Filtering in Gaussian Linear Systems With Fuzzy Switches

Zied Bouyahia, Stéphane Derrode, and Wojciech Pieczynski

Abstract—**This paper presents recent results on conditionally Gaussian observed Markov switching models by incorporating fuzzy switches in the model, instead of hard ones. This kind of generalization is of interest for applications involving continuous switching regimes, such as tracking an object using cameras in intermittent sunlight and shadow conditions. The filter developed hereby is recursive, optimal, and exact, up to an approximation of integrals according to some fuzzy measures. Experiences on simulated and on real data—dealing with outdoor air temperature and power consumption of a building—confirm the accuracy and effectiveness of the proposed filter compared with the hard filter with "crisp" switches.**

Index Terms—**Fast filtering, fuzzy switching linear model, triplet Markov models.**

Nomenclature

CMSHLM	Conditionally Markov switching hidden linear model.
CGMSM	Conditionally Gaussian Markov switching model.
CGOMSM	Conditionally Gaussian observed Markov switching model, a CGMSM where (5) holds.
CGOFMSM	Conditionally Gaussian observed fuzzy Markov switching model, a CGOMSM with fuzzy switches.
$\boldsymbol{X}_1^N$	Stochastic process of size N.
X_n, x_n	Random variable at time index n, and a realization.
$\boldsymbol{X}_1^N, \boldsymbol{Y}_1^N, \boldsymbol{R}_1^N$	State, observation, and switch (also called jump) processes, respectively.
$\boldsymbol{Z}_n, \boldsymbol{T}_n$	Denotes $(X_n, Y_n)^\intercal$ and $(X_n, R_n, Y_n)^\intercal$, respectively.
K	Number of switches.
m, q	Dimension of the states and observations.
$\mathbb{E}\left[X_n \middle\vert \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n\right]$	Filter at time index n [also denoted $x_n(\boldsymbol{y}_1^n)$].
$\boldsymbol{M}_n(r_n)$	Denotes $\mathbb{E}\left[\left(X_n, Y_n\right)^\intercal \middle\vert r_n\right]$.
$\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$	Fuzzy measure on $[0,1]$ used, where δ is the Dirac mass, and μ the Lebesgue measure.
$\nu \otimes \nu$	Denotes the product of measures.

Manuscript received February 1, 2019; revised April 28, 2019; accepted June 4, 2019. Date of publication June 10, 2019; date of current version August 4, 2020. *(Corresponding author: Stéphane Derrode.)*

Z. Bouyahia is with the College of Arts and Applied Sciences, Dhofar University, Salalah 211, Oman (e-mail: zbouyahia@du.edu.om).

S. Derrode is with the University of Lyon, CNRS, Central School of Lyon, LIRIS, CNRS UMR 5205, 69134 Ecully, France (e-mail: stephane.derrode@ec-lyon.fr).

W. Pieczynski is with Telecom SudParis, SAMOVAR, CNRS UMR 5157, Polytechnic Institute of Paris, 91000 Évry, France (e-mail: wojciech.pieczynski@telecom-sudparis.eu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2019.2921944

I. Introduction

Let us consider the problem of statistical optimal filtering in the presence of switches. Three stochastic sequences are involved: states $\boldsymbol{X}_1^N = (X_1, \ldots, X_N)$, switches $\boldsymbol{R}_1^N = (R_1, \ldots, R_N)$, and observations $\boldsymbol{Y}_1^N = (Y_1, \ldots, Y_N)$. For each $n = 1, \ldots, N$, the random variables X_n and Y_n take their values in $\mathbb{R}^m$ and $\mathbb{R}^q$, respectively, whereas R_n takes its values in the finite discrete set $\Omega = \{0, 1, \ldots, K-1\}$. For the sake of simplification, we will assume in the remainder of this paper that 1) $m = q = 1$, i.e., $\boldsymbol{X}_1^N$ and $\boldsymbol{Y}_1^N$ are scalar-valued processes, and 2) that $K = 2$. We consider these hypotheses only to simplify the presentation of the filtering method; the algorithms proposed in the following can be extended to the cases of vectorial processes and for a number of switches greater than 2.

The problem is to sequentially estimate each X_{n+1} from $\boldsymbol{Y}_1^{n+1}$. Fast recursive optimal filters compute the estimated $\hat{x}_{n+1}(\boldsymbol{y}_1^{n+1}) = \mathbb{E}[X_{n+1} \vert \boldsymbol{Y}_1^{n+1} = \boldsymbol{y}_1^{n+1}]$ from $\hat{x}_n(\boldsymbol{y}_1^n)$ and y_{n+1}. The "conditionally Gaussian linear state-space models" (CGLSSM), considered as a natural way to extend Gaussian systems to Gaussian switching ones, do not allow for a filtering scheme that can be performed in a reasonable running time [1]–[5]. Classically, CGLSSMs rely on the following assumptions.

1) $\boldsymbol{R}_1^N$ is Markov.
2) $\boldsymbol{X}_1^N$ is Markov conditionally on $\boldsymbol{R}_1^N$.
3) (Y_n), $1 \leq n \leq N$, are Gaussian, independent conditionally on $(\boldsymbol{R}_1^N, \boldsymbol{X}_1^N)$ and verify

$$p\left(y_n \left\vert \boldsymbol{r}_1^N, \boldsymbol{x}_1^N\right.\right) = p\left(y_n \left\vert r_n, x_n\right.\right). \tag{1}$$

In CGLSSMs, $\boldsymbol{R}_1^N$ and $(\boldsymbol{R}_1^N, \boldsymbol{X}_1^N)$ are both Markov, and $p(y_n \vert \boldsymbol{r}_1^N, \boldsymbol{x}_1^N)$ is very simple. These assumptions do not allow for exact computation of recursive filters, as $(\boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$ is not Markov and $p\left(r_n \left\vert \boldsymbol{y}_1^n\right.\right)$ cannot be computed sequentially and exactly. This problem has been addressed in recent "conditionally Markov switching hidden linear models" (CMSHLMs [6]), in which both $\boldsymbol{R}_1^N$ and $(\boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$ are Markov, and $p\left(y_n \left\vert \boldsymbol{r}_1^N, \boldsymbol{x}_1^N\right.\right)$ is pretty general. Here, we consider particular

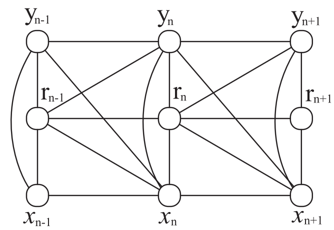

Fig. 1. Dependence graph of CGOMSM.

Gaussian CMSHLMs called "conditionally Gaussian observed Markov switching models" (CGOMSMs [7]–[9]), which verify

1) $\boldsymbol{R}_1^N, (\boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$ and $(\boldsymbol{X}_1^N, \boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$ are Markov;
2) $(\boldsymbol{X}_1^N, \boldsymbol{Y}_1^N)$ is Gaussian conditionally on $\boldsymbol{R}_1^N$.

Fig. 1 illustrates the dependencies between the stochastic processes defining the studied CGOMSM. Such an approach is different from classic ones since the hidden chain $(\boldsymbol{R}_1^N, \boldsymbol{X}_1^N)$ is no longer assumed Markov, as it has been usually done. Thus, recursive exact filtering is feasible in CGOMSM s and the interesting point is that these models can be quite close to the classic CGLSSMs [7], [10].

The values $\boldsymbol{R}_n^{n+1} = \boldsymbol{r}_n^{n+1}$ govern the parameters of the distribution

$$p\left(x_{n+1}, y_{n+1} \left| x_n, \boldsymbol{r}_n^{n+1}, y_n \right.\right)$$

and thus there exist four possible transitions according to $\boldsymbol{r}_n^{n+1} \in \Omega^2 = \{0,1\}^2$. The main limitation of the classical switching model, which relies on hard switches is that it does not take into account the transient transition between switches. In real-world applications (c.f., Section III), the crisp transition can cause the model to discard significant information that corresponds to the time period during which the system switches from one regime to another. Therefore, the hard switches modeling impacts the accuracy of the filtering scheme. If we consider the example of tracking a moving object using sensors in intermittent sunlight and shadow conditions (both corresponding to hard switches 0 and 1), there exist intermediary situations as sunlight condition may fade into shadow in a continuous manner. Considering only the hard switches imposes the filtering method to consider only one of the states 0 and 1 during the transition and consequently this shall compromise the accuracy of the filtering as the system state during the transitory phase is a mixture of the two hard switches. Thus, a more complete model would be to associate a set of parameters to each $\boldsymbol{r}_n^{n+1} \in [0,1]^2$. This is the very aim of the paper: we extend the Markov chain $\boldsymbol{R}_1^N$ used in CGOMSM s to a "fuzzy" one. Such models called "fuzzy models" have been proposed in [11] in a simple context, without Markovianity, to deal with fuzzy image segmentation. Then, they have been extensively used in hidden Markov chains [12]–[14] and hidden Markov fields [14]–[16]. Here, the hidden fuzzy Markov chain (FMC) will be considered to model the pair (switches and observations) in the context of filtering in presence of jumps, which is possible as the pair $(\boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$ is Markov in CGOMSM s.

