KALMAN FILTERING APPROXIMATIONS IN TRIPLET MARKOV GAUSSIAN SWITCHING MODELS

Noufel Abbassi[1], *Dalila Benboudjema*[2], *and Wojciech Pieczynski*[1]

[1] Institut Telecom, Telecom Sudparis, Département CITI, CNRS UMR 5157, 9, rue Charles Fourier, 91000 Evry, France (E-mail : Noufel.Abbassi@it-sudparis.eu, Wojciech.Pieczynski@it-sudparis.eu)
[2] ENSEA, Université Cergy-Pontoise, Equipe ETIS, CNRS UMR 8051, F- 95000 Cergy-Pontoise, France (E-mail : Dalila.Benboudjema@ensea.fr)

ABSTRACT

We consider a general triplet Markov Gaussian linear system (X,R,Y), where X is hidden continuous, R is hidden discrete, and Y is observed continuous. Exact Kalman filter (KF) is not workable and two approximations are considered in the paper. The classical one consists of particle filtering, which is a new extension of the classical method we propose. Another new method we propose consists of replacing the model by a simpler one, in which (R,Y) is Markovian and in which exact KF can be performed. We show the interest of our method via experiments.

Index Terms— Gaussian switching system, exact Kalman filtering, particle filter

1. INTRODUCTION

Let us consider three random sequences $X_1^N = (X_1, ..., X_N)$, $R_1^N = (R_1, ..., R_N)$, and $Y_1^N = (Y_1, ..., Y_N)$, where the sequences X_1^N and Y_1^N are real valued, while R_1^N is discrete finite, each R_n taking its values in $S = \{1, ..., K\}$. X_1^N and R_1^N are hidden, while Y_1^N is observed.

The problem we deal with is the sequential search of (R_1^N, X_1^N) from Y_1^N. In classical linear Gaussian models, in which R_1^N is Markovian and X_1^N is a linear Gaussian system conditionally on R_1^N, exact computation of optimal filter is not workable and different approximations must be used [1, 3, 4, 5, 6, 10]. Here we consider more general model and we compare two filtering methods: the particle filter based one, which is a simple extension of the classical method [1, 5] to the general model considered, and a new method, which is an exact one in an approximated model [2], the latter being a particular case of the model proposed in [8].

Let us set $T_n = (X_n, R_n, Y_n)$, and let us assume the triplet $T_1^N = (T_1, ..., T_N)$ Markovian and stationary. In addition, (X_1^N, Y_1^N) will be assumed Gaussian conditionally on R_1^N, and the transition $p(t_{n+1}|t_n)$ will be assumed of the form:

$$p(t_{n+1}|t_n) = p(r_{n+1}|r_n) p(x_{n+1}, y_{n+1}|r_n, r_{n+1}, x_n, y_n), \qquad (1.1)$$

which implies that R_1^N a Markov chain. Such a model will be called "Triplet Markov Gaussian Switching Model" (TMGSM).

The aim of our paper is double:

(1) Extend the particle filter based method of searching X_1^N and R_1^N from Y_1^N, valid in the case of classical systems [1, 5], to TMGSM ;

(2) Compare its efficiency with a new method based on exact computation in an approximate model suggested in [2].

Thus we compare an approximate filter (particle filter) based on the true model with an optimal exact filter, proposed in [2], based on an approximate model. We show that the latter method can be slightly better, and, above all, is much faster than the former.

