequence, $p(y_1^N | x_1^N)$ are not Gaussian either. Finally, when $n = 1, \ldots, N$, distributions $p(y_n | x_1^N)$ are not Gaussian and depend on all $x_1, \ldots, x_N$. As a consequence, $p(y_n | x_n)$ are rich mixtures of non-Gaussian distributions. Of course, we find again classic HMM-IN transitions for $p(y_2 | x_1, x_2) = p(y_2 | x_2)$ and $p(y_1, y_2 | x_1, x_2) = p(y_1 | x_1) p(y_2 | x_2)$: $p(y_{n+1}, x_{n+1} | x_n, y_n) = p(x_{n+1} | x_n) p(y_{n+1} | x_{n+1})$.

Example 2.2

Let us consider the case of stationary reversible PMM with X_1^N and Y_1^N continuous multidimensional, and (X_1^N, Y_1^N) Gaussian. Its distribution is defined by Gaussian distribution of (X_1, Y_1, X_2, Y_2), given by means $M_X = E[X_1] = E[X_2]$, $M_Y = E[Y_1] = E[Y_2]$, and variance-covariance matrix

$$\Gamma = \begin{bmatrix} \Gamma_{X_1} & \Gamma_{X_1 Y_1} & \Gamma_{X_1 X_2} & \Gamma_{X_1 Y_2} \\ \Gamma_{Y_1 X_1} & \Gamma_{Y_1} & \Gamma_{Y_1 X_2} & \Gamma_{Y_1 Y_2} \\ \Gamma_{X_2 X_1} & \Gamma_{X_2 Y_1} & \Gamma_{X_2} & \Gamma_{X_2 Y_2} \\ \Gamma_{Y_2 X_1} & \Gamma_{Y_2 Y_1} & \Gamma_{Y_2 X_2} & \Gamma_{Y_2} \end{bmatrix} = \begin{bmatrix} \Gamma_X & B & A & D \\ B^T & \Gamma_Y & D & C \\ A^T & D^T & \Gamma_X & B \\ D^T & C^T & B^T & \Gamma_Y \end{bmatrix}, \tag{2.2.5}$$

The second equality being due to $\Gamma_{X_1} = \Gamma_{X_2}$, $\Gamma_{Y_1} = \Gamma_{Y_2}$, $\Gamma_{X_1 Y_2} = \Gamma_{X_2 Y_1}$, and $\Gamma_{X_1 Y_1} = \Gamma_{X_2 Y_2}$. Then condition (C2) in Proposition 2.1, being equivalent to the independence of X_1 and Y_2 conditionally on X_2, is written

$$D = A \Gamma_X^{-1} B, \tag{2.2.6}$$

and thus X_1^N is not Markov for $D \neq A \Gamma_X^{-1} B$.

For example, taking X_1^N and Y_1^N real with means null and variances equal to one, we have $\Gamma = \begin{bmatrix} 1 & b & a & d \\ b & 1 & d & c \\ a & d & 1 & b \\ d & c & b & 1 \end{bmatrix}$. Such a PMM is a PMM-IN iff $c = ab^2$, it is a HMM-CN iff $d = ab$, and it is a classic HMM-IN iff $c = ab^2$ and $d = ab$. We see that PMM is defined with four parameters while HMM-IN is defined with only two parameters.

Kalman filtering is workable in general Gaussian PMMs [49], and parameters can be estimated in homogeneous case with stochastic gradient [32] or Expectation-Maximization (EM) algorithm [42].

2.3 Triplet Markov models

Let X_1^N, Y_1^N, and $Z_1^N = (Z_1,...,Z_N)$, with $Z_n = (X_n, Y_n)$ for $n = 1, \ldots, N$, be random sequences as above. As above, X_1^N and Y_1^N are assumed to present some concrete phenomenon. Let $R_1^N = (R_1,...,R_N)$ be a third random sequence, which can have some concrete signification or not. Let us assume that $T_1^N = (T_1,...,T_N)$, with $T_n = (X_n, R_n, Y_n)$ for $n = 1, \ldots, N$, is Markov. Such a model will be called "triplet Markov model" (TMM).

Setting $T_1^N = (S_1^N, Y_1^N)$, with $S_n = (X_n, R_n)$ for $n = 1, \ldots, N$, we see that $T_1^N = (S_1^N, Y_1^N)$ is a PMM. Thus $S_1^N = (X_1^N, R_1^N)$ can be searched from Y_1^N in some situations, which gives both X_1^N and R_1^N.

As there are no constraints in choosing R_1^N, TMMs form an extremely wide family of models. This family is strictly richer than PMMs; indeed, $Z_1^N = (X_1^N, Y_1^N)$ is not necessarily Markov in general TMM. More precisely, no one of the six processes X_1^N, Y_1^N, R_1^N, (X_1^N, Y_1^N), (X_1^N, R_1^N), (R_1^N, Y_1^N) is necessarily Markov. Of course, X_1^N is Markov conditionally on (R_1^N, Y_1^N), Y_1^N is Markov conditionally on (X_1^N, R_1^N), R_1^N is Markov conditionally on (X_1^N, Y_1^N), (X_1^N, Y_1^N) is Markov conditionally on R_1^N, (X_1^N, R_1^N) is Markov conditionally on Y_1^N, and (R_1^N, Y_1^N) is Markov conditionally on X_1^N. These six conditional Markovianity provide different interpretations and different possibilities of processing of interest. To comment some of them let us return to Case (i) and Case (ii) of subsection 2.1.

Case (i): X_1^N is discrete finite and Y_1^N is either discrete finite or continuous (real or multidimensional). If R_1^N is discrete finite the problem of searching X_1^N is similar to the problem of searching X_1^N in the PMMs case. Indeed, setting $S_1^N = (X_1^N, R_1^N)$, $T_1^N = (S_1^N, Y_1^N)$ is a PMM and searching $S_1^N = (X_1^N, R_1^N)$ is workable. Such a TMM can be interpreted as a switching system, so that one can simultaneously find classes $X_1^N = x_1^N$ and switches $R_1^N = r_1^N$.

Case (ii): (X_1^N, Y_1^N) is multidimensional Gaussian. If R_1^N also is multidimensional Gaussian the problem of searching $S_1^N = (X_1^N, R_1^N)$ in TMMs is similar to the problem of searching X_1^N in the PMMs case. Indeed, likely to the previous case, $T_1^N = (S_1^N, Y_1^N)$ is a multidimensional Gaussian PMM and searching $S_1^N = (X_1^N, R_1^N)$ is workable, while (X_1^N, Y_1^N) is not necessarily Markov. For example, Kalman Filtering (KF) is workable, and thus there exist models (X_1^N, Y_1^N) which are not Markov and in which fast exact KF is feasible [3, 15].

Thus we can say that when the third added process R_1^N is of the same nature than X_1^N (both discrete finite or both continuous Gaussian) the processing problem can be solved in a similar manner than in PMM (X_1^N, Y_1^N).

In the following we deal with the case where R_1^N and X_1^N are of different nature: R_1^N is discrete finite and X_1^N is multidimensional. More precisely, R_1^N is discrete finite and (X_1^N, Y_1^N) is multidimensional Gaussian conditionally on R_1^N. In such a switching Gaussian case fast exact processing – filtering or smoothing – are no longer possible in general. However, there exist a sub-family of these models in which they are, and thus once a given triplet model has been approximated by a model in the latter family one can propose an approximate fast processing, which is an exact one in an approximate model. As specified in Remark 2.2 below such sub-families are obtained relaxing the Markovianity of $S_1^N = (X_1^N, R_1^N)$.

Remark 2.2

Proposed in [46] in a general context, some particular TMMs have been known before. For example, hidden semi-Markov models can be seen as particular TMMs. As specified in Introduction, another example are SLDSs, where the third process R_1^N models switches of the distribution of the couple $Z_1^N = (X_1^N, Y_1^N)$. However, apart from [46] and subsequently developed models, all classic models are of "hierarchical" kind: the hidden couple $S_1^N = (X_1^N, R_1^N)$ is assumed Markov, and the Markovianity of S_1^N conditional on Y_1^N, needed for processing, is ensured by taking $p(y_1^N | r_1^N, x_1^N)$ simple enough. For example, according to Proposition 2.1, in stationary invertible case assuming $S_1^N = (X_1^N, R_1^N)$ Markov is equivalent to assuming $p(y_n | r_1^N, x_1^N) = p(y_n | r_n, x_n)$ for each $n = 1, \ldots, N$, which can appear as a quite strong hypothesis in some real applications. Similarly to what have been mentioned for PMMs, giving up the hierarchical way of defining the distribution $p(r_1^N, x_1^N, y_1^N)$, opens ways for considering more complex Markovian noises $p(y_1^N | r_1^N, x_1^N)$. This also lead to propose CMSHLMs and CGOMSMs allowing fast exact filtering in spite of the presence of unknown switches [1, 2, 48].

3. FAST EXACT FILTERING IN SWITCHING MODELS

Let us consider three random sequences X_1^N, R_1^N and Y_1^N as specified in introduction. The new fast smoothing we will specify in the next section is based on two fast filters: one of a left-right kind, and another of a right-left kind. It will be proposed in the next section in the frame of "stationary in law" models specified below. This section is devoted to recall the related filters and models in which they run.