To the best of our knowledge, although there exist several research works dealing with fuzzy Markov models, the literature for fuzzy Markov jump model filtering is surprisingly scant despite its practical potential. Recently, the discrete-time Takagi–Sugano (T–S) approach to fuzzy filter design for Markovian jump model has been gaining a significant interest over the last few years—see for example [17] and[18] and the references therein—especially in the fuzzy control and fault detection research community. Unlike the T-S model, our approach assumes that, conditionally to jumps, the model is pairwise linear, which enriches the classical linear models. The proposed fuzzy filter allows for exact calculations for the filter, up to numerical approximation of some integrals, as precised in the text. In the remaining, we start with a brief description of the CGOMSM in Section II, and pursue with the description of the original fuzzy jump model in Section III. Sections IV and V depict how the corresponding "fuzzy" filter runs. Section VI reports experimental results that show how the fuzzy filter can improve filtering from the CGOMSM, whereas Section VII reports comparative filtering result on real data.

II. Conditionally Gaussian Observed Markov Hard Switching Model

Let us set $\boldsymbol{Z}_n = (X_n, Y_n)^\intercal$, $\boldsymbol{T}_n = (X_n, R_n, Y_n)^\intercal$, and assume the following.

1) $\boldsymbol{T}_1^N$ is Markov.
2) $p(r_{n+1}|\boldsymbol{t}_n) = p(r_{n+1}|r_n)$, which implies the Markovianity of $\boldsymbol{R}_1^N$.
3) $\boldsymbol{Z}_1^N = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N)$ is Gaussian conditionally on $\boldsymbol{R}_1^N$.

Such a model, introduced in [8], is called "conditionally Gaussian Markov switching model" (CGMSM), and is defined by $p(t_1)$, transitions $p(r_{n+1}|r_n)$, and

$$\boldsymbol{Z}_{n+1} = \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{Z}_n + \boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{W}_{n+1} + \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1}) \quad (2)$$

for $n = 1, \ldots, N-1$, and where

1) $\boldsymbol{W}_n = (U_n, V_n)^\intercal$ with $U_1, V_1, \ldots, U_N, V_N$ Gaussian zero-mean independent vectors with identity covariance matrices;
2) Matrices $\boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})$ and $\boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1})$

$$\boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1}) = \begin{bmatrix} a_{n+1}^1(\boldsymbol{r}_n^{n+1}) & a_{n+1}^2(\boldsymbol{r}_n^{n+1}) \\ a_{n+1}^3(\boldsymbol{r}_n^{n+1}) & a_{n+1}^4(\boldsymbol{r}_n^{n+1}) \end{bmatrix}$$

$$\boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1}) = \begin{bmatrix} b_{n+1}^1(\boldsymbol{r}_n^{n+1}) & b_{n+1}^2(\boldsymbol{r}_n^{n+1}) \\ b_{n+1}^3(\boldsymbol{r}_n^{n+1}) & b_{n+1}^4(\boldsymbol{r}_n^{n+1}) \end{bmatrix}$$

3) Means $\boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1}) = (N_{n+1}^X(\boldsymbol{r}_n^{n+1}), N_{n+1}^Y(\boldsymbol{r}_n^{n+1}))^\intercal$ are given by

$$\boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1}) = \boldsymbol{M}_{n+1}(r_{n+1}) - \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{M}_n(r_n)$$

with

$$\boldsymbol{M}_n(r_n) = \mathbb{E}\left[\left(\begin{array}{c} X_n \\ Y_n \end{array}\right) \middle| r_n\right] = \begin{bmatrix} M_n^X(r_n) \\ M_n^Y(r_n) \end{bmatrix}. \quad (3)$$

Recursive filtering is not workable in the general CGMSM. Besides, let us notice that the classic CGLSSM [2] is a particular CGMSM obtained by setting, for each $\boldsymbol{r}_n^{n+1} \in \Omega^2$

$$a_{n+1}^2\left(\boldsymbol{r}_n^{n+1}\right) = a_{n+1}^4\left(\boldsymbol{r}_n^{n+1}\right) = b_{n+1}^2\left(\boldsymbol{r}_n^{n+1}\right) = 0. \quad (4)$$

However, another particular CGMSM, called "CGOMSM" obtained from CGMSM by setting, for each $\boldsymbol{r}_n^{n+1} \in \Omega^2$ [7]–[10]

$$a_{n+1}^3\left(\boldsymbol{r}_n^{n+1}\right) = 0 \tag{5}$$

allows for recursive optimal filtering even with switches [8]. Indeed, CGOMSM belongs to the category of "CMSHLMs" in which recursive optimal filtering is workable [6].

The aim of this paper is to extend the CGOMSM defined by (1)–(3) and (5) to a "fuzzy" CGOMSM (denoted by CGOFMSM) and to show how the related recursive optimal "fuzzy" filter runs.

III. Conditionally Gaussian Observed Fuzzy Markov Switching Model (CGOFMSM)

Let us begin by illustrating with three examples the interest of the new proposed model in real situations.

In the first example, let sequence $\boldsymbol{X}_1^N$ model the positions at time index $1, \ldots, N$ of a flying object, and let sequence $\boldsymbol{Y}_1^N$ model the measurements provided by some optical sensors situated on the ground. During the tracking process, the sunlight can be partially or totally hidden due to the presence of clouds, which gives two models for the distribution of $\boldsymbol{Y}_1^N$. This can be modeled by a "hard" model with $\boldsymbol{R}_1^N$ such that each R_n takes its value in $\Omega = \{0, 1\}$, with 0 corresponds to total sunlight exposure and 1 to shadow condition. In some situations, during cloudy weather conditions that hide the sun partially, the transition from sunlight to shadow is "continuous," and the duration of "intermediary" light can be of paramount importance to the tracking process. This motivates the introduction of "fuzzy" model with each R_n belonging to $\Omega = [0, 1]$ rather than to $\Omega = \{0, 1\}$.

However, the distribution of R_n on $\Omega = [0, 1]$ has to verify some properties. Willing to have non-null probability to have sunshine—and likewise for shadow—implies that there should be two Dirac masses on 0 and 1. Then, one can complete the distribution of R_n on $\Omega = [0, 1]$ by setting continuous probability on $]0, 1[$. Finally, the distribution of R_n is defined by its density $p : [0, 1] \to \mathbb{R}$ with respect to $\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$, where δ_0, δ_1 are Dirac's distributions on 0, 1, and $\mu_{]0,1[}$ is the Lebesgue's measure on $]0, 1[$.

Let us consider a second example dealing with pedestrian tracking for surveillance purposes, which consists in tracking the movements of pedestrians using aggregated data acquired from deployed sensors in the monitored area [19]. Due to the dynamic aspect of pedestrian motion in the presence of several contextual information such as crowd, the use of a two-motion model (corresponding to crowded/uncrowned configurations) is necessary. However, the concept of "crowd" can be seen as a fuzzy phenomenon. Hence, relying on an abrupt change of parameters within the two-jump scheme does not take into account the intermediate states of pedestrian motion and impacts the accuracy of the tracking process. Another example of the same problem relates to car traffic speed and density in a road segment [20].

A last example showing the potential interest of a fuzzy model appears when we want to study the phenomenon associated to outdoor air temperature. Typically, during one day, temperature reaches minimal values during the night and maximal during the afternoon. Between these two ranges, temperatures increase and decrease and can be represented by the fuzzy nature of the jumps considered in our model. An example of such a situation is detailed in Section VII.

From this perspective, the use of fuzzy transitions to model the transient change of parameters is more relevant than the salient switching model. The definition of the new "CGOFMSM" we propose is similar to (1)–(3) and (5), except that we limit our study to two hard classes and each R_n takes its values in $\Omega = [0, 1]$.

Definition 1: Let $\boldsymbol{X}_1^N$, $\boldsymbol{Y}_1^N$, and $\boldsymbol{R}_1^N$ be three stochastic sequences of random variables taking their values in $\mathbb{R}$, $\mathbb{R}$, and $[0, 1]$, respectively. The triplet $\boldsymbol{T}_1^N = (\boldsymbol{T}_1, \ldots, \boldsymbol{T}_N)$, with $\boldsymbol{T}_n = (X_n, R_n, Y_n)^\intercal$, will be said "CGOFMSM" if

1) $\boldsymbol{T}_1^N$ verifies (1)–(3) and (5);
2) The distribution of each R_n is defined by a density (possibly depending on n) $p : [0, 1] \to \mathbb{R}$ with respect to $\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$, where δ_0, δ_1 are Dirac's distributions on 0, 1, and $\mu_{]0,1[}$ is the Lebesgue's measure on $]0, 1[$.