2. TRIPLET MARKOV GAUSSIAN SWITCHING MODEL

There are two equivalent representations of TMGSM :

(i) the distribution of T_1^N is defined by the distribution $p(t_1, t_2)$, which is given by $p(r_1, r_2)$ and the Gaussian distributions $p(x_1, y_1, x_2, y_2|r_1, r_2)$. Setting $r_1^2 = (r_1, r_2)$, we will assume that all the means of these Gaussian distributions are null, and that the variances-covariances matrix is given by [2]:

$$\Delta(r_1^2) = \begin{array}{c} \\ X_1 \\ Y_1 \\ X_2 \\ Y_2 \end{array}\begin{array}{c} \begin{array}{cccc} X_1 & Y_1 & X_2 & Y_2 \end{array} \\ \begin{bmatrix} 1 & b(r_1) & a(r_1^2) & d(r_1^2) \\ b(r_1) & 1 & e(r_1^2) & c(r_1^2) \\ a(r_1^2) & e(r_1^2) & 1 & b(r_2) \\ d(r_1^2) & c(r_1^2) & b(r_2) & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \Gamma(r_1) & A(r_1^2)^T \\ A(r_1^2) & \Gamma(r_2) \end{bmatrix},$$

with $\Gamma(r_1) = \begin{bmatrix} 1 & b(r_1) \\ b(r_1) & 1 \end{bmatrix}$, $A(r_1^2) = \begin{bmatrix} a(r_1^2) & e(r_1^2) \\ d(r_1^2) & c(r_1^2) \end{bmatrix}$; (1.2)

(ii) the distribution of T_1^N is defined by the distribution $p(x_1, r_1, y_1) = p(r_1) p(x_1, y_1 | r_1)$, with $p(r_1)$ the marginal distribution of $p(r_1, r_2)$ and $p(x_1, y_1 | r_1)$ zero means Gaussian distribution with covariance matrix $\Gamma(r_1)$, and the linear system

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \alpha(r_n^{n+1}) & \beta(r_n^{n+1}) \\ \gamma(r_n^{n+1}) & \delta(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \Sigma^*(r_n^{n+1}) \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}, \quad (1.3)$$

with $\begin{bmatrix} \alpha(r_n^{n+1}) & \beta(r_n^{n+1}) \\ \gamma(r_n^{n+1}) & \delta(r_n^{n+1}) \end{bmatrix} = A(r_n^{n+1}) \Gamma(r_n)^{-1} =$

$$\begin{bmatrix} a(r_n^{n+1}) & e(r_n^{n+1}) \\ d(r_n^{n+1}) & c(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} 1 & b(r_n) \\ b(r_n) & 1 \end{bmatrix}^{-1} =$$

$$\frac{1}{1 - b(r_n)^2} \begin{bmatrix} a(r_n^{n+1}) - e(r_n^{n+1}) b(r_n) & -a(r_n^{n+1}) b(r_n) + e(r_n^{n+1}) \\ d(r_n^{n+1}) - c(r_n^{n+1}) b(r_n) & -d(r_n^{n+1}) b(r_n) + c(r_n^{n+1}) \end{bmatrix}, \text{ and}$$

$\Sigma^*(r_n^{n+1})$ such that $\Sigma^*(r_n^{n+1})(\Sigma^*(r_n^{n+1}))^T = \Sigma(r_n^{n+1})$, with

$$\Sigma(r_n^{n+1}) = \Gamma(r_{n+1}) - A(r_n^{n+1}) \Gamma(r_n)^{-1} A(r_n^{n+1})^T =$$

$$= \begin{bmatrix} \alpha^1(r_n^{n+1}) & \beta^1(r_n^{n+1}) \\ \beta^1(r_n^{n+1}) & \delta^1(r_n^{n+1}) \end{bmatrix}, \text{ and } \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right),$$

independent from (X_1^n, Y_1^n) for each $n = 1, \ldots, N-1$.

Thus for K possible switches the general model is defined by the probability distribution $p(r_1, r_2)$ on $S^2 = \{1, \ldots, K\}^2$ and by the parameters $a(r_1^2)$, $b(r_1)$, $c(r_1^2)$, $d(r_1^2)$, $e(r_1^2)$, with the constraint that the covariance matrix $\Delta(r_1^2)$ must be positive definite.