Let us consider the following "conditionally Markov switching hidden linear model" (CMSHLM), slightly more general that CMSHLM proposed in [48].

Definition 3.1

The triplet $T_1^N = (X_1^N, R_1^N, Y_1^N)$ is called "conditionally Markov switching hidden linear model" (CMSHLM) if:

$T_1^N = (X_1^N, R_1^N, Y_1^N)$ is Markov; (3.1.1)

for $n = 1, \ldots, N-1$:

$p(r_{n+1}, y_{n+1} | x_n, r_n, y_n) = p(r_{n+1}, y_{n+1} | r_n, y_n)$ ($Q_1^N = (R_1^N, Y_1^N)$ is then Markov); (3.1.2)

$$X_{n+1} = G_{n+1}(R_n^{n+1}, Y_n^{n+1}) X_n + H_{n+1}(R_n^{n+1}, Y_n^{n+1}, U_{n+1}). \quad (3.1.3)$$

with U_1, …, U_N some sequence of random variables.

Then we have the following

Proposition 3.1

Let $T_1^N = (X_1^N, R_1^N, Y_1^N)$ be a CMSHLM. Let $E[H_{n+1} | R_n^{n+1}, Y_n^{n+1}] = M_{n+1}(R_n^{n+1}, Y_n^{n+1})$. Then

(i) $p(r_{n+1} | y_1^{n+1})$ is given from $p(r_{n+1}, y_{n+1} | r_n, y_n)$ and $p(r_n | y_1^n)$ with (3.1.5), (3.1.6);

(ii) $E[X_{n+1} | r_{n+1}, y_1^{n+1}]$ is given from $G_{n+1}(r_n^{n+1}, y_n^{n+1})$, $M_{n+1}(r_n^{n+1}, y_n^{n+1})$, and $E[X_n | r_n, y_1^n]$ with

$$E[X_{n+1} | r_{n+1}, y_1^{n+1}] = \sum_{r_n} p(r_n | r_{n+1}, y_1^{n+1})[A_{n+1}(r_n^{n+1}, y_n^{n+1}) E[X_n | r_n, y_1^n] + M_{n+1}(r_n^{n+1}, y_n^{n+1})], \quad (3.14)$$

with $p(r_n | r_{n+1}, y_1^{n+1})$ computed from (3.1.5)

Proof

We wish to compute $E[X_{n+1} | r_{n+1}, y_1^{n+1}]$ and $p(x_{n+1} | r_1^{n+1}]$ from $E[X_n | r_n, y_1^n]$ and $p(x_n | r_1^n]$.

As $Q_1^N = (R_1^N, Y_1^N)$ is Markov, $p(r_{n+1} | y_1^{n+1})$ and $p(r_n, | r_{n+1}, y_1^{n+1})$ are computable from

$$p(r_n, r_{n+1} \big| y_1^{n+1}) = \frac{p(r_n, r_{n+1}, y_{n+1} \big| y_1^n)}{p(y_{n+1} \big| y_1^n)} = \frac{p(r_{n+1}, y_{n+1} | r_n, y_n)}{p(y_{n+1} \big| y_1^n)} p(r_n \big| y_1^n) ; \tag{3.1.5}$$

$$p(r_{n+1} \big| y_1^{n+1}) = \sum_{r_n} p(r_n, r_{n+1} \big| y_1^{n+1}) . \tag{3.1.6}$$

Besides, taking conditional expectation of (3.1.3) we have $E[X_{n+1} \big| r_n^{n+1}, y_1^{n+1}] =$ $A_{n+1}(r_n^{n+1}, y_n^{n+1}) E[X_n \big| r_n^{n+1}, y_1^{n+1}] + M_{n+1}(r_n^{n+1}, y_n^{n+1})$. As (3.1.2) implies $E[X_n \big| r_n^{n+1}, y_1^{n+1}] = E[X_n \big| r_n, y_1^n]$, we have

$$E[X_{n+1} \big| r_n^{n+1}, y_1^{n+1}] = A_{n+1}(r_n^{n+1}, y_n^{n+1}) E[X_n \big| r_n, y_1^n] + M_{n+1}(r_n^{n+1}, y_n^{n+1}) , \tag{3.1.7}$$

which leads to (3.1.4) using $E[X_{n+1} \big| r_{n+1}, y_1^{n+1}] = \sum_{r_n} p(r_n \big| r_{n+1}, y_1^{n+1}) E[X_{n+1} \big| r_n^{n+1}, y_1^{n+1}]$, and ends the proof.

The exact fast smoothing method we propose will be applicable in a subfamily of the following family of models:

Definition 3.2

The triplet $T_1^N = (X_1^N, R_1^N, Y_1^N)$ is called "stationary in law conditionally Gaussian pairwise Markov switching model" (SL-CGPMSM) if:

$$T_1^N = (X_1^N, R_1^N, Y_1^N) \text{ is Markov;} \tag{3.1.8}$$

$$p(r_{n+1} | x_n, r_n, y_n) = p(r_{n+1} | r_n) \text{ for each } n = 1, \ldots, N-1; \tag{3.1.9}$$

$$p(x_n, y_n, x_{n+1}, y_{n+1} | r_n, r_{n+1}) \text{ are Gaussian for each } n = 1, \ldots, N-1; \tag{3.1.10}$$

Let us remark that in SL-CGPMSM we also have, for $n = 1, \ldots, N-1$:

$$p(x_n, y_n, | r_n, r_{n+1}) = p(x_n, y_n, | r_n), \; p(x_{n+1}, y_{n+1} | r_n, r_{n+1}) = p(x_{n+1}, y_{n+1} | r_{n+1}) . \tag{3.1.11}$$

Indeed, the first equality in (3.1.11) comes directly from (3.1.9), and the second equality comes from the fact that if $p(x_{n+1}, y_{n+1} | r_n, r_{n+1})$ was depending on r_n, the distribution $p(x_{n+1}, y_{n+1} | r_{n+1})$ would be a Gaussian mixture, which is not the case because of the first one.

Let us notice that "stationarity in law" does not imply stationarity: in SL-CGPMSM distributions of $T_n = (X_n, R_n, Y_n)$ can vary with n. "Stationarity in law" means that the form of distributions of T_n does not vary: here $p(x_n, y_n | r_n)$ are Gaussain for any $n = 1, \ldots, N$.

Let $T_1^N = (X_1^N, R_1^N, Y_1^N)$ be a SL-CGPMSM. To simplify writing, let us assume that means of all Gaussian distributions $p(x_n, y_n, x_{n+1}, y_{n+1} | r_n, r_{n+1})$ are null. Distribution of T_1^N is then defined by the distribution of R_1^N and the covariance matrices

$$\Gamma_{n,n+1}(r_1^2) = E\left[\begin{bmatrix} X_n \\ Y_n \\ X_{n+1} \\ Y_{n+1} \end{bmatrix} \begin{bmatrix} X_n^T & Y_n^T & X_{n+1}^T & Y_{n+1}^T \end{bmatrix} \middle| R_n^{n+1} = r_n^{n+1}\right] = \tag{3.1.12}$$

$$\begin{bmatrix} \Gamma_{X_n}(r_n) & \Gamma_{X_nY_n}(r_n) & \Gamma_{X_nX_{n+1}}(r_n^{n+1}) & \Gamma_{X_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_nX_n}(r_n) & \Gamma_{Y_n}(r_n) & \Gamma_{Y_nX_{n+1}}(r_n^{n+1}) & \Gamma_{Y_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{X_{n+1}X_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{X_{n+1}}(r_{n+1}) & \Gamma_{X_{n+1}Y_{n+1}}(r_{n+1}) \\ \Gamma_{Y_{n+1}X_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}X_{n+1}}(r_{n+1}) & \Gamma_{Y_{n+1}}(r_{n+1}) \end{bmatrix} = \begin{bmatrix} \Gamma_{Z_n}(r_n) & \Gamma_{Z_nZ_{n+1}}(r_n^{n+1}) \\ \Gamma_{Z_{n+1}Z_n}(r_n^{n+1}) & \Gamma_{Z_{n+1}}(r_{n+1}) \end{bmatrix}$$

with $Z_n = \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$ for $n = 1, \ldots, N$. According to (3.1.8)-(3.1.10) we can write for $n = 1, \ldots, N-1$:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} a_{n+1}^1(r_n^{n+1}) & a_{n+1}^2(r_n^{n+1}) \\ a_{n+1}^3(r_n^{n+1}) & a_{n+1}^4(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} b_{n+1}^1(r_n^{n+1}) & b_{n+1}^2(r_n^{n+1}) \\ b_{n+1}^3(r_n^{n+1}) & b_{n+1}^4(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}, \tag{3.1.13}$$

where U_2, $V_2 \ldots$, U_N, V_N are standard Gaussian white noise variable: means are null and variance-covariance matrices are identity.