Let us recall some basic rules for integrating a function with respect to $\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$. Such integration has two components: sum of its values on 0, 1, and "classic" integration over $]0, 1[$. More precisely, for any function $\phi : [0, 1] \to \mathbb{R}$, we have

$$\begin{aligned}\int_0^1 \phi(r) d\nu(r) &= \int_0^1 \phi(r)(\delta_0 + \delta_1 + \mu_{]0,1[}) \\ &= \phi(0) + \phi(1) + \int_0^1 \phi(r) dr.\end{aligned} \tag{6}$$

In particular, the expectation of $\phi(R_n)$ is written

$$\begin{aligned}\mathbb{E}\left[\phi(r_n)\right] &= \int_0^1 \phi(r) p\left(r\right) d\nu(r) = \phi(0) p\left(0\right) \\ &\quad + \phi(1) p\left(1\right) + \int_0^1 \phi(r) p\left(r\right) dr.\end{aligned} \tag{7}$$

The distribution of an FMC $\boldsymbol{R}_1^N$ is defined by the density $p\left(r_1\right)$ and the conditional densities $p\left(r_{n+1} \left| r_n\right.\right)$. All of them are thus defined on $\Omega = [0, 1]$ and are densities w.r.t. $\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$. According to the general integration w.r.t. ν rule in (6), we have

$$\int_0^1 p\left(r_1\right) d\nu(r_1) = p\left(0\right) + p\left(1\right) + \int_0^1 p\left(r_1\right) dr_1 = 1 \tag{8}$$

and

$$\begin{aligned}p\left(r_{n+1}\right) &= \int_0^1 p\left(r_n\right) p\left(r_{n+1} \left| r_n\right.\right) d\nu(r_n) \\ &= p\left(0\right) p\left(r_{n+1} \left| 0\right.\right) + p\left(1\right) p\left(r_{n+1} \left| 1\right.\right) \\ &\quad + \int_0^1 p\left(r_n\right) p\left(r_{n+1} \left| r_n\right.\right) dr_n.\end{aligned} \tag{9}$$

Finally, optimal filtering in "fuzzy" CGOFMSM is not very different from that in "hard" CGOMSM, the difference being that, in CGOMSM, integrating with respect to r_n consists in summing, while in CGOFMSM, it consists of integrating with respect to ν.

IV. Optimal Fuzzy Switching Recursive Filter (OFSRF)

We wish to compute $p(r_{n+1}|\boldsymbol{y}_1^{n+1})$, $\mathbb{E}[X_{n+1}|r_{n+1},\boldsymbol{y}_1^{n+1}]$, and $\mathbb{E}[X_{n+1}^2|r_{n+1},\boldsymbol{y}_1^{n+1}]$ from $p(r_n|\boldsymbol{y}_1^n)$, $\mathbb{E}[X_n|r_n,\boldsymbol{y}_1^n]$, $\mathbb{E}[X_n^2|r_n,\boldsymbol{y}_1^n]$, and y_{n+1}. According to (2) and (5), $(\boldsymbol{R}_1^N,\boldsymbol{Y}_1^N)$ is a hidden Markov chain, which makes possible the computation of $p\left(r_{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)$ as explained below.

First, let us note that the probabilities

$$p\left(\boldsymbol{r}_n^{n+1},y_{n+1}\left|\boldsymbol{y}_1^n\right.\right) = p\left(y_{n+1}\left|\boldsymbol{r}_n^{n+1},y_n\right.\right)p\left(r_n\left|\boldsymbol{y}_1^n\right.\right)p\left(r_{n+1}\left|r_n\right.\right) \quad (10)$$

can be calculated since

1) $p(r_{n+1}|r_n)$ are given;
2) $p(r_n|\boldsymbol{y}_1^n)$ can be calculated from(13);
3) $p(y_{n+1}|\boldsymbol{r}_n^{n+1},y_n)$ are conditional densities of the multivariate Gaussians defined by (2). Taking into account (5), by using [21, Sec. 8.1.3, p. 40], we obtain means and variances of $a_{n+1}^4(\boldsymbol{r}_n^{n+1})(y_n - M_n^Y(r_n)) + M_{n+1}^Y(r_{n+1})$ and $(b_{n+1}^3(\boldsymbol{r}_n^{n+1}))^2 + (b_{n+1}^4(\boldsymbol{r}_n^{n+1}))^2$, respectively.

Second, the following probabilities:

$$p\left(y_{n+1}\left|\boldsymbol{y}_1^n\right.\right) = \iint_0^1 p\left(\boldsymbol{r}_n^{n+1},y_{n+1}\left|\boldsymbol{y}_1^n\right.\right)\left(d\nu(r_n)\otimes d\nu(r_{n+1})\right) \quad (11)$$

can also be computed accordingly

$$p\left(\boldsymbol{r}_n^{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right) = \frac{p\left(\boldsymbol{r}_n^{n+1},y_{n+1}\left|\boldsymbol{y}_1^n\right.\right)}{p\left(y_{n+1}\left|\boldsymbol{y}_1^n\right.\right)}. \quad (12)$$

Finally, using (10), (12) gives the so-called forward probabilities

$$p\left(r_{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right) = \int_0^1 p\left(\boldsymbol{r}_n^{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)d\nu(r_n) \times \frac{\int_0^1 p\left(\boldsymbol{r}_n^{n+1},y_{n+1}\left|\boldsymbol{y}_1^n\right.\right)d\nu(r_n)}{p\left(y_{n+1}\left|\boldsymbol{y}_1^n\right.\right)}. \quad (13)$$

In addition, for the later use, note that

$$p\left(r_n\left|r_{n+1},\boldsymbol{y}_1^{n+1}\right.\right) = \frac{p\left(\boldsymbol{r}_n^{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)}{p\left(r_{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)}. \quad (14)$$

The "OFSRF" we proposed consists of five steps outlined as follows. To start the iterations, we first use the distribution of $\boldsymbol{T}_1$. It is then possible to run the OFSRF iterations, assuming that all quantities have been computed for sample n

1) Compute $p\left(r_{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)$ with (12) and (13).
2) Compute $\mathbb{E}[\boldsymbol{Z}_{n+1}|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n]$ and $\mathrm{Var}[\boldsymbol{Z}_{n+1}|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n]$. From (2), we have

$$\mathbb{E}\left[\boldsymbol{Z}_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right] = \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right] + \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1}). \quad (15)$$

Recalling that R_{n+1} and $\boldsymbol{Z}_n$ are independent conditionally on R_n (Condition 2 in the definition of CGMSM), we have

$$\mathbb{E}\left[\boldsymbol{Z}_n\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right] = \begin{bmatrix}\mathbb{E}\left[X_n\middle|r_n,\boldsymbol{y}_1^n\right]\\ y_n\end{bmatrix}.$$

In addition, using (2) and from classical calculations detailed in Appendix C, we have

$$\begin{aligned}\mathrm{Var}&\left[\boldsymbol{Z}_{n+1}\left|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right.\right]\\ &= \boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{B}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1})\\ &\quad + \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathrm{Var}\left[\boldsymbol{Z}_n\left|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right.\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1})\\ &= \boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{B}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1})\\ &\quad + \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathrm{Var}\left[\boldsymbol{Z}_n\left|r_n,\boldsymbol{y}_1^n\right.\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}).\end{aligned} \quad (16)$$

For the later convenience, let us note

$$\mathrm{Var}\left[\boldsymbol{Z}_{n+1}\left|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right.\right] = \begin{bmatrix}\alpha_{n+1}(\boldsymbol{r}_n^{n+1}) & \beta_{n+1}(\boldsymbol{r}_n^{n+1})\\ \xi_{n+1}(\boldsymbol{r}_n^{n+1}) & \delta_{n+1}(\boldsymbol{r}_n^{n+1})\end{bmatrix}.$$

3) From the multivariate normal distribution specified by (15) and (16), compute the parameters $\mathbb{E}[X_{n+1}|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}]$ and $\mathbb{E}[X_{n+1}^2|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}]$ of its marginal $X_{n+1}|y_{n+1},\boldsymbol{r}_n^{n+1}$ (see [21, Sec. 8.1.3, p. 40]) according to

$$\begin{aligned}\mathbb{E}&\left[X_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}\right]\\ &= \mathbb{E}\left[X_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right] + \frac{\beta_{n+1}(\boldsymbol{r}_n^{n+1})}{\delta_{n+1}(\boldsymbol{r}_n^{n+1})}\\ &\quad\times\left(y_{n+1} - \mathbb{E}\left[Y_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n\right]\right)\end{aligned} \quad (17)$$

with $\mathbb{E}[X_{n+1}|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n]$ and $\mathbb{E}[Y_{n+1}|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^n]$ given by (15), and

$$\begin{aligned}\mathbb{E}&\left[X_{n+1}^2\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}\right]\\ &= \mathbb{E}^2\left[X_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}\right] + \alpha_{n+1}(\boldsymbol{r}_n^{n+1})\\ &\quad - \frac{\beta_{n+1}(\boldsymbol{r}_n^{n+1})}{\delta_{n+1}(\boldsymbol{r}_n^{n+1})}\xi_{n+1}(\boldsymbol{r}_n^{n+1}).\end{aligned} \quad (18)$$