Remark 2.1
The classical Conditionally Gaussian Linear State-Space Model (CGLSSM) can be defined as a system verifying the following:
R_1^N is a Markov chain ;
$X_{n+1} = \alpha(r_{n+1}) X_n + \sigma_1^*(r_{n+1}) U_{n+1}$;
$Y_{n+1} = \gamma(r_{n+1}) X_{n+1} + \sigma_2^*(r_{n+1}) V_{n+1}$.
Thus CGLSSM is a particular TMGSM in which $\beta(r_n^{n+1}) = 0$, $\delta(r_n^{n+1}) = 0$ and the covariances in $\Sigma^*(r_n^{n+1})$ are zero.

Let us consider two particular cases (PC) of the TMGSM. In the first one, called TMGSM-PC1, we take $\beta(r_n^{n+1}) = 0$, which means that $e(r_n^{n+1}) = a(r_n^{n+1}) b(r_n^{n+1})$ for each r_n^{n+1}. In TMGSM-PC1 both R_1^N, (X_1^N, R_1^N) are Markovian, and it is an extension of the classical CGLSSM mentioned in Remark 2.1. In the second case, called TMGSM-PC2, we take $\gamma(r_n^{n+1}) = 0$, which means that $d(r_n^{n+1}) = c(r_n^{n+1}) b(r_n^{n+1})$ for each r_n^{n+1}. In TMGSM-PC2 both R_1^N and (R_1^N, Y_1^N) are Markovian.

3. EXACT FILTERING IN TMGSM-PC2

Let us specify how the exact filtering runs in TMGSM-PC2. In TMGSM-PC2 we have:
$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \alpha(r_n^{n+1}) & \beta(r_n^{n+1}) \\ 0 & \delta(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \Sigma^*(r_n^{n+1}) \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}$, which means that $p(x_{n+1}, y_{n+1} | x_n, r_n^{n+1}, y_n)$ is Gaussian $N\left(\begin{pmatrix} \alpha(r_n^{n+1}) x_n + \beta(r_n^{n+1}) y_n \\ \delta(r_n^{n+1}) y_n \end{pmatrix}, \begin{bmatrix} \alpha^1(r_n^{n+1}) & \beta^1(r_n^{n+1}) \\ \beta^1(r_n^{n+1}) & \delta^1(r_n^{n+1}) \end{bmatrix}\right)$. Applying the classical rules of Gaussian conditioning, we can say that the mean of the Gaussian distribution $p(x_{n+1} | x_n, r_n, y_n, r_{n+1}, y_{n+1})$ is

$$\alpha(r_n^{n+1}) x_n + \beta(r_n^{n+1}) y_n + \frac{\beta^1(r_n^{n+1})}{\delta^1(r_n^{n+1})}(y_{n+1} - \delta(r_n^{n+1}) y_n) =$$

$$\alpha(r_n^{n+1}) x_n + [\beta(r_n^{n+1}) - \frac{\beta^1(r_n^{n+1}) \delta(r_n^{n+1})}{\delta^1(r_n^{n+1})}] y_n + \frac{\beta^1(r_n^{n+1})}{\delta^1(r_n^{n+1})} y_{n+1} =$$

$A(r_n^{n+1}) x_n + B(r_n^{n+1}) y_n + C(r_n^{n+1}) y_{n+1}$. This means that we can write

$$X_{n+1} = A(R_n^{n+1}) X_n + B(R_n^{n+1}) Y_n + C(R_n^{n+1}) Y_{n+1} + D(R_n^{n+1}) W_{n+1}, \quad (2.1)$$

with $E[W_{n+1} | x_n, r_1^{n+1}, y_1^{n+1}] = 0$. Then $p(r_{n+1} | y_1^{n+1})$ and $E[X_{n+1} | r_{n+1}, y_1^{n+1}]$ can be computed from $p(r_n | y_1^n)$, $E[X_n | r_n, y_1^n]$, and y_{n+1} as follows.