Setting $A_{n+1}(r_n^{n+1}) = \begin{bmatrix} a_{n+1}^1(r_n^{n+1}) & a_{n+1}^2(r_n^{n+1}) \\ a_{n+1}^3(r_n^{n+1}) & a_{n+1}^4(r_n^{n+1}) \end{bmatrix}$, $B_{n+1}(r_n^{n+1}) = \begin{bmatrix} b_{n+1}^1(r_n^{n+1}) & b_{n+1}^2(r_n^{n+1}) \\ b_{n+1}^3(r_n^{n+1}) & b_{n+1}^4(r_n^{n+1}) \end{bmatrix}$, and $W_{n+1} = \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}$, (3.1.13) will also be sometimes written in a concise form

$$Z_{n+1} = A_{n+1}(r_n^{n+1}) Z_n + B_{n+1}(r_n^{n+1}) W_{n+1}. \tag{3.1.14}$$

$A_{n+1}(r_n^{n+1})$ and $B_{n+1}(r_n^{n+1})$ in (3.1.13), (3.1.14) are classically obtained from (3.1.12) with

$$A_{n+1}(r_n^{n+1}) = \Gamma_{Z_{n+1}Z_n}(r_n^{n+1})\Gamma_{Z_n}^{-1}(r_n)\,. \tag{3.1.15}$$

$$B_{n+1}(r_n^{n+1})B_{n+1}^T(r_n^{n+1}) = \Gamma_{Z_{n+1}}(r_{n+1}) - \Gamma_{Z_{n+1}Z_n}(r_n^{n+1})\Gamma_{Z_n}^{-1}(r_n)\Gamma_{Z_nZ_{n+1}}(r_n^{n+1}) \tag{3.1.16}$$

Let us notice that switching systems are usually presented in form (3.1.13), (3.1.14). We can equivalently define SL-CGPMSM using the following

Definition 3.3

The triplet $T_1^N = (X_1^N, R_1^N, Y_1^N)$ is called "stationary in law conditionally Gaussian pairwise Markov switching model" (SL-CGPMSM) if:

$$T_1^N = (X_1^N, R_1^N, Y_1^N) \text{ is Markov;} \tag{3.1.17}$$

$$p(r_{n+1}|x_n, r_n, y_n) = p(r_{n+1}|r_n) \text{ for } n = 1, \ldots, N-1; \tag{3.1.18}$$

Setting $Z_n = \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$, $Z_1^N = (Z_1, \ldots, Z_N)$ verifies for $n = 1, \ldots, N-1$

$$Z_{n+1} = A_{n+1}(r_n^{n+1})Z_n + B_{n+1}(r_n^{n+1})W_{n+1}\,,$$

with W_2 , …, W_N standard Gaussian white noise variables and $A_2(r_1^2)$, $B_2(r_1^2)$, …, $A_N(r_{N-1}^N)$, $B_N(r_{N-1}^N)$ such that for $n = 1, \ldots, N$

$$E\left[Z_n \middle| r_1^n\right] = E\left[Z_n | r_n\right] \text{ and } E\left[Z_n Z_n^T \middle| r_1^n\right] = E\left[Z_n Z_n^T | r_n\right] \tag{3.1.19}$$

Let us notice that (3.1.19) is equivalent to : for $n = 1, \ldots, N-1$,

$$A_{n+1}(r_n^{n+1})E\left[Z_n Z_n^T | r_n\right]A_{n+1}^T(r_n^{n+1}) + B_{n+1}(r_n^{n+1})B_{n+1}^T(r_n^{n+1}) \tag{3.1.20}$$

does not depend on r_n.

Fast exact filtering is feasible in the following particular SL-CGPMSM.

Definition 3.4

SL-CGPMSM $T_1^N = (X_1^N, R_1^N, Y_1^N)$ is called "stationary in law conditionally Gaussian observed Markov switching model" (SL-CGOMSM) if $A_{n+1}(r_n^{n+1})$ in (3.1.13), (3.1.14) is of the form

$$A_{n+1}(r_n^{n+1}) = \begin{bmatrix} a_{n+1}^1(r_n^{n+1}) & a_{n+1}^2(r_n^{n+1}) \\ 0 & a_{n+1}^4(r_n^{n+1}) \end{bmatrix}. \tag{3.1.21}$$

Let us show that SL-CGOMSM is a CMSHLM (see Definition 3.1).
We have
$p(r_{n+1}, y_{n+1}|x_n, r_n, y_n) = p(r_{i+1}|x_n, r_n, y_n)p(y_{i+1}|x_n, r_n, y_n, r_{n+1}) = p(r_{n+1}|\, r_n)p(y_{n+1}|x_n, y_n, r_n^{n+1})$.
According to (3.1.13) and (3.1.21) $Y_{n+1} = a_{n+1}^4(r_n^{n+1})Y_n + V_{n+1}^*$, with $V_{n+1}^* = b_{n+1}^3(r_n^{n+1})U_{n+1} + b_{n+1}^4(r_n^{n+1})V_{n+1}$ independent from (X_1^n, R_1^n, Y_1^n), which implies $p(y_{n+1}|x_n, y_n, r_n^{n+1}) = p(y_{n+1}|y_n, r_n^{n+1})$.
Finally

$$p(r_{n+1}, y_{n+1}|x_n, r_n, y_n) = p(r_{n+1}, y_{n+1}|\, r_n, y_n) = p(r_{n+1}|\, r_n)p(y_{n+1}|y_n, r_n^{n+1}), \tag{3.1.22}$$

This means that in SL-CGOMSM $p(r_{n+1}, y_{n+1}|x_n, r_n, y_n)$ are of the form (3.1.2) with $p(y_{n+1}|\, y_n, r_n^{n+1})$ Gaussian with mean $a_{n+1}^4(r_n^{n+1})y_n$ and variance $[b_{n+1}^3(r_n^{n+1})][b_{n+1}^3(r_n^{n+1})]^T + [b_{n+1}^4(r_n^{n+1})][b_{n+1}^4(r_n^{n+1})]^T$:

$$p(y_{n+1}|\, y_n, r_n^{n+1}) = \mathrm{N}(a_{n+1}^4(r_n^{n+1})y_n, [b_{n+1}^3(r_n^{n+1})][b_{n+1}^3(r_n^{n+1})]^T + [b_{n+1}^4(r_n^{n+1})][b_{n+1}^4(r_n^{n+1})]^T) \tag{3.1.23}$$

Finally $p(r_{n+1}, y_{n+1}|\, r_n, y_n)$ used in computation in fast filter in CMSHLM is given in SL-CGOMSM with (3.1.22) and (3.1.23).

Let us verify (3.1.3) - also see Remark 3.1. According to (3.1.13) - with (3.1.21) - $p(x_{n+1}, y_{n+1}|x_n, y_n, r_n^{n+1})$ is Gaussian with mean and variance

$$M = \begin{bmatrix} a_{n+1}^1(r_n^{n+1})x_n + a_{n+1}^2(r_n^{n+1})y_n \\ a_{n+1}^4(r_n^{n+1})y_n \end{bmatrix}; \tag{3.1.24}$$

$$\Gamma = \begin{bmatrix} \gamma_{n+1}^1(r_n^{n+1}) & \gamma_{n+1}^2(r_n^{n+1}) \\ \gamma_{n+1}^3(r_n^{n+1}) & \gamma_{n+1}^4(r_n^{n+1}) \end{bmatrix} = \begin{bmatrix} b_{n+1}^1(r_n^{n+1}) & b_{n+1}^2(r_n^{n+1}) \\ b_{n+1}^3(r_n^{n+1}) & b_{n+1}^4(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} b_{n+1}^1(r_n^{n+1}) & b_{n+1}^2(r_n^{n+1}) \\ b_{n+1}^3(r_n^{n+1}) & b_{n+1}^4(r_n^{n+1}) \end{bmatrix}^T, \tag{3.1.25}$$

and thus according to Gaussian conditioning rules $p(x_{n+1}|x_n, y_n, y_{n+1}, r_n^{n+1})$ is Gaussian with mean and variance

$$M^* = a_{n+1}^1(r_n^{n+1})x_n + a_{n+1}^2(r_n^{n+1})y_n + \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}(y_{n+1} - a_{n+1}^4(r_n^{n+1})y_n)\,; \tag{3.1.26}$$

$$\Gamma^* = \gamma_{n+1}^1(r_n^{n+1}) - \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}\gamma_{n+1}^3(r_n^{n+1})\,. \tag{3.1.27}$$