4) Compute $\mathbb{E}[X_{n+1}|r_{n+1},\boldsymbol{y}_1^{n+1}]$ and $\mathbb{E}[X_{n+1}^2|r_{n+1},\boldsymbol{y}_1^{n+1}]$ using (14) with

$$\mathbb{E}\left[X_{n+1}\middle|r_{n+1},\boldsymbol{y}_1^{n+1}\right] = \int_0^1 \mathbb{E}\left[X_{n+1}\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}\right]p \times\left(r_n\left|r_{n+1},\boldsymbol{y}_1^{n+1}\right.\right)d\nu(r_n) \quad (19)$$

$$\mathbb{E}\left[X_{n+1}^2\middle|r_{n+1},\boldsymbol{y}_1^{n+1}\right] = \int_0^1 \mathbb{E}\left[X_{n+1}^2\middle|\boldsymbol{r}_n^{n+1},\boldsymbol{y}_1^{n+1}\right]p \times\left(r_n\left|r_{n+1},\boldsymbol{y}_1^{n+1}\right.\right)d\nu(r_n). \quad (20)$$

(5) Finally, compute the filtering equations

$$\mathbb{E}\left[X_{n+1}\middle|\boldsymbol{y}_1^{n+1}\right] = \int_0^1 \mathbb{E}\left[X_{n+1}\middle|r_{n+1},\boldsymbol{y}_1^{n+1}\right]p\left(r_{n+1}\left|\boldsymbol{y}_1^{n+1}\right.\right)d\nu(r_{n+1}) \quad (21)$$

and

$$\mathbb{E}\left[X_{n+1}^2\middle|\,\boldsymbol{y}_1^{n+1}\right] = \int_0^1 \mathbb{E}\left[X_{n+1}^2\middle|\,r_{n+1},\boldsymbol{y}_1^{n+1}\right] p\left(r_{n+1}\middle|\,\boldsymbol{y}_1^{n+1}\right) d\nu(r_{n+1}). \tag{22}$$

Remark 1: Integration with respect to ν above cannot be written in a closed-form formula. It is, then, approximated by numerical integration. Let F denote the number of discrete steps used to compute integrals on $]0,1[$. The impact of F on the restoration results will be discussed in the experimental section.

V. Model Parametrization

In order to assess the interest of the filtering algorithm on CGOFMSM simulated data, we will consider stationary CGOFMSM models, with distribution defined by $p\left(\boldsymbol{x}_1^2,\boldsymbol{r}_1^2,\boldsymbol{y}_1^2\right) = p\left(\boldsymbol{r}_1^2\right) p\left(\boldsymbol{x}_1^2,\boldsymbol{y}_1^2\middle|\,\boldsymbol{r}_1^2\right)$. Thus, we have to define $p\left(\boldsymbol{r}_1^2\right)$ (see Section V-A) and $p\left(\boldsymbol{x}_1^2,\boldsymbol{y}_1^2\middle|\,\boldsymbol{r}_1^2\right)$ (see Section V-B). Thanks to the particular structure of ν, this can be done in such a way that when the fuzziness disappears, it is to say when $p(r_n) = 0$ on $]0,1[$ for $n = 1,\ldots,N$, a CGOFMSM becomes a classical CGOMSM.

A. Distribution of (R_1, R_2)

Let us notice that in the "hard" case with two possible switches, the distribution $P_{(R_1,R_2)}$ is simply a probability over $\{0,1\}^2$. In the fuzzy case we deal with, it is a distribution on $[0,1]^2$, which provides a wide range of possibilities for choosing its shape. We next describe two possible shapes of interest for $P_{(R_1,R_2)}$ (called FMC1 and FMC2 models, where FMC stands for "Fuzzy Markov Chain"), that will be experimented in the next section.

1) First Case (FMC1 Model): The density $p\left(\boldsymbol{r}_1^2\right)$ of $P_{(\boldsymbol{R}_1^2)}$ w.r.t. $\nu\otimes\nu$—where $\nu = \delta_0 + \delta_1 + \mu_{]0,1[}$—is of the form

$$\begin{aligned} &p(0,0) = p(1,1) = \alpha \\ &p(1,0) = p(0,1) = \beta \\ &p(r_1,r_2) = \eta + (\delta-\eta)\,|r_1 - r_2| \\ &\text{for } (r_1,r_2)\in[0,1]^2\setminus\{0,1\}^2 \end{aligned}$$

with $\iint_0^1 p(r_1,r_2)\, d\nu(r_1)\, d\nu(r_2) = 1$. A possible shape for this density is illustrated in Fig. 2.

Remark 2: We obtain a "fuzzy constant" model by setting $\delta = \eta$ and, in particular, we get a "purely hard" CGOMSM model by setting $\delta = \eta = 0$.

The density $p(r_1)$ of $P_{(R_1)}$ is computed as follows:

$$p(r_1) = \begin{cases} \alpha+\beta+\frac{\delta+\eta}{2} & \text{if } r_1 = 0 \\ \alpha+\beta+\frac{\delta+\eta}{2} & \text{if } r_1 = 1 \\ \frac{3}{2}(\delta+\eta) + (\delta-\eta)(r_1^2 - r_1) & \text{if } r_1 \in]0,1[\end{cases}. \tag{23}$$

Knowing that $\int_0^1 p(r_1)\, d\nu(r_1) = 1$, we get

$$\beta = \frac{1 - \frac{5}{2}(\delta+\eta) + \frac{1}{6}(\delta-\eta)}{2} - \alpha. \tag{24}$$

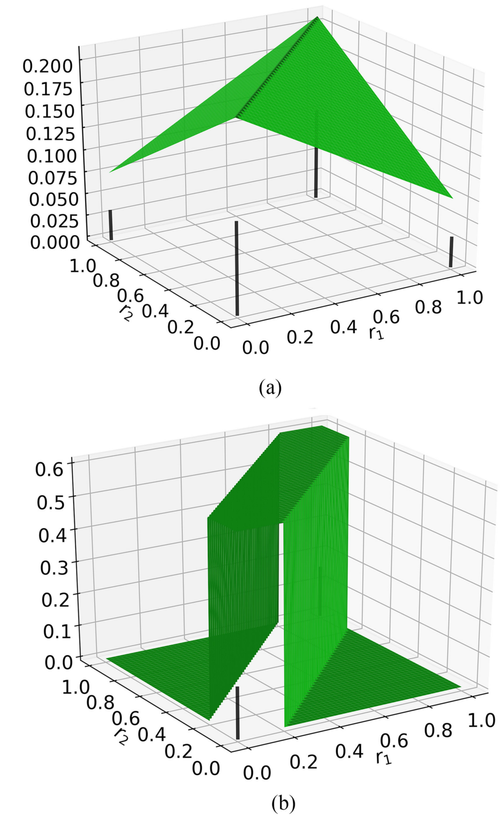

Fig. 2. Density $p(r_1,r_2)$ for (a) the FMC1 model [parameters: $\alpha = 0.10$, $\eta = 0.21$, $\delta = 0.076$ ($\beta = 0.03$, $p_H = 0.55$)] and (b) the FMC2 model [parameters: $\alpha = 0.15$, $\gamma = 0.60$, $\delta = 0.20$ ($\beta = 0.0$, $p_H = 0.54$).].

Hence, this model is only parametrized by $\{\alpha,\delta,\eta\}$ (the calculations are detailed in Appendix A).

The limit proportion of hard data (p_H) with respect to fuzzy ones (p_F) in a sampled sequence is

$$\begin{aligned} p_H &= p(0) + p(1) = 2(\alpha+\beta) + (\delta+\eta) \\ p_F &= 1 - p_H = \frac{3}{2}(\delta+\eta) - \frac{1}{6}(\delta-\eta). \end{aligned} \tag{25}$$

The density $p(r_2|r_1)$ of distribution $P_{R_2|R_1}$, w.r.t. ν, is the ratio between the joint density and the marginal density. We have to distinguish between different cases, according to the value of r_1

$$p(r_2|r_1 = 0) = \begin{cases} \frac{\alpha}{D_1} & \text{if } r_2 = 0 \\ \frac{\beta}{D_1} & \text{if } r_2 = 1 \\ \frac{\eta+(\delta-\eta)r_2}{D_1} & \text{if } r_2 \in]0,1[\end{cases} \tag{26}$$

$$p(r_2|r_1 = 1) = \begin{cases} \frac{\beta}{D_1} & \text{if } r_2 = 0 \\ \frac{\alpha}{D_1} & \text{if } r_2 = 1 \\ \frac{\delta+(\eta-\delta)r_2}{D_1} & \text{if } r_2 \in]0,1[\end{cases} \tag{27}$$

$$p(r_2|r_1 \in]0,1[) = \begin{cases} \frac{\eta+(\delta-\eta)r_1}{D_2} & \text{if } r_2 = 0 \\ \frac{\delta+(\eta-\delta)r_1}{D_2} & \text{if } r_2 = 1 \\ \frac{\eta+(\delta-\eta)|r_1-r_2|}{D_2} & \text{if } r_2 \in]0,1[\end{cases} \tag{28}$$

with $D_1 = \alpha + \beta + \frac{\delta+\eta}{2}$ and $D_2 = \frac{3}{2}(\delta + \eta) + (\delta - \eta)(r_1^2 - r_1)$.