First, (R_1^N, Y_1^N) being a hidden Markov chain we have $p(r_{n+1}|y_1^{n+1}) = \frac{\sum_{r_n} p(r_n|y_1^n)p(r_{n+1}, y_{n+1}|r_n, y_n)}{\sum_{r_n, r_2} p(r_n|y_1^n)p(r_{n+1}, y_{n+1}|r_n, y_n)]}$. Second, in TMGSM-PC2 we have $p(x_n|r_n, r_{n+1}, y_1^{n+1}) = p(x_n|r_n, y_1^n)$, which implies that $E[X_n|r_n, r_{n+1}, y_1^{n+1}] = E[X_n|r_n, y_1^n]$. Thus taking the conditional expectation of (2.1) we obtain

$E[X_{n+1}|r_n, r_{n+1}, y_1^{n+1}] =$

$A(r_n^{n+1})E[X_n|r_n, r_{n+1}, y_1^{n+1}] + B(r_n^{n+1})y_n + C(r_n^{n+1})y_{n+1} =$

$A(r_n^{n+1})E[X_n|r_n, y_1^n] + B(r_n^{n+1})y_n + C(r_n^{n+1})y_{n+1}$, and thus

$E[X_{n+1}|r_{n+1}, y_1^{n+1}] = \sum_{r_n} E[X_{n+1}|r_n, r_{n+1}, y_1^{n+1}]p(r_n|y_1^{n+1}) =$

$A(r_n^{n+1})[\sum_{r_n} E[X_n|r_n, y_1^n]p(r_n|y_1^{n+1}) + B(r_n^{n+1})y_n + C(r_n^{n+1})y_{n+1} =$

$A(r_n^{n+1})\frac{\sum_{r_n} E[X_n|r_n, y_1^n]p(r_n|y_1^n)p(y_{n+1}|r_n, y_n)}{\sum_{r_n} p(r_n|y_1^n)p(y_{n+1}|r_n, y_n)} +$

$B(r_n^{n+1})y_n + C(r_n^{n+1})y_{n+1}$.

4. PARTICLE FILTERING IN TMGSM

Let us consider a general TMGSM. The problem is that $p(r_{n+1}|y_1^{n+1})$ cannot be computed sequentially and is has to be approximated. We briefly specified how the particle filter can be used, which is not very different from its use in the classical switching systems developed in [1, 5].

Let us imagine that $R_1^N = r_1^N$ is given. Then the distribution of (X_1^N, Y_1^N) conditional on $R_1^N = r_1^N$ is the distribution of a pairwise Gaussian Markov chain. Thus $p(x_{n+1}|r_1^{n+1}, y_1^{n+1})$ is computable from $p(x_n|r_1^{n+1}, y_1^n)$ and y_{n+1} using Kalman filter technique extended to pairwise Gaussian Markov chains, as proposed in [7]. Besides, as in the classical models, we have the following formula, which gives $p(r_1^{n+1}|y_1^{n+1})$ from $p(r_1^n|y_1^n)$ and y_{n+1} :

$$p(r_1^{n+1}|y_1^{n+1}) = \frac{p(y_{n+1}|r_1^{n+1}, y_1^n)p(r_{n+1}|r_n)}{p(y_{n+1}|y_1^n)} p(r_1^n|y_1^n)$$
$$= \frac{p(y_{n+1}|r_1^{n+1}, y_1^n)p(r_{n+1}|r_n)}{\sum_{r_1^{n+1}} p(y_{n+1}|r_1^{n+1}, y_1^n)p(r_{n+1}|r_n)p(r_1^n|y_1^n)} p(r_1^n|y_1^n) \quad (4.1)$$