Then

$$\begin{aligned} X_{n+1} &= a_{n+1}^1(r_n^{n+1})X_n + \left[a_{n+1}^2(r_n^{n+1}) - \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}a_{n+1}^4(r_n^{n+1})\right]Y_n + \\ &+ \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}Y_{n+1} + Q_{n+1}(r_n^{n+1})U_{n+1} \end{aligned} \tag{3.1.28}$$

with

$$Q_{n+1}(r_n^{n+1})Q_{n+1}^T(r_n^{n+1}) = \Gamma^*\,. \tag{3.1.29}$$

We see that X_{n+1} is of the form $X_{n+1} = G_{n+1}(R_n^{n+1}, Y_n^{n+1})X_n + H_{n+1}(R_n^{n+1}, Y_n^{n+1})$ - see (3.1.3) - with

$$G_{n+1}(r_n^{n+1}, y_n^{n+1}) = a_{n+1}^1(r_n^{n+1})\,; \tag{3.1.30}$$

$$\begin{aligned} H_{n+1}(r_n^{n+1}, y_n^{n+1}, U_{n+1}) &= \left[a_{n+1}^2(r_n^{n+1}) - \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}a_{n+1}^4(r_n^{n+1})\right]y_n + \\ &+ \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}y_{n+1} + Q_{n+1}(r_n^{n+1})U_{n+1}, \end{aligned} \tag{3.1.31}$$

and

$$\begin{aligned} M_{n+1}(r_n^{n+1}, y_n^{n+1}) &= \left[a_{n+1}^2(r_n^{n+1}) - \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}a_{n+1}^4(r_n^{n+1})\right]y_n + \\ &+ \gamma_{n+1}^2(r_n^{n+1})\left[\gamma_{n+1}^4(r_n^{n+1})\right]^{-1}y_{n+1} \end{aligned} \tag{3.1.32}$$

Finally, fast exact filter in SL-CGOMSM is given by (3.1.5), (3.1.6), (3.14), with $p(r_{n+1}, y_{n+1}|\, r_n, y_n)$ given by (3.1.22)-(3.1.23), and G_{n+1}, H_{n+1}, and M_{n+1}, given by (3.1.30)-(3.1.32).

Remark 3.1

Formulas (3.1.28)-(3.1.32) use matrices $A_{n+1}(r_n^{n+1})$, $B_{n+1}(r_n^{n+1})$, and thus they are of interest when the model is defined with (3.1.13), (3.1.14). When it is defined with matrices (3.1.12) the filter can be obtained straightforwardly from these matrices, which would be possibly simpler to program. According to (3.1.12) covariance matrix of $\begin{bmatrix} X_n \\ Y_n \\ Y_{n+1} \\ X_{n+1} \end{bmatrix}$ (X_{n+1} and Y_{n+1} have been inverted) is

$$\begin{bmatrix} \Gamma_{X_n}(r_n) & \Gamma_{X_nY_n}(r_n) & \Gamma_{X_nY_{n+1}}(r_n^{n+1}) & \Gamma_{X_nX_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_nX_n}(r_n) & \Gamma_{Y_n}(r_n) & \Gamma_{Y_nY_{n+1}}(r_n^{n+1}) & \Gamma_{Y_nX_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_{n+1}X_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}Y_n}(r_n^{n+1}) & \Gamma_Y(r_{n+1}) & \Gamma_{YX}(r_{n+1}) \\ \Gamma_{X_{n+1}X_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{XY}(r_{n+1}) & \Gamma_X(r_{n+1}) \end{bmatrix} \tag{3.1.33}$$

Thus $M*$ and $\Gamma*$ given with (3.1.26) and (3.1.27) are also given with

$$M* = \begin{bmatrix} \Gamma_{X_{n+1}X_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_{n+1}}(r_{n+1}) \end{bmatrix} \begin{bmatrix} \Gamma_{X_n}(r_n) & \Gamma_{X_nY_n}(r_n) & \Gamma_{X_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_nX_n}(r_n) & \Gamma_{Y_n}(r_n) & \Gamma_{Y_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_{n+1}X_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}}(r_{n+1}) \end{bmatrix}^{-1} \begin{bmatrix} x_n \\ y_n \\ y_{n+1} \end{bmatrix} \tag{3.1.34}$$

and

$$\Gamma* = \Gamma_{X_{n+1}}(r_{n+1}) - \begin{bmatrix} \Gamma_{X_{n+1}X_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{X_{n+1}Y_{n+1}}(r_{n+1}) \end{bmatrix} \begin{bmatrix} \Gamma_{X_n}(r_n) & \Gamma_{X_nY_n}(r_n) & \Gamma_{X_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_nX_n}(r_n) & \Gamma_{Y_n}(r_n) & \Gamma_{Y_nY_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_{n+1}X_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}Y_n}(r_n^{n+1}) & \Gamma_{Y_{n+1}}(r_{n+1}) \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_{X_nX_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_nX_{n+1}}(r_n^{n+1}) \\ \Gamma_{Y_{n+1}X_{n+1}}(r_{n+1}) \end{bmatrix} \tag{3.1.35}$$

Remark 3.2

Classic switching Gaussian model $T_1^N = (X_1^N, R_1^N, Y_1^N)$, called "Conditionally Gaussian Linear State-Space model" (CGLSSM), is defined as follows:

R_1^N is Markov; (3.1.36)

$$X_{n+1} = C_{n+1}(r_{n+1})X_n + D_{n+1}(r_{n+1})U_{n+1} \quad (3.1.37)$$

$$Y_{n+1} = E_{n+1}(r_{n+1})X_{n+1} + F_{n+1}(r_{n+1})V_{n+1} \quad (3.1.38)$$

Reporting (3.1.28) into (3.1.29) we see that such a model verifies particular (3.1.13) equation:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} C_{n+1}(r_{n+1}) & 0 \\ E_{n+1}(r_{n+1})C_{n+1}(r_{n+1}) & 0 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} D_{n+1}(r_{n+1}) & 0 \\ E_{n+1}(r_{n+1})D_{n+1}(r_{n+1}) & F_{n+1}(r_{n+1}) \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}. \quad (3.1.39)$$

Let us notice that CGLSSM cannot be a SL-CGPMSM, even in homogeneous – i.e. when $C_{n+1} = C$, $D_{n+1} = D$, $E_{n+1} = E$, and $F_{n+1} = F$ - case. Indeed, distributions $p(x_n, y_n | r_n)$ are Gaussian mixtures with number of components increasing with n. This property is often put forth to state that fast exact filtering cannot be perform in CGLSSMs.

Remark 3.3

We believe that SL-CGPMSM is better suited to real situations than classic switching models - in which margins become richer and richer mixtures - because of the independence of the form of margins and transition from n. For example, suppose there are two observers of a phenomenon studied – as tracking some flying target – and both of them use CGLSSM. The first starts at $n = 1$, and the second one starts at $n = 5$. The problem is that they will not use the same phenomenon probability distribution for $n = 5$, while they would when using SL-CGPMSM.

4. FAST APPROXIMATE SMOOTHING IN SWITCHING MODELS

4.1 n-SL-CGOMSM approximations of a SL-CGPMSM

Let $T_1^N = (X_1^N, R_1^N, Y_1^N)$ be a SL-CGPMSM, with distribution defined by $\Gamma_{1,2}(r_1^2)$, … , $\Gamma_{n-1,n}(r_{n-1}^n)$, $\Gamma_{n,n+1}(r_n^{n+1})$, … , $\Gamma_{N-1,N}(r_{N-1}^N)$. The problem we deal with is to compute $E\left[X_n \middle| r_n, y_1^N\right]$, $E\left[X_n^2 \middle| r_n, y_1^N\right]$, and $p(r_n \middle| y_1^N)$ for each $n = 1, \ldots, N$. The idea is to approximate the distribution $p(t_1^N) = p(x_1^N, r_1^N, y_1^N)$ by the following "n-SL-CGOMSM" distribution. The distribution $p(t_1^n) = p(x_1^n, r_1^n, y_1^n)$ of $T_1^n = (T_1, \ldots, T_n)$ is classically replaced by the following SL-CGOMSM's distribution $p^*(t_1^n) = p^*(x_1^n, r_1^n, y_1^n)$: matrices $\Gamma_{1,2}(r_1^2)$, .., $\Gamma_{n-1,n}(r_1^2)$ are modified by replacing in each

$\Gamma_{k,k+1}(r_1^2)$, for $k=1, \ldots, n-1$, $\Gamma_{X_kY_{k+1}}(r_k^{k+1})$ by $\Gamma^*_{X_kY_{k+1}}(r_n^{n+1}) = \Gamma_{X_kY_k}(r_k^{k+1})\Gamma^{-1}_{Y_k}(r_k)\Gamma_{Y_kY_{k+1}}(r_k^{k+1})$. Then

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} a^{1*}_{n+1}(r_n^{n+1}) & a^{2*}_{n+1}(r_n^{n+1}) \\ 0 & a^{4*}_{n+1}(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} b^{1*}_{n+1}(r_n^{n+1}) & b^{2*}_{n+1}(r_n^{n+1}) \\ b^{3*}_{n+1}(r_n^{n+1}) & b^{4*}_{n+1}(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}, \tag{4.1.1}$$

so that $p^*(x_1^n, r_1^n, y_1^n)$ indeed is a CGOMSM distribution.

Remark 4.1

Let us notice that $p^*(t_1^n)$ and $p(t_1^n)$ are close each to another in that:

- $p(r_1^n)$ and $p^*(r_1^n)$ are identical;

- for $k=1, \ldots, n-1$, $\Gamma_{k,k+1}(r_1^2)$ and $\Gamma^*_{k,k+1}(r_1^2)$ are defined with ten different covariance matrices: nine are the same and just one is different. In particular, margins $p(t_k)$ and $p^*(t_k)$ are identical;

- one can see that $p(x_k|y_k, y_{k-1}, y_{k-2})$ and $p^*(x_k|y_k, y_{k-1}, y_{k-2})$ are identical for $k=3, \ldots, n$.