For the particular case where $\delta = \eta = 0$, we have a classical Markov chain with two states, and $p(r_2|r_1)$ writes

$$p(r_2|r_1 = 0) = \begin{cases} \frac{\alpha}{\alpha+\beta} & \text{if } r_2 = 0 \\ \frac{\beta}{\alpha+\beta} & \text{if } r_2 = 1 \end{cases} \tag{29}$$

$$p(r_2|r_1 = 1) = \begin{cases} \frac{\beta}{\alpha+\beta} & \text{if } r_2 = 0 \\ \frac{\alpha}{\alpha+\beta} & \text{if } r_2 = 1 \end{cases}. \tag{30}$$

2) Second Case (FMC2 Model): The density $p\left(\boldsymbol{r}_1^2\right)$ of $P_{(\boldsymbol{R}_1^2)}$ w.r.t. $\nu \otimes \nu$ is of the form

$$\begin{aligned} &p(0,0) = p(1,1) = \alpha \\ &p(1,0) = p(0,1) = \beta \\ &\gamma \text{ for } r_1, r_2 \in [0,1]^2 \setminus \{0,1\}^2 \text{ and } -\delta \le r_2 - r_1 \le \delta \\ &0 \text{ elsewhere} \end{aligned}$$

with $\alpha, \beta \ge 0$, $0 \le \delta < \frac{1}{2}$, and under the constraint that $\iint_0^1 p(r_1, r_2)\, d\nu(r_1)\, d\nu(r_2) = 1$. A possible shape for this density is illustrated in Fig. 2. By varying δ, this model allows expressing transient fuzzy changes.

Remark 3: If $\alpha + \beta = \frac{1}{2}$, then $\gamma = 0$, and the joint law is only made of the four Dirac's distributions at the four corners, which gives a CGOMSM.

The density $p(r_1)$ of $P_{(R_1)}$ is computed as follows:

$$p(r_1) = \begin{cases} \alpha + \beta + \gamma\delta & \text{if } r_1 = 0 \\ \gamma(\delta + r_1 + 1) & \text{if } r_1 \in]0, \delta] \\ 2\gamma\delta & \text{if } r_1 \in]\delta, 1 - \delta[\\ \gamma(2 + \delta - r_1) & \text{if } r_1 \in [1 - \delta, 1[\\ \alpha + \beta + \gamma\delta & \text{if } r_1 = 1 \end{cases}. \tag{31}$$

Since $\int_0^1 p(r_1)\, d\nu(r_1) = 1$, we get

$$\beta = \frac{1 - \gamma M}{2} - \alpha \tag{32}$$

with $M = \delta(6 - \delta)$, and under the constraint that $\gamma \le \frac{1-2\alpha}{M}$. Hence, this model is only parametrized by (α, γ, δ).

The limit proportion of hard data (p_H) to fuzzy ones (p_F) in a sampled sequence is

$$\begin{aligned} p_H &= p(0) + p(1) = 2(\alpha + \beta + \gamma\delta) \\ p_F &= 1 - p_H = \gamma\delta(4 - \delta). \end{aligned} \tag{33}$$

The density $p(r_2|r_1)$ of distribution $P_{R_2|R_1}$, w.r.t. ν, is the ratio between the joint density and the marginal density. Similarly to the first case, we have to distinguish between different configurations, according to the value of r_1

$$p(r_2|r_1 = 0) = \begin{cases} \frac{\alpha}{\alpha+\beta+\gamma\delta} & \text{if } r_2 = 0 \\ \frac{\gamma}{\alpha+\beta+\gamma\delta} & \text{if } r_2 \in]0, \delta] \\ \frac{\beta}{\alpha+\beta+\gamma\delta} & \text{if } r_2 = 1 \\ 0 & \text{elsewhere} \end{cases} \tag{34}$$

$$p(r_2|r_1 \in]0, \delta]) = \begin{cases} \frac{1}{\delta+r_1+1} & \text{if } r_2 = 0 \\ \frac{1}{\delta+r_1+1} & \text{if } r_2 \in]0, r_1 + \delta] \\ 0 & \text{elsewhere} \end{cases} \tag{35}$$

$$p(r_2|r_1 \in]\delta, 1 - \delta[) = \begin{cases} \frac{1}{2\delta} & \text{if } r_2 \in]r_1 - \delta, r_1 + \delta[\\ 0 & \text{elsewhere} \end{cases} \tag{36}$$

$$p(r_2|r_1 \in [1 - \delta, 1[) = \begin{cases} \frac{1}{2+\delta-r_1} & \text{if } r_2 \in [r_1 - \delta, 1[\\ \frac{1}{2+\delta-r_1} & \text{if } r_2 = 1 \\ 0 & \text{elsewhere} \end{cases} \tag{37}$$

$$p(r_2|r_1 = 1) = \begin{cases} \frac{\beta}{\alpha+\beta+\gamma\delta} & \text{if } r_2 = 0 \\ \frac{\gamma}{\alpha+\beta+\gamma\delta} & \text{if } r_2 \in [1 - \delta, 1[\\ \frac{\alpha}{\alpha+\beta+\gamma\delta} & \text{if } r_2 = 1 \\ 0 & \text{elsewhere} \end{cases}. \tag{38}$$

Remark 4: In the two examples of fuzzy Markov models, we assume that the distribution $P_{(R_1,R_2)}$ is defined by four masses on the corners, i.e., at locations $\{0,1\} \times \{0,1\}$, and a density on the remaining $[0,1]^2 \setminus \{0,1\}^2$. It is to say that we assume the distributions on the sides of the square $[0,1] \times [0,1]$, *i.e.*, the distributions $P_{(0,R)}, P_{(1,R)}, P_{(R,0)}$, and $P_{(R,1)}$ for R in [0,1], to be identical to the inner density of the square. This is a particular case, used here to simplify the parametrization, since we can set them independently.

B. Distributions of $(\boldsymbol{X}_1^2, \boldsymbol{Y}_1^2)$ Conditional on $\boldsymbol{R}_1^2$

To finalize the description of stationary $(\boldsymbol{X}_1^N, \boldsymbol{R}_1^N, \boldsymbol{Y}_1^N)$, we need to define the four-dimensional Gaussian distributions $p\left(\boldsymbol{x}_1^2, \boldsymbol{y}_1^2 \middle| \boldsymbol{r}_1^2\right) = p\left(\boldsymbol{z}_1^2 \middle| \boldsymbol{r}_1^2\right)$ for $r_1, r_2 \in [0,1]$. The means and covariance matrices of the four "hard Gaussians" corresponding to $r_1 = i, r_2 = j$, with $i, j \in \{0,1\}$, are given by

$$\mu_{i,j} = \mathbb{E}\left[\begin{matrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{matrix} \middle| r_1 = i, r_2 = j\right] = \begin{bmatrix} \mathbb{E}\left[\begin{matrix} x_1 \\ y_1 \end{matrix} \middle| r_1 = i\right] \\ \mathbb{E}\left[\begin{matrix} x_2 \\ y_2 \end{matrix} \middle| r_2 = j\right] \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_i \\ \boldsymbol{M}_j \end{bmatrix} \tag{39}$$

and

$$\boldsymbol{\Gamma}_{i,j} = \begin{bmatrix} (\sigma_i^X)^2 & b_i & a_{ij} & d_{ij} \\ b_i & (\sigma_i^Y)^2 & e_{ij} & c_{ij} \\ a_{ij} & e_{ij} & (\sigma_j^X)^2 & b_j \\ d_{ij} & c_{ij} & b_j & (\sigma_j^Y)^2 \end{bmatrix} \tag{40}$$

with $d_{ij} = b_i c_{ij}$ in order to verify (5), i.e., for the model to be a CGOMSM.

The mean of a "fuzzy Gaussians" with $r_1 \in]0,1[$ is defined by linear interpolation of $\boldsymbol{M}_i$ and $\boldsymbol{M}_j$

$$\boldsymbol{M}_{r_1} = (1 - r_1)\boldsymbol{M}_0 + r_1 \boldsymbol{M}_1 \tag{41}$$

and its covariance matrix with $r_1, r_2 \in]0,1[$ is defined by bilinear interpolation of the "hard covariance matrices" $\mathbf{\Gamma}_{i,j}$

$$\mathbf{\Gamma}_{\boldsymbol{r}_1^2} = (1-r_1)(1-r_2)\mathbf{\Gamma}_{0,0} + r_1 r_2 \mathbf{\Gamma}_{1,1} + r_1(1-r_2)\mathbf{\Gamma}_{1,0} + r_2(1-r_1)\mathbf{\Gamma}_{0,1}. \tag{42}$$

Then, according to (2), we have

$$\boldsymbol{Z}_{n+1} = \boldsymbol{A}(\boldsymbol{r}_n^{n+1})\boldsymbol{Z}_n + \boldsymbol{B}(\boldsymbol{r}_n^{n+1})\boldsymbol{W}_{n+1} + \boldsymbol{N}(\boldsymbol{r}_n^{n+1}) \tag{43}$$

with $\boldsymbol{N}(\boldsymbol{r}_n^{n+1}) = \boldsymbol{M}_{r_{n+1}} - \boldsymbol{A}(\boldsymbol{r}_n^{n+1})\boldsymbol{M}_{r_n}$, and

$$\boldsymbol{A}(\boldsymbol{r}_n^{n+1}) = \begin{bmatrix} a_{\boldsymbol{r}_n^{n+1}} & e_{\boldsymbol{r}_n^{n+1}} \\ d_{\boldsymbol{r}_n^{n+1}} & c_{\boldsymbol{r}_n^{n+1}} \end{bmatrix} \begin{bmatrix} (\sigma_{r_n}^X)^2 & b_{r_n} \\ b_{r_n} & (\sigma_{r_n}^Y)^2 \end{bmatrix}^{-1}$$

$$\boldsymbol{B}(\boldsymbol{r}_n^{n+1}) = \begin{bmatrix} (\sigma_{r_{n+1}}^X)^2 & b_{r_{n+1}} \\ b_{r_{n+1}} & (\sigma_{r_{n+1}}^Y)^2 \end{bmatrix}^{-1} \tag{44}$$

$$- \boldsymbol{A}(\boldsymbol{r}_n^{n+1}) \begin{bmatrix} a_{\boldsymbol{r}_n^{n+1}} & d_{\boldsymbol{r}_n^{n+1}} \\ e_{\boldsymbol{r}_n^{n+1}} & c_{\boldsymbol{r}_n^{n+1}} \end{bmatrix}. \tag{45}$$