This formula is then used to sequentially approximate $p(r_{n+1}|y_1^{n+1})$ with the particle filter technique. First, $p(r_1^n|y_1^n)$ is approximated with

$$p(r_1^n|y_1^n) \approx \hat{p}_{N_p}(r_1^n|y_1^n) = \frac{1}{N_p}\sum_{i=1}^{N_p} \delta_{\{r_1^{n,i}\}}, \quad (4.2)$$

where N_p is the number of sampled trajectories. Having $p(r_1^n|y_1^n)$ gives the searched $p(x_n|y_1^n)$ with the formula

$$p(x_n|y_1^n) = \sum_{r_1^n} p(r_1^n|y_1^n)p(x_n|r_1^n, y_1^n) \quad (4.3)$$

Injecting (4.2) into (4.3) we obtain

$$p(x_n|y_1^n) \approx \hat{p}_{N_p}(x_n|y_1^n) = \frac{1}{N_p}\sum_{i=1}^{N_p} p(x_n|r_1^{n,i}, y_1^n), \quad (4.4)$$

knowing that the parameters of the densities $p(y_{n+1}|r_1^{n+1}, y_1^n)$ in (4.1) and $p(x_n|r_1^{n,i}, y_1^n)$ for $i \in \{1, ..., N_p\}$ in (4.4) are computed for by the Kalman algorithm [7].

5. EXPERIMENTS

Let us consider the matrix (1.2) with $b(r_1) = b(r_2) = b$ $a(r_1^2) = a(r_2)$, $c(r_1^2) = c(r_2)$, $d(r_1^2) = d(r_2)$, and $e(r_1^2) = e(r_2)$. Our aim is to simulate realizations of $T_1^N = (T_1, ..., T_N)$, with $T_n = (X_n, R_n, Y_n)$, and estimate (R_1^N, X_1^N) from Y_1^N. We will consider three methods:

(i) the "reference method", denoted by RM, where $R_1^N = r_1^N$ is considered as known and where X_1^N is obtained by the optimal Kalman filter;

(ii) the particle filter based method, denoted by PF; and

(iii) the "alternative" method, denoted by AM, which consists of taking an approximate model by considering that $\gamma(r_n) = 0$ for each r_n, and by applying the exact filtering described in section 3.

We will consider the case of two classes for the switches: each r_n can be 0 or 1. We will consider four experiments, corresponding to the following four models. In all of them, we chose the following parameters $b = 0.3$, $a(0) = 0.1$, $a(1) = 0.5$, $c(0) = 0.4$, $c(1) = 0.9$, $e(0) = 0.75$, $e(1) = 0.33$. Then we chose five different couples $(d^i(0), d^i(1))$, $i = 1, ..., 5$, in such a way that the corresponding $\gamma(r_n) = [d(r_n) - bc(r_n)]/(1 - b^2)$ (see section 1) are: Case 1 : $\gamma(0) = \gamma(1) = 0.0$; Case 2 : $\gamma(0) = \gamma(1) = 0.1$; Case 3 :

$\gamma(0)=\gamma(1)=0.2$; Case 4 : $\gamma(0)=\gamma(1)=0.3$, Case 5 : $\gamma(0)=\gamma(1)=0.4$.

In each case we simulate $T_1^N = t_1^N = (x_1^N, r_1^N, y_1^N)$ according to the real parameters and (x_1^N, r_1^N) are searched with RM, PF, and AM. The difference between x_1^N and its estimate $\hat{x}_1^N$ is measured with the squared error $\frac{1}{N}\sqrt{\sum_{n=1}^{N}(x_n-\hat{x}_n)^2}$, and the difference between r_1^N and its estimate $\hat{r}_1^N$ is measured with the error ratio. We took $N=1000$, and errors are measured over the last 200 data. In particle filter method we used 500 particles.