Due to that filtering data sampled with $p(t_1^n)$ using the fast filter specified in section 3 based on $p^*(t_1^n)$ gives results fairly as good as optimal results obtained with particle filter based on the true distribution $p(t_1^n)$, which is significantly more time consuming.

The distribution of $T_n^N = (T_n, \ldots, T_N)$ is replaced by the following SL-CGOMSM's distribution. First, one inverts time: the distribution of $T_n^N = (T_n, \ldots, T_N)$ is also defined by the distributions $p(t_N)$, $p(t_{N-1}|t_N)$, $\ldots$, $p(t_n|t_{n+1})$. Each $p(t_{k-1}|t_k)$, for $k=n+1, \ldots, N$, is written $p(t_{k-1}|t_k) = p(r_{k-1}|r_k)\, p(x_{k-1}, y_{k-1}|x_k, y_k, r_{k-1}^k)$, where $p(r_{k-1}|r_k)$ is obtained from $p(r_{k-1}, r_k)$ and $p(x_{k-1}, y_{k-1}|x_k, y_k, r_{k-1}^k)$ are defined by

$$\begin{bmatrix} X_{k-1} \\ Y_{k-1} \end{bmatrix} = \begin{bmatrix} c^1_{k-1}(r_{k-1}^k) & c^2_{k-1}(r_{k-1}^k) \\ c^3_{k-1}(r_{k-1}^k) & c^4_{k-1}(r_{k-1}^k) \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} d^1_{k-1}(r_{k-1}^k) & d^2_{k-1}(r_{k-1}^k) \\ d^3_{k-1}(r_{k-1}^k) & d^4_{k-1}(r_{k-1}^k) \end{bmatrix} \begin{bmatrix} U_k \\ V_k \end{bmatrix}, \tag{4.1.2}$$

with $C_{k-1}(r_{k-1}^k)=\begin{bmatrix} c_{k-1}^1(r_{k-1}^k) & c_{k-1}^2(r_{k-1}^k) \\ c_{k-1}^3(r_{k-1}^k) & c_{k-1}^4(r_{k-1}^k) \end{bmatrix}$ and $D_{k-1}(r_{k-1}^k)=\begin{bmatrix} d_{k-1}^1(r_{k-1}^k) & d_{k-1}^2(r_{k-1}^k) \\ d_{k-1}^3(r_{k-1}^k) & d_{k-1}^4(r_{k-1}^k) \end{bmatrix}$ given from $\Gamma_{k-1,k}(r_{k-1}^k)=\begin{bmatrix} \Gamma_{Z_{k-1}}(r_{k-1}) & \Gamma_{Z_{k-1}Z_k}(r_{k-1}^k) \\ \Gamma_{Z_kZ_{k-1}}(r_{k-1}^k) & \Gamma_{Z_k}(r_k) \end{bmatrix}$ (recall that $Z_k=\begin{bmatrix} X_k \\ Y_k \end{bmatrix}$) with

$$C_{k-1}(r_{k-1}^k)=\Gamma_{Z_{k-1}}(r_{k-1})\Gamma_{Z_k}^{-1}(r_k)\,, \tag{4.1.3}$$

$$D_{k-1}(r_{k-1}^k)D_{k-1}^T(r_{k-1}^k)=\Gamma_{Z_{k-1}}(r_{k-1})-\Gamma_{Z_{k-1}Z_k}(r_{k-1}^k)\Gamma_{Z_k}^{-1}(r_k)\Gamma_{Z_kZ_{k-1}}(r_{k-1}^k)\,. \tag{4.1.4}$$

Then, likely to the modification above, the distribution of $T_n^N=(T_n,\ldots,T_N)$ is replaced by the following SL-CGOMSM's distribution $p*(t_n^N)=p*(x_n^N,r_n^N,y_n^N)$: matrices $\Gamma_{n,n+1}(r_n^{n+1})$, .., $\Gamma_{N-1,N}(r_{N-1}^N)$ are modified by replacing in each $\Gamma_{k,k+1}(r_1^2)$, for $k=n$, …, $N-1$, $\Gamma_{Y_kX_{k+1}}(r_k^{k+1})$ by $\Gamma_{Y_kX_{k+1}}^*(r_n^{n+1})=\Gamma_{Y_kY_{k+1}}(r_k^{k+1})\Gamma_{Y_{k+1}}^{-1}(r_k)\Gamma_{Y_{k+1}X_{k+1}}(r_k^{k+1})$. Then $p*(x_{k-1},y_{k-1}|x_k,y_k,r_{k-1}^k)$ are defined by

$$\begin{bmatrix} X_{k-1} \\ Y_{k-1} \end{bmatrix}=\begin{bmatrix} c_{k-1}^{1*}(r_{k-1}^k) & c_{k-1}^{2*}(r_{k-1}^k) \\ 0 & c_{k-1}^{4*}(r_{k-1}^k) \end{bmatrix}\begin{bmatrix} X_k \\ Y_k \end{bmatrix}+\begin{bmatrix} d_{k-1}^{1*}(r_{k-1}^k) & d_{k-1}^{2*}(r_{k-1}^k) \\ d_{k-1}^{3*}(r_{k-1}^k) & d_{k-1}^{4*}(r_{k-1}^k) \end{bmatrix}\begin{bmatrix} U_k \\ V_k \end{bmatrix}, \tag{4.1.5}$$

and $T_n^{N*}=(T_1^*,\ldots,T_{N-n}^*)$, with $T_1^*=T_N$, $T_2^*=T_{N-1}$, …, $T_{N-n+1}^*=T_n$ is a CGOMSM.

Important is that $E*\left[X_n\middle|r_n,y_1^n\right]$ and $E*\left[X_n\middle|r_n,y_n^N\right]$ are computable by fast filters. In addition, these computations are independent each from another.

We will show in the subsection that

$$E*\left[X_n\middle|r_n,y_1^N\right]=E*\left[X_n\middle|r_n,y_1^n\right]+E*\left[X_n\middle|r_n,y_n^N\right]-E*\left[X_n\middle|r_n,y_n\right], \tag{4.1.6}$$

which will solve the problem of fast smoothing.

4.2 Fast smoothing in n-SL-CGOMSMs

Let us show the following

Proposition 4.1

Let $T_1^N = (X_1^N, R_1^N, Y_1^N)$ be a SL-CGPMSM. Then

$$E\left[X_n \middle| r_n, y_1^N\right] = E\left[X_n \middle| r_n, y_1^n\right] + E\left[X_n \middle| r_n, y_n^N\right] - E\left[X_n \middle| r_n, y_n\right]. \tag{4.2.1}$$

Proof
We will use the following property of Gaussian distributions:

Lemma 4.1

Let A, B, C be Gaussian vectors such that B and C are independent. Then

$$E\left[\mathrm{A} \middle| (\mathrm{B},\mathrm{C})\right] = E\left[\mathrm{A} \middle| \mathrm{B}\right] + E\left[\mathrm{A} \middle| \mathrm{C}\right] - E\left[\mathrm{A}\right] \tag{4.2.2}$$

Proof
Let $\mathrm{M_A}$, $\mathrm{M_B}$, $\mathrm{M_C}$, Γ_A, Γ_B, Γ_C, be means and covariance matrices of A, B, C, and let $\Gamma_{(\mathrm{B},\mathrm{C})}$ be the covariance matrix of $\begin{pmatrix}\mathrm{B}\\ \mathrm{C}\end{pmatrix}$, $\Gamma_{\mathrm{A},\mathrm{B}} = E\left[(\mathrm{A}-\mathrm{M_A})(\mathrm{B}^T - \mathrm{M_B^T})\right]$, $\Gamma_{\mathrm{A},\mathrm{C}} = E\left[(\mathrm{A}-\mathrm{M_A})(\mathrm{C}^T - \mathrm{M_C^T})\right]$, and $\Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})} = E\left[(\mathrm{A}-\mathrm{M_A})(\mathrm{B}^T - \mathrm{M_B^T}, \mathrm{C}^T - \mathrm{M_C^T})\right] = \left[E\left[(\mathrm{A}-\mathrm{M_A})(\mathrm{B}^T - \mathrm{M_B^T})\right], E(\mathrm{C}^T - \mathrm{M_C^T})(\mathrm{A}-\mathrm{M_A})\right] = (\Gamma_{\mathrm{A},\mathrm{B}}, \Gamma_{\mathrm{A},\mathrm{C}})$. As $\Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})} = (\Gamma_{\mathrm{A},\mathrm{B}}, \Gamma_{\mathrm{A},\mathrm{C}})$, $\Gamma_{(\mathrm{B},\mathrm{C})} = \begin{bmatrix}\Gamma_\mathrm{B} & 0\\ 0 & \Gamma_\mathrm{C}\end{bmatrix}$ and $\Gamma_{(\mathrm{B},\mathrm{C})}^{-1} = \begin{bmatrix}\Gamma_\mathrm{B}^{-1} & 0\\ 0 & \Gamma_\mathrm{C}^{-1}\end{bmatrix}$, we have