Additionally, for the later use, note that, according to (43), we have

$$p\left(\boldsymbol{z}_{n+1} \middle| \boldsymbol{z}_n, \boldsymbol{r}_n^{n+1}\right) = \mathcal{N}\left(\boldsymbol{A}(\boldsymbol{r}_n^{n+1})(\boldsymbol{z}_n - \boldsymbol{M}_{r_n}) + \boldsymbol{M}_{r_{n+1}}, \boldsymbol{B}(\boldsymbol{r}_n^{n+1})\boldsymbol{B}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1})\right). \tag{46}$$

Remark 5: Fuzzy means (41) and variances (42) can be seen as linear interpolations of hard ones. Other kind of interpolations can be used, once they are compatible with the "hard" CGOMSM.

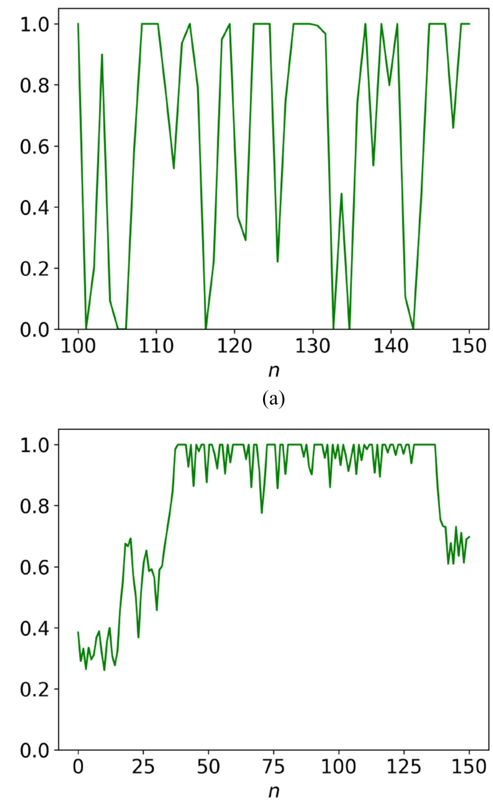
(a)
(b)

Fig. 3. Two excerpt of trajectories simulated from the two fuzzy models with the parameters of Fig. 2 for the FMC1 model (a) and for the FMC2 model (b).

VI. Experimental Studies

Experiments below report results of restoration on simulated data—using the two joint laws presented in the previous section—in order to measure the quality of the OFSRF restoration algorithm, and the influence of the number of discretization steps F in approximation of integrals. Comparison is also performed with the "hard" CGOMSM to measure the error in using this model when there exist transient changes in data.

A. Simulations of a CGOFMSM Sample

To simulate a CGOFMSM ($\boldsymbol{X}_1^N$, $\boldsymbol{R}_1^N$, $\boldsymbol{Y}_1^N$) sample, we first simulate an FMC $\boldsymbol{r}_1^N$, and then simulate the observations and states $(\boldsymbol{y}_1^N, \boldsymbol{x}_1^N)$.

To draw a sample from the FMC, we first simulate r_1 using $p(r_1)$, and then, for each $n = 1, \ldots, N-1$, r_{n+1} is obtained from $p(r_{n+1} | r_n)$, which is equal to $p(r_2 | r_1)$. For each r_{n+1} to be simulated, we first decide if the jump will be a "hard" one or a "fuzzy" one. To do so, a draw is performed beforehand according to the proportion of hard and fuzzy jumps. Let us explain in detail the simulation process based on the FMC2 model, the procedure is the same for the FMC1 model.

1) Simulation of r_1. First, make a draw according to the proportion in (33) to determine if the sample will be a "hard" one or a "fuzzy" one.
   a) If it is hard, then, using (31), make a draw according to the probability vector $[\frac{1}{2}, \frac{1}{2}]$, to determine if the sample is "0" or "1."
   b) If it is fuzzy, then make a draw according to the density specified by the three equations in (31) corresponding to $r_1 \in]0,1[$. The target density is not trivial because of the slopes in its shape, but most of specialized random number generation libraries offer a solution for sampling such density (e.g., library "scipy" in Python language).
2) Simulation of r_{n+1}, knowing r_n. Let us assume $r_n = 0$, the other possible values of r_n can be processed similarly.
   a) According to (34), first make a draw according to the probability vector $[\frac{\alpha}{S}, \frac{\beta}{S}, \frac{\gamma}{S}]$, with $S = \alpha + \beta + \gamma\delta$, to determine if r_{n+1} will be "0," "1" or "fuzzy."
   b) If it is fuzzy, then make a uniform draw on $]0, \delta[$ to get r_{n+1}.

Fig. 3 shows two excerpts of simulated trajectories corresponding to the FMC1 and FMC2 models (green plain line).

Then, knowing (r_n, r_{n+1}) and (x_n, y_n), each pair (y_{n+1}, x_{n+1}) is sampled from Gaussian distributions (46) whose parameters are obtained by (41) and (42). For experiments conducted hereafter, we set the "hard" mean vectors and covariance matrices to be

$$\boldsymbol{M}_0 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \boldsymbol{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$\mathbf{\Gamma}_{0,0} = \begin{bmatrix} 0.5 & 0.3 & 0.1 & 0.06 \\ 0.3 & 1.0 & 0.35 & 0.4 \\ 0.1 & 0.35 & 0.5 & 0.3 \\ 0.06 & 0.4 & 0.3 & 1.0 \end{bmatrix}$$

$$\mathbf{\Gamma}_{0,1} = \begin{bmatrix} 0.5 & 0.3 & 0.5 & 0.14 \\ 0.3 & 1.0 & 0.33 & 0.6 \\ 0.5 & 0.33 & 0.75 & 0.3 \\ 0.14 & 0.6 & 0.3 & 0.5 \end{bmatrix}$$

$$\mathbf{\Gamma}_{1,0} = \begin{bmatrix} 0.75 & 0.3 & 0.1 & 0.24 \\ 0.3 & 0.5 & 0.35 & 0.4 \\ 0.1 & 0.35 & 0.5 & 0.3 \\ 0.24 & 0.4 & 0.3 & 1.0 \end{bmatrix}$$

$$\mathbf{\Gamma}_{1,1} = \begin{bmatrix} 0.75 & 0.3 & 0.5 & 0.18 \\ 0.3 & 0.5 & 0.33 & 0.3 \\ 0.5 & 0.33 & 0.75 & 0.3 \\ 0.18 & 0.3 & 0.3 & 0.5 \end{bmatrix}.$$

B. Restoration Results

The restoration of simulated data is performed according to the OFSRF algorithm detailed in Section IV, from simulated observations only. Fig. 4 shows an example of the restoration of $\boldsymbol{r}_1^N$ and $\boldsymbol{x}_1^N$ for the FMC2 law and when the number of discrete fuzzy jumps are set to $F = 5$. The restoration of $\boldsymbol{r}_1^N$ was obtained by applying maximum posteriori mode (MPM) principle.

1) If $p(r_n = 0 \,|\boldsymbol{y}_1^n) + p(r_n = 1 \,|\boldsymbol{y}_1^n) > 0.5$, then the restoration will be "hard," else it will be "fuzzy."
2) If the restoration is "hard," set $\hat{r}_n$ to be "0" if $p(r_n = 0 \,|\boldsymbol{y}_1^n) > p(r_n = 1 \,|\boldsymbol{y}_1^n)$, else set $\hat{r}_n$ to be "1."
3) If the restoration is "fuzzy," set $\hat{r}_n$ to the discrete fuzzy jump, which maximizes $p(r_n \,|\boldsymbol{y}_1^n)$.

Note however that the restoration of $\boldsymbol{r}_1^N$ is only performed for illustration purpose and is not required for the restoration of $\boldsymbol{x}_1^N$. We can observe the numerical effect of F with the presence of stair-steps in the restoration of jumps in Fig. 4(b). Fig. 4(a) shows the restoration of $\boldsymbol{x}_1^N$ assuming that the jumps are known (i.e., using the simulated FMC for $\boldsymbol{r}_1^N$). Fig. 4(c) assumes that the jumps are unknown (OFSRF algorithm). Additionally, Fig. 4(c) allows to observe the restoration difference between the "classic" CGOMSM and the "fuzzy" CGOFMSM, the latter follows the simulated states better than the first one for $n \leq 30$. This behavior must be compared with that of the fuzzy jumps in Fig. 4(b), which shows a large difference in the restoration of jumps when $n \leq 30$.