	Case 1	Case 2	Case 3	Case 4	Case 5
γ	0.0	0.1	0.2	0.3	0.4
	Squared error between x_1^N and $\hat{x}_1^N$				
RM	0.0591	0.0577	0.0562	0.0591	0.0555
PF	0.0616	0.0604	0.0599	0.0617	0.0587
AM	0.0616	0.0598	0.0590	0.0625	0.0595
	Error ratio between r_1^N and $\hat{r}_1^N$				
RM	0 %	0%	0 %	0 %	0 %
PF	25.1%	27.6%	25.9%	23.4%	27.1%
AM	24.2%	25.1%	26.0%	24.4%	29.0%
	Computer time in seconds				
RM	0.15	0.13	0.14	0.14	0.16
PF	133.91	111.87	108.18	109.04	145.93
AM	0.39	0.31	0.29	0.30	0.48

Tab. 1. Errors obtained with the reference (optimal) method (RM), a particle filter based method (PF), and the new alternative method (AM). The results are means of 10 independent experiments.

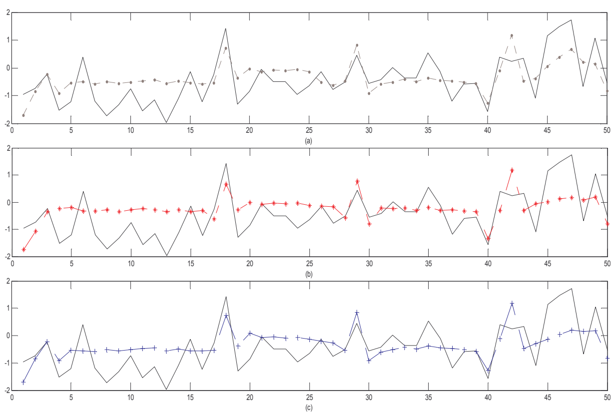

Fig. 1. Un example of trajectories (50 last points) corresponding to the case 4: $\gamma(0)=\gamma(1)=0.3$. True x_1^N (continuous line) and $\hat{x}_1^N$ obtained with RM (high), PF (middle), and AM (bottom).

We can see that even in the case 5, where the true model is "far" from the TMGSM-PC2 used (they are equal in the case 1), AM method is comparable to the PF one, and remains much faster.

6. CONCLUSION

We proposed an original particle filter valid in a general "Triplet Markov Gaussian Switching Model" (TMGSM), which extends the classical Conditionally Gaussian Linear State-Space Model (CGLSSM). We compared its efficiency with an exact filter valid in a particular TMGSM, which has been seen as an approximation of the general TMGSM. We showed that both methods are, at least in the context of our study, of comparable efficiency; however, the second one is much faster.

7. REFERENCES

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

[2] D. Benboudjema, M. Malainin, and W. Pieczynski, Exact Kalman filtering in pairwise Gaussian switching systems, Applied Stochastic Models and Data Analysis (ASMDA 2011), June 7-10, Rome, Italy, 2011.

[3] O. Cappé, E. Moulines, and T. Ryden, *Inference in hidden Markov models*, Springer, 2005.

[4] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete time Markov jump linear systems*, New York, Springer-Verlag, 2005.

[5] A. Doucet, N. J. Gordon, and V. Krishnamurthy, Particle filters for state estimation of Jump Markov Linear Systems, *IEEE Trans. on Signal Processing*, Vol. 49m pp.613-624, 2001.

[6] P. Giordani, R. Kohn, and D. van Dijk, A unified approach to nonlinearity, structural change, and outliers, *Journal of Econometrics*, 137, 112-133, 2007.

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

[8] W. Pieczynski, Exact filtering in conditionally Markov switching hidden linear models, *Comptes Rendus Mathematique*, Vol. 349, No. 9-10, pp. 587-590, May 2011.

[9] W. Pieczynski and N. Abbassi, Exact filtering and smoothing in short or long memory stochastic switching systems, 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2009), September 2-4, Grenoble, France, 2009.

[10] J. K. Tugnait, Adaptive estimation and identification for discrete systems with Markov jump parameters, *IEEE Trans. on Automatic Control*, AC-25, 1054-1065, 1982.