$$E\left[\mathrm{A} \middle| (\mathrm{B},\mathrm{C})\right] = \mathrm{M_A} - \Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})}\Gamma_{(\mathrm{B},\mathrm{C})}^{-1}\Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})}^T = \mathrm{M_A} - \Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})}\begin{bmatrix}\Gamma_\mathrm{B}^{-1} & 0\\ 0 & \Gamma_\mathrm{C}^{-1}\end{bmatrix}\Gamma_{\mathrm{A},(\mathrm{B},\mathrm{C})}^T =$$

$$\mathrm{M_A} - (\Gamma_{\mathrm{A},\mathrm{B}}, \Gamma_{\mathrm{A},\mathrm{C}})\begin{bmatrix}\Gamma_\mathrm{B}^{-1} & 0\\ 0 & \Gamma_\mathrm{C}^{-1}\end{bmatrix}\begin{pmatrix}\Gamma_{\mathrm{A},\mathrm{B}}^T\\ \Gamma_{\mathrm{A},\mathrm{C}}^T\end{pmatrix} = \mathrm{M_A} - \Gamma_{\mathrm{A},\mathrm{B}}\Gamma_\mathrm{B}^{-1}\Gamma_{\mathrm{A},\mathrm{B}}^T - \Gamma_{\mathrm{A},\mathrm{C}}\Gamma_\mathrm{C}^{-1}\Gamma_{\mathrm{A},\mathrm{C}}^T =$$

$$\mathrm{M_A} - \Gamma_{\mathrm{A},\mathrm{B}}\Gamma_\mathrm{B}^{-1}\Gamma_{\mathrm{A},\mathrm{B}}^T + \mathrm{M_A} - \Gamma_{\mathrm{A},\mathrm{C}}\Gamma_\mathrm{C}^{-1}\Gamma_{\mathrm{A},\mathrm{C}}^T - \mathrm{M_A} = E\left[\mathrm{A} \middle| \mathrm{B}\right] + E\left[\mathrm{A} \middle| \mathrm{C}\right] - \mathrm{M_A},$$

which ends the proof of Lemma 4.1.

Applying Lemma 4.1 to $\mathrm{A} = X_n$, $\mathrm{B} = Y_1^{n-1}$, $\mathrm{C} = Y_{n+1}^N$, and $E[\,.\,] = E\left[\,.\,\middle| r_1^N, y_n\right]$, we can write: $E\left[X_n \middle| r_1^N, y_1^N\right] = E\left[X_n \middle| r_1^N, y_n, y_1^{n-1}\right] + E\left[X_n \middle| r_n^N, y_n, y_{n+1}^N\right] - E\left[X_n \middle| r_1^N, y_n\right]$, and thus

$$E\left[X_n \middle| r_1^N, y_1^N\right] = E\left[X_n \middle| r_1^N, y_1^n\right] + E\left[X_n \middle| r_1^N, y_n^N\right] - E\left[X_n \middle| r_1^N, y_n\right] \qquad (4.2.3)$$

Besides, as $p(x_n | r_1^n, y_1^n)$ is Gaussian, $p(x_n | r_1^N, y_1^n)$ cannot depend on r_{n+1}, …, r_N (if they did $p(x_n | r_1^n, y_1^n)$ would be a mixture), so that $p(x_n | r_1^N, y_1^n) = p(x_n | r_1^n, y_1^n)$ and thus $E\left[X_n \middle| r_1^N, y_1^n\right] = E\left[X_n \middle| r_1^n, y_1^n\right]$. Similarly $E\left[X_n \middle| r_1^N, y_n^N\right] = E\left[X_n \middle| r_n^N, y_n^N\right]$ and $E\left[X_n \middle| r_1^N, y_n\right] = E\left[X_n | r_n, y_n\right]$. Finally

$$E\left[X_n \middle| r_1^N, y_1^N\right] = E\left[X_n \middle| r_1^n, y_1^n\right] + E\left[X_n \middle| r_n^N, y_n^N\right] - E\left[X_n | r_n, y_n\right] \qquad (4.2.4)$$

Besides, as (R_1^N, Y_1^N) is Markov we have

$$p(r_1^{n-1}, r_{n+1}^N | y_1^{n-1}, r_n, y_n, y_{n+1}^N) = p(r_1^{n-1} | y_1^{n-1}, r_n, y_n)\, p(r_{n+1}^N | r_n, y_n, y_{n+1}^N), \qquad (4.2.5)$$

So that, using (4.2.4) and (4.2.5) we have

$$E\left[X_n \middle| r_n, y_1^N\right] = \sum_{(r_1^{n-1}, r_{n+1}^N)} p(r_1^{n-1}, r_{n+1}^N | y_1^{n-1}, r_n, y_{n+1}^N) E\left[X_n \middle| r_1^N, y_1^N\right] =$$

$$\sum_{(r_1^{n-1}, r_{n+1}^N)} p(r_1^{n-1} | y_1^{n-1}, r_n, y_n)\, p(r_{n+1}^N | r_n, y_n, y_{n+1}^N)\left(E\left[X_n \middle| r_1^n, y_1^n\right] + E\left[X_n \middle| r_n^N, y_n^N\right] - E\left[X_n | r_n, y_n\right]\right)$$

We have for the first term of the above sum:

$$\sum_{(r_1^{n-1}, r_{n+1}^N)} p(r_1^{n-1} | y_1^{n-1}, r_n, y_n)\, p(r_{n+1}^N | r_n, y_n, y_{n+1}^N) E\left[X_n \middle| r_1^{n-1}, y_1^{n-1}, r_n, y_n\right] =$$

$$\sum_{r_1^{n-1}} p(r_1^{n-1} | y_1^{n-1}, r_n, y_n)\left[\sum_{r_{n+1}^N} p(r_{n+1}^N | r_n, y_n, y_{n+1}^N)\right] E\left[X_n \middle| r_1^{n-1}, y_1^{n-1}, r_n, y_n\right] =$$

$$\sum_{r_1^{n-1}} p(r_1^{n-1} | y_1^{n-1}, r_n, y_n) E\left[X_n \middle| r_1^{n-1}, y_1^{n-1}, r_n, y_n\right] = E\left[X_n \middle| y_1^{n-1}, r_n, y_n\right] = E\left[X_n \middle| y_1^n, r_n\right]$$

After similar calculation for the second and the third term we obtain (4.2.1), which ends the proof.

Therefore (4.2.1) is valid in general SL-CGPMSM, and thus it also is in n - SL-CGOMSM, which gives (4.1.6).

Let $T_1^N = (X_1^N, R_1^N, Y_1^N)$ be a SL-CGPMSM, with distribution defined by $\Gamma_{1,2}(r_1^2)$, … , $\Gamma_{n-1,n}(r_{n-1}^n)$, $\Gamma_{n,n+1}(r_n^{n+1})$, … , $\Gamma_{N-1,N}(r_{N-1}^N)$.

The new smoothing Double Filtering Based Smoothing (DFBS) algorithm runs as follows.

For each $n \in \{1,\ldots,N\}$:

- Modify $\Gamma_{1,2}(r_1^2)$, … , $\Gamma_{n-1,n}(r_{n-1}^n)$, to obtain $\Gamma_{1,2}^{LR}(r_1^2)$, … , $\Gamma_{n-1,n}^{LR}(r_{n-1}^n)$ defining a Left-Right CGOMSM. Use the fast exact filter to find $E^{LR}\left[X_n \middle| y_1^n, r_n\right]$;

- Modify $\Gamma_{n,n+1}(r_n^{n+1})$, … , $\Gamma_{N-1,N}(r_{N-1}^N)$, to obtain $\Gamma_{n,n+1}^{RL}(r_1^2)$, … , $\Gamma_{n-1,n}^{LL}(r_{n-1}^n)$ defining a Right-Left CGOMSM. After having reversed time use the fast exact filter to find $E^{RL}\left[X_n \middle| y_1^n, r_n\right]$;

-compute $E\left[X_n \middle| y_n, r_n\right]$;

- set

$$\hat{x}_n(r_n, y_1^N) = E^{LR}\left[X_n \middle| y_1^n, r_n\right] + E^{RL}\left[X_n \middle| y_1^n, r_n\right] - E\left[X_n \middle| y_n, r_n\right] \tag{4.2.6}$$

4.3 Dependence graphs

Let us illustrate different models and smoothing methods discussed above by some dependence graphs. Graphs of classic SLDSs, CGPMSMs, and CGOMSMs are presented in Figure 1. The aim is to show that once CGPMSMs are admitted as being reference models, approximating a given CGPMSM with a CGOMSM is more precise that approximating it with a SLDS.