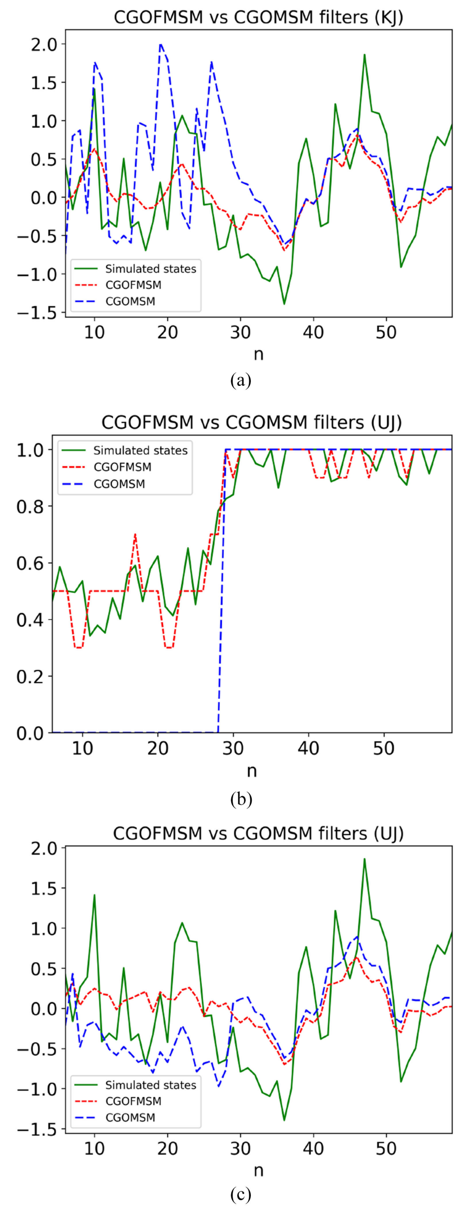

Fig. 4. Comparison of restoration of jumps and states when $F = 5$, for simulated data from the FMC2 law with the same parameters as in Fig. 2 ($N = 80$). "KJ" stands for known jumps and "UJ" for unknown jumps. The difference between the hard filter and the fuzzy one is clearly visible in (c), for $n < 30$. (a) Restoration of states with KJ. The MSEs for CGOFMSM and CGOMSM are 0.35 and 0.75. (b) MPM restoration of fuzzy jumps. The MSEs for CGOFMSM and CGOMSM are 0.03 and 0.11. (c) Restoration of states with UJ. The MSEs for CGOFMSM and CGOMSM are 0.47 and 0.54.

To measure the quality of restorations with respect to F, we compute the mean MSE of 50 independent experiments of $N = 300$ samples, for the FMC1 Markov law. Fig. 5 shows the MSE evolution for increasing values of F for both $\boldsymbol{x}_1^N$ and $\boldsymbol{r}_1^N$. The result of applying the CGOMSM filter on the data is reported in the same graph (horizontal black dotted line, denoted as "Hard filter—UJ"). The parameters used for the hard filters are same as the ones used for the fuzzy filter, except that we "harden" the Markov laws by integrating the fuzzy laws on all four quadrants of $[0, 1]$ to get $P(R_1, R_2)$. In this figure, we can observe that the excess error over the model with known jumps is halved. Additionally, the state MSE reaches its minimum when $F > 3$; this value depends on the fuzzy Markov model and on its parameters. According to some other experiments not reported here, F is always kept relatively small (typically $F \leq 5$), which is of interest since the larger F is, the more the computing time increases. Indeed, the complexity is linear w.r.t. to F, which is to say that the computational burden of CGOFMSM is approximately the same as the one of CGOMSM with $2 + F$ jumps.

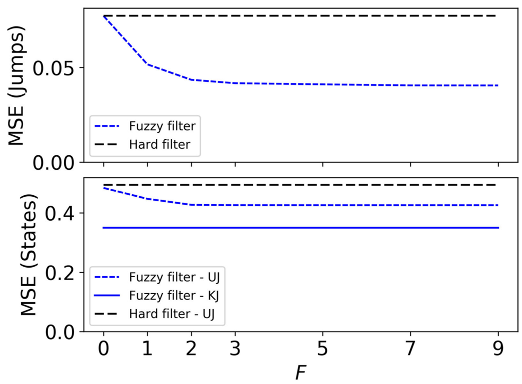

Fig. 5. Evolution of jump (up) and state (down) MSEs according to the value of the discrete fuzzy steps F (means of 50 experiments of $N = 300$ samples) for the FMC1 Markov law. Jumps MSEs were computed from an MPM classification. Parameters: $\alpha = 0.07$, $\eta = 0.16$, $\delta = 0.05$ ($\beta = 0.158$, $p_H = 0.69$).

TABLE I
EVOLUTION OF MSES W.R.T. TO THE PERCENTAGE OF HARD DATA IN THE SAMPLES FOR THE FCM1 ($F = 4$) AND FMC2 ($F = 10$) MODELS

% of 'hard' samples - FMC1	**48%**	**58%**	**67%**	**75%**	**93%**
jumps MSE (CGOFMSM)	0.051	0.046	0.039	0.034	0.017
states MSE (CGOFMSM)	0.475	0.446	0.431	0.415	0.368
jumps MSE (CGOMSM)	0.126	0.094	0.074	0.057	0.020
states MSE (CGOMSM)	0.599	0.535	0.500	0.468	0.386

% of 'hard' samples - FMC2	**48%**	**62%**	**75%**	**87%**	**100%**
jumps MSE (CGOFMSM)	0.030	0.026	0.021	0.014	0.012
states MSE (CGOFMSM)	0.442	0.442	0.420	0.399	0.37
jumps MSE (CGOMSM)	0.070	0.040	0.033	0.021	0.011
states MSE (CGOMSM)	0.538	0.479	0.457	0.422	0.374

Jumps MSEs were computed from an MPM classification. Results are means of 20 independent experiments of $N = 1000$ data.

The last experiment, whose results are reported in Table I for the FMC1 and FMC2 laws, shows the restoration MSE for the CGOMSM and the CGOFMSM filters when the number of fuzzy samples in the simulated data is decreasing (by adjusting parameter' values). We can observe that, for both fuzzy Markov models, the hard filter reaches the performance of the fuzzy filter only when the percent of hard jumps is near 100%. Elsewhere the fuzzy filter provides lower MSE, and the difference can be very large for hard sample rates lower than 50%. This result confirms the interest of the fuzzy filter in comparison with the hard one in the presence of transient changes in observation data.

VII. ILLUSTRATION ON REAL DATA

This section intends to illustrate the behavior of the proposed algorithm when confronted to real data. The experimental data are a time series representing the energy power (in kilowatt) consumed by some building along with the outdoor temperature (in Fahrenheit).[1] We try to understand to what extend the fuzzy model is able to infer the consumed energy from the outdoor temperature for some building during the first week of June 2010, c.f., Fig. 6.

[1]This time series, collected by the American Department of Energy, is open-data and can be downloaded for free. [Online]. Available: https://openei.org/datasets/dataset/consumption-outdoor-air-temperature-11-commercial-buildings. The experiments focus on building #5.

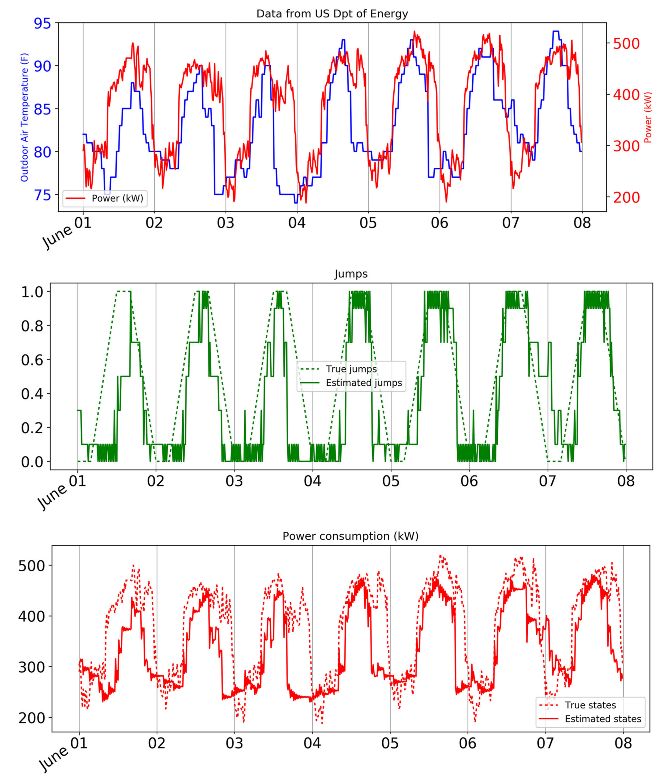

Fig. 6. Outdoor temperature (blue) and energy power consumed (red) by some building during the first week of June 2010 (source: American DOE).

Thus, outdoor temperature is considered as the observation ($\boldsymbol{Y}_1^N$), and the consumed power energy as the state ($\boldsymbol{X}_1^N$), to be estimated, with $N = 672$. The fuzzy filter requires to know the jumps $\boldsymbol{R}_1^N$ to estimate the best-suited parameters for data. Regardless the data, we have fixed that the lowest temperature appears between 1 and 5 am (corresponding to hard jump "0"), and that the highest appear between 1 and 5 pm (corresponding to hard jump "1"). Between these ranges, temperatures increase and decrease linearly and are represented by the fuzzy nature of the jumps considered in this model. The shape of the observations suggests that a fuzzy model is better suited than a hard model (we choose to use the FMC1 model). From this pseudo ground-truth for jumps, using classical empirical estimators, it is possible to estimate all the required parameters of the model: on the one hand, mean values and covariance matrices of ($\boldsymbol{X}_1^N, \boldsymbol{Y}_1^N$) conditionally jumps for the two hard jumps, and, on the other hand, the α, β, η, and δ parameters required to define the law of $\boldsymbol{R}_1^N$.

The results of fuzzy and hard filtering are reported in red in Figs. 7 and 8, respectively. For the fuzzy filtering, we considered $F = 5$ because it gives good results while maintaining low computation times. The MSE of estimated states with respect to the true consumed power energy is 8701 for the hard model, and 6759 for the fuzzy model. Regarding the jumps, the MSE is 0.13 for the hard model, and 0.07 for the fuzzy one. The better results obtained with the fuzzy model w.r.t. the hard one are illustrated by both the estimated jumps and the estimated consumed power energy.

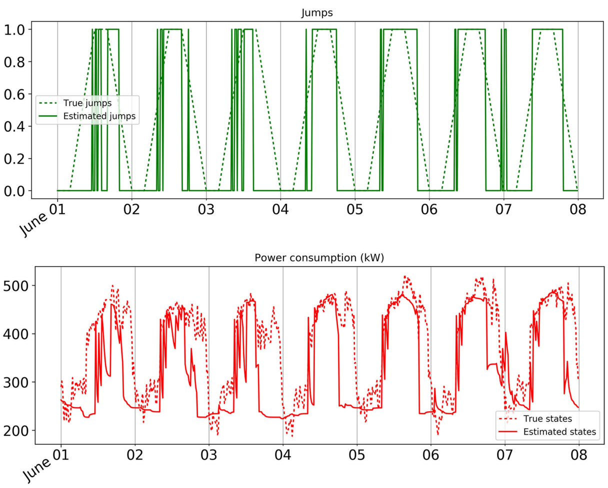

Fig. 7. Filtering result with estimated jumps (up) and estimated states (down) for the fuzzy model with $F = 5$. The MSE for jumps is 0.07, while the MSE for states is 6759.

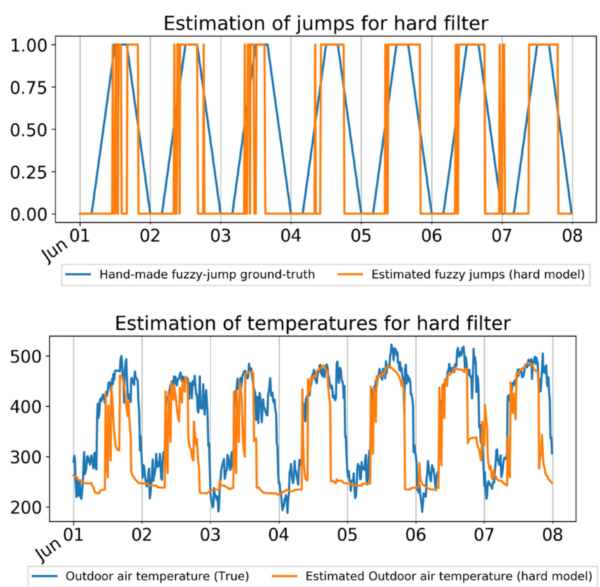

Fig. 8. Filtering result with estimated jumps (up) and estimated states (down) for the hard model. The MSE for jumps is 0.13, while the MSE for states is 8701.

VIII. Conclusion

We proposed a new jump Markov model made of a triplet random process (observations, hidden states, and hidden fuzzy switches), and designed the related optimal recursive fuzzy filter, which is able to restore switches and states from observations. We called this model "CGOFMSM." This paper is based on two key ideas.

1) A recursive and exact filter to deal with hard jumps, called CGOMSM, is available, see [7]–[10]. This filter only assumes the presence of a zero-term (5) in the transition matrix of the very general CGMSM defined by (2).
2) The definition of a mixed measure including two Dirac masses for hard classes "0" or "1" and a Lebesgue measure to deal with fuzziness. It should be noted that integral calculations required some simple and low-time consuming numerical approximation.

We showed through an experimental study that the proposed model and its filter provide interesting results in terms of data restoration accuracy. This behavior is confirmed when the model is confronted to real data dealing with outdoor air temperature. In that case, the fuzzy jumps allow a better modeling of the increasing and decreasing air temperature cycle during one day.

In this paper, we assume scalar states and scalar observations for notation convenience; the extension to a vectorial filter is somewhat straightforward, using matrix products. However, the extension of the filter to three and more classes is not as easy, with quite complex fuzzy Markov laws to deal with, but could be inspired from the work [22]. The next step in the development of an unsupervised parameter estimation method for this CGOFMSM—similar to the one proposed for the "hard" model [23]—is the derivation of a fuzzy smoother for offline processing. Application of the fuzzy model to deal with the design of a control system for road traffic congestion prediction in which traffic dynamics would be modeled by a switching regime model is another perspective for further work.

Appendix
Calculation Regarding the FMC1 Model

Here are the details to specify the margin $p(r_1)$ and the parameter β from the joint density $p(r_1, r_2)$ defined in Section V-A1.

A. *Calculation for Margin Law* $p(r_1)$

The density $p(r_1)$ of distribution P_{R_1}, w.r.t. ν, is obtained by

$$p(r_1) = \int_0^1 p(r_1, r_2)\, d\nu(r_2) = p(r_1, 0) + p(r_1, 1) + \underbrace{\int_0^1 \eta + (\delta - \eta)\,|r_1 - r_2|\ dr_2}_{A(r_1)}$$

with

$$A(r_1) = \eta + (\delta - \eta) \underbrace{\int_0^1 |r_1 - r_2|\, dr_2}_{B(r_1) = B_1(r_1) + B_2(r_1)}$$

and

$$B_1(r_1) = \int_0^{r_1} (r_1 - r_2)\ dr_2 = r_1^2 - \frac{1}{2} r_1^2 = \frac{1}{2} r_1^2$$

$$B_2(r_1) = \int_{r_1}^1 (r_2 - r_1)\ dr_2 = \frac{1}{2}(1 - r_1^2) - r_1(1 - r_1) = \frac{1}{2}(1 + r_1^2) - r_1.$$

Thus, we have $B(r_1) = \frac{1}{2} + r_1^2 - r_1$, and $A(r_1) = \eta + (\delta - \eta)(\frac{1}{2} + r_1^2 - r_1)$.

So, for $r_1 = 0$, $p(r_1) = \alpha + \beta + A(O) = \alpha + \beta + \frac{\delta + \eta}{2}$, for $r_1 = 1$, $p(r_1) = \alpha + \beta + A(1) = \alpha + \beta + \frac{\delta + \eta}{2}$, and for $r_1 \in]0, 1[$

$$p(r_1) = \eta + (\delta - \eta) r_1 + \eta + (\delta - \eta)(1 - r_1) + \eta$$

$$+ (\delta - \eta)\left(\frac{1}{2} + r_1^2 - r_1\right)$$
$$= 3\eta + (\delta - \eta)\frac{3}{2} + (\delta - \eta)(r_1^2 - r_1)$$
$$= \frac{3}{2}(\delta + \eta) + (\delta - \eta)(r_1^2 - r_1).$$

Hence, we get (23).

B. Calculation of β

We have

$$\int_0^1 p(r_1)\, d\nu(r_1) = p(0) + p(1) + \int_0^1 p(r_1)\, dr_1$$
$$= 2(\alpha + \beta) + (\delta + \eta) + \underbrace{\int_0^1 \frac{3}{2}(\delta + \eta) + (\delta - \eta)(r_1^2 - r_1)\ dr_1}_{C = C_1 + C_2}$$

with

$$C_1 = \int_0^1 \frac{3}{2}(\delta + \eta)\, dr_1 = \frac{3}{2}(\delta + \eta)$$
$$C_2 = (\delta - \eta)\int_0^1 \left(r_1^2 - r_1\right)\ dr_1 = -\frac{1}{6}(\delta - \eta).$$

Knowing that $\int_0^1 p(r_1)\, d\nu(r_1) = 1$, we get $2(\alpha + \beta) + (\delta + \eta) + \frac{3}{2}(\delta + \eta) - \frac{1}{6}(\delta - \eta) = 1$, and find result in (24).

C. Calculation to Get (16)

From

$$\mathrm{Var}\left[\boldsymbol{Z}_{n+1}\left|\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right.\right] = \mathbb{E}\left[\boldsymbol{Z}_{n+1}\boldsymbol{Z}_{n+1}^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right] - \mathbb{E}\left[\boldsymbol{Z}_{n+1}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\mathbb