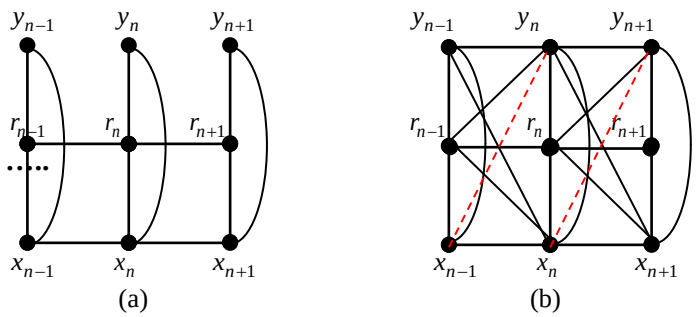


Figure 1**.** Dependence graph of (a) classic switching linear dynamical system (SLDS), (b) CGPMSM (in which dotted red segments would be black ones) and CGOMSM (in which dotted red segments are removed). CGPMSMs and CGOMSMs only differ by existence, or not, of dotted red segment.

Figure 2 represents distributions LR-SL-CGOMSM, RL-SL-CGOMSM, and n-SL-CGOMSM conditional on R_1^N. It illustrates how n-SL-CGOMSM varies with n.

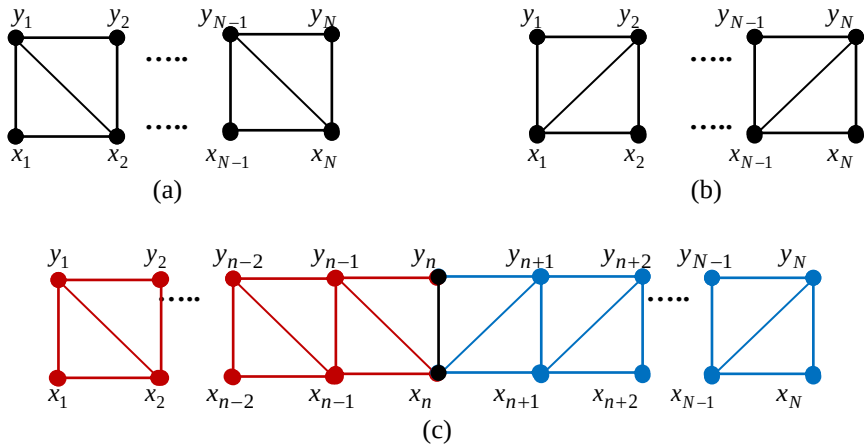
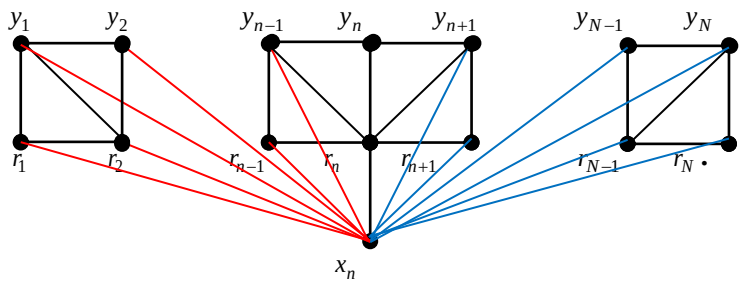
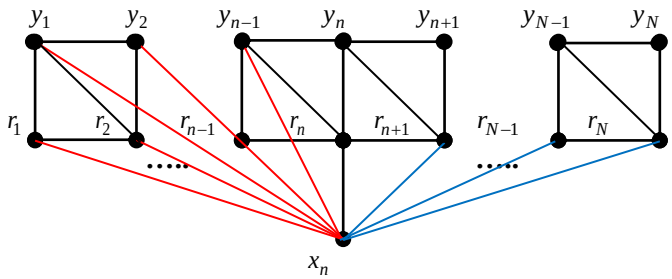


Figure 2. Dependence graphs of distributions $p(x_1^N, y_1^N | r_1^N)$: LR-SL-CGOMSM (a), RL-SL-CGOMSM (b), and n-SL-CGOMSM (c).

Figure 3 illustrates arguments developed in Introduction: the new smoothing uses the information in a more complete manner than the previous one based on CGOMSMs.

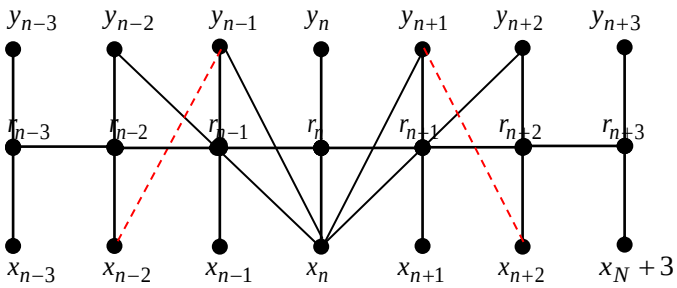


(a)

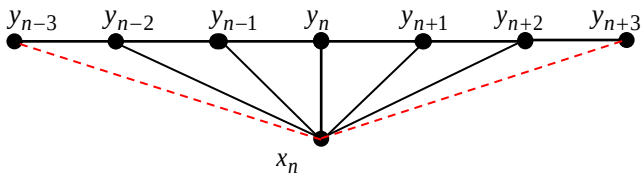


(b)

Figure 3. Dependence of X_n on (R_1^N, Y_1^N) in n-SL-CGOMSM (a), and in LR-CGOMSM (b).



(a)



(b)

Figure 4. Distributions of $(X_n, R_{n-2}^{n+2}, Y_{n-2}^{n+2})$ are identical in SL-CGPMSM and n-SL-CGOMSM (a). Then $p(x_n, y_{n-2}^{n+2})$ also are identical, and $p(x_n | y_{n-2}^{n+2})$ also are as well (b).

5. CONCLUSIONS AND PERSPECTIVES

We proposed an original method of fast smoothing in switching Markovian systems, called "double filtering-based smoothing" (DFBS). Its originality is to consider, for each time $n \in \{1,\ldots,N\}$, two approximating models: one on the left side of n, and another on his right side. In each of these models, called "conditionally Gaussian observed Markov switching models" (CGOMSMs), fast exact filtering is feasible in spite of switches [1, 2, 45]. The "left" CGOMSM is obtained directly, while the "right" CGOMSM is obtained by inverting time. The method is valid in general "conditionally Gaussian pairwise Markov switching models" (CGPMSMs), which extend, in particular, "switching linear dynamical systems" (SLDSs) currently used. It has been seen from previous experiments presented in [1, 2], that the efficiency of fast exact filter applied to the CGOMSM approximating a given CGPMSM was comparable to the efficiency of particle filter applied to the true CGPMSM, the latter being optimal (when taking enough particles). Therefore we may expect that the method is close to the optimal smoothing as well.

As perspectives, let us mention extension to continuous time cases [55, 56]. Another wide perspective is to extend the Markov chain (R_1^N, Y_1^N) considered in n-SL-CGOMSMs introduced in the paper to triplet chains, as mentioned in subsection 2.3, case (i) (see also [10, 34]).

6. REFERENCES

[1] N. Abbassi, D. Benboudjema, and W. Pieczynski, Kalman filtering approximations in triplet Markov Gaussian switching models, IEEE Workshop on Statistical Signal Processing (SSP2011), Nice, France, 28-30 June, 2011.

[2] N. Abbassi, D. Benboudjema, S. Derrode, and W. Pieczynski. Optimal filter approximations in conditionally Gaussian pairwise Markov switching models. IEEE Trans. on Automatic Control, 60(4), pp. 1104-1109, 2015.

[3] B. Ait-el-Fquih and F. Desbouvries, Kalman filtering in triplet Markov chains, IEEE Trans. on Signal Processing, 54(8), pp. 2957-63, 2006.

[4] M. S. Ali, C. K. Ahn, H. R. Karimi, P. Shi, Stability of Markovian Jump Generalized Neural Networks With Interval Time-Varying Delays R. Saravanakumar, IEEE Trans. on Neural Networks and Learning Systems, 28(8), pp. 1840 - 1850, 2017.

[5] C. Andrieu, M. Davy, A. Doucet, Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions, IEEE Trans. on Signal Processing, 51(7), pp. 1762-1770, 2003.

[6] M. Azzouzi and I. T. Nabney. Modelling financial time series with switching state space models. In Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr), pp. 240-249, April 1999. doi: 10.1109/CIFER.1999.771123.

[7] Y. Bar-Shalom, K. C. Chang, H. A. P. Blom, Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm, IEEE Trans. on Aerospace and Electronic Systems, 25(2), pp. 296-300, 1989.

[8] D. Barber. Expectation correction for smoothed inference in switching linear dynamical systems. The Journal of Machine Learning Research, 7, pp. 2515-2540, 2006.

[9] V. S. Barbu and N. Limnios, Semi-Markov chains and hidden semi-Markov models toward applications, Lecture Notes in Statistics, Springer, 2008.

[10] R. Bhar and S. Hamori, Hidden Markov Models: Applications to Financial Economics, Springer Science & Business Media, 2006.

[11] M. Y. Boudaren and W. Pieczynski, Dempster-Shafer fusion of evidential pairwise Markov chains, IEEE Trans. on Fuzzy Systems, 24(6), pp. 1598-1610, 2016.

[12] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models, Annals of the Institute of Statistical Mathematics, 62(1), pp. 61-89, 2010.

[13] O. Cappé, E. Moulines, T. Rydén, Inference in Hidden Markov Models, Springer Verlag, 2005.

[14] O. Costa, M. Fragoso, and R. Marques, Discrete-time Markov jump linear systems, Springer Science & Business Media, 2006.

[15] F. Desbouvries, W. Pieczynski, Modèles de Markov Triplet et filtrage de Kalman, Comptes Rendus de l'Académie des Sciences – Mathématique, Série I, 336(8), pp. 667-670, 2003.

[16] R. Douc, J. Olsson, F. Roueff, Posterior consistency for partially observed Markov models, https://arxiv.org/abs/1608.06851, submitted.

[17] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. IEEE Trans. on Signal Processing, 49(3), pp. 613-624, 2001.

[18] Y. Ephraim and N. Merhav, Hidden Markov processes, IEEE Trans. on Information Theory, 48(6), pp. 1518-1569, 2002.

[19] Y. Ephraim, B. L. Mark, Bivariate Markov processes and their estimation, Foundations and Trends® in Signal Processing 6(1), pp. 1-95, 2013.

[20] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model : analysis and applications, Machine Learning, 32(1) pp. 41-62, 1998.

[21] W. Fong, S. Godsill, A. Doucet, M. West. Monte carlo smoothing with application to audio signal enhancement. IEEE Trans. on Signal Processing, 50(2), pp. 438-449, 2002.

[22] Z. Ghaghramani, M. Jordan, Factorial hidden Markov models, Machine Learning, 29(2), pp. 245-273, 1997.

[23] N. Gordon, B. Ristic, S. Arulampalam. Beyond the Kalman filter: particle filters for tracking applications. Artech House, London, 830, 2004.

[24] I. Gorynin, H. Gangloff, E. Monfrini, W. Pieczynski, Assessing the segmentation performance of pairwise and triplet Markov models, Signal Processing, 145, pp. 183-192, 2018.

[25] I. Gorynin, S. Derrode, E. Monfrini, W. Pieczynski, Fast smoothing in switching approximations of non-linear and non-Gaussian models, Computational Statistics & Data Analysis, 114, 38-46, 2017.

[26] I. Gorynin, S. Derrode, E. Monfrini, and W. Pieczynski. Fast filtering in switching approximations of nonlinear Markov systems with applications to stochastic volatility. IEEE Trans. on Automatic Control, 62(2), pp. 853-862, 2017.

[27] I. Gorynin, E. Monfrini, W. Pieczynski. Bayesian-Assimilation-Based Smoothing in Switching Systems with Application to Financial Trend Analysis, IEEE Trans. on Signal Processing, submitted, 2018.

[28] S. He, Z. Wu, H. Shen, Y. Yin, and Q. Zhu, (Guest Editors) Robust Control, Optimization, and Applications to Markovian Jumping Systems, Abstract and Applied Analysis, Hindawi Publishing Corporation, 2014.

[29] C. J. Kim. Dynamic linear models with Markov-switching, Journal of Econometrics, 60(1, pp. 1-22, 1994.

[30] C. J. Kim and C. Nelson. State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications, vol. 1. The MIT Press, 1 edition, 1999.

[31] T. Koski, Hidden Markov Models for Bioinformatics, Computational Biology, Springer Science & Business Media, 2001.

[32] M. V. Kulikova, Gradient-Based Parameter Estimation in Pairwise Linear Gaussian System, IEEE Trans. on Automatic Control, 62 (3), pp. 1511-1517, 2017.

[33] M. V. Kulikova, J. V. Tsyganova, The UD-based Approach for Designing Pairwise Kalman Filtering Algorithms, IFAC-PapersOnLine, 50(1), pp. 1619-1624, 2017.

[34] P. Lanchantin, J. Lapuyade-Lahorgue and W. Pieczynski, Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise, Signal Processing, 91(2) pp. 163-175, 2011.

[35] S. Le Cam, F. Salzenstein, and C. Collet, Fuzzy pairwise Markov chain to segment correlated noisy data, Signal Processing 88(10), pp. 2526-2541, 2008.

[36] J. Lember, H. Matzinger, J. Sova, F. Zucca, Lower bounds for moments of global scores of pairwise Markov chains, Stochastic Processes and their Applications, 128(5), pp. 1678-1710, 2018.

[37] U. Lerner. Hybrid Bayesian networks for reasoning about complex systems. PhD thesis, Stanford university, 2002.

[38] H. Lin, P. J. Antsaklis, Stability and stabilizability of switched linear systems: a survey of recent results, IEEE Trans. on Automatic Control, 54(2), pp. 308-322, 2009.

[39] R. Mamon and R. Elliott, Hidden Markov Models in Finance, Springer, 2007.

[40] B. Mesot and D. Barber. Switching linear dynamical systems for noise robust speech recognition. IEEE Trans. on Audio, Speech, and Language Processing, 15(6), pp. 1850-1858, 2007.

[41] K. Murphy and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In Sequential Monte Carlo methods in practice, pp. 499-515. Springer, 2001.

[42] V. Nemesin, S. Derrode, Robust blind pairwise Kalman algorithms using QR decompositions. IEEE Trans. on Signal Processing, 61(1), pp. 5-9, 2013.

[43] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, P. Shi, M. T. Lim, Switching Extensible FIR Filter Bank for Adaptive Horizon State Estimation With Application, IEEE Trans. on Control Systems Technology, 24(3), pp. 1052-1058, 2016.

[44] V. Pavlovic, J. Rehg, and J. Maccormick. Learning switching linear models of human motion. In Advances in Neural Information Processing systems (NIPS 13), pp. 981-987, 2000.

[45] Y. Petetin and F. Desbouvries, A class of fast exact Bayesian filters in dynamical models with jumps, IEEE Trans. on Signal Processing, 62(14), pp. 3643-53, 2014.

[46] W. Pieczynski, C. Hulard, and T. Veit, Triplet Markov Chains in hidden signal restoration, SPIE's International Symposium on Remote Sensing, September 22-27, Crete, Greece, 2002.

[47] W. Pieczynski, Pairwise Markov chains, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(5), pp. 634-639, 2003.

[48] W. Pieczynski. Exact filtering in conditionally Markov switching hidden linear models, Comptes Rendus Mathematique, 349(9), pp. 587-590, 2011.

[49] W. Pieczynski and F. Desbouvries, Kalman Filtering using Pairwise Gaussian Models, International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hong-Kong, April 2003.

[50] R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE, 77(2), pp. 257-286.

[51] A. Rosti and M. Gales. Switching linear dynamical systems for speech recognition. University of Cambridge, Department of Engineering, 2003.

[52] F. Salzenstein, C. Collet, S. Le Cam, and M. Hatt, Non stationary fuzzy Markov chains, Pattern Recognition Letters, 28(6), pp. 2201-2208, 2007.

[53] S. Sarkka, P. Bunch, and S. Godsill. A backward-simulation based Rao-Blackwellized particle smoother for conditionally linear Gaussian models, IFAC Proceedings Volumes, 45(16), pp. 506-511, 2012.

[54] B. Shen, J. K. Tugnait, Interacting Multiple Model Fixed-Lag Smoothing Algorithm For Markovian Switching Systems, IEEE Trans. on Aerospace and Electronic Systems, 36(1), pp. 243-250, 2000.

[55] P. Shi, EK Boukas, RK Agarwal Kalman filtering for continuous-time uncertain systems with Markovian jumping parameters, IEEE Trans. on Automatic Control 44 (8), 1592-1597, 1999.

[56] P. Shi, Y. Xia, G. P. Liu, D. Rees, On designing of slinding-mode control for stochastic jump systems, IEEE Trans. on Automatic Control, 51(1), pp. 97-103, 2006.

[57] M. Vidyasagar, Hidden Markov Processes: Theory and Applications to Biology, Princeton Series in Applied Mathematics, Princeton University Press, 2014.

[58] K. Weiss, N. Kaempchen, and A. Kirchner. Multiple-model tracking for the detection of lane change maneuvers. In IEEE Intelligent Vehicles Symposium, 2004, pp. 937-942, 2004.

[59] S.-Z. Yu, Hidden semi-Markov models, Artificial Intelligence, 174(2), pp. 215-243, 2010.

[60] S.-Z. Yu, S.-Z. Yu, Hidden semi-Markov models. Theory, algorithms and applications, Elsevier, 2016.

[61] L. Zhang, J. Lam, Necessary and sufficient conditions for analysis and synthesis of Markov jump linear systems with incomplete transition descriptions, IEEE Trans. on Automatic Control, 55(7), 1695-1701, 2010.

[62] L. Zhang, Y. Zhu, P. Shi, Q. Lu, Time-dependent switched discrete-time linear systems: control and filtering. Studies in Systems, Decision and Control, vol. 53, Springer, 2016.

[63] F. Zheng, S. Derrode, and W. Pieczynski, Parameter estimation in switching Markov systems and unsupervised smoothing, IEEE Trans. on Automatic Control, accepted, July 2018.

[64] O. Zoeter and T. Heskes. Deterministic approximate inference techniques for conditionally Gaussian state space models, Statistics and Computing, 16(3), pp. 279-292, 2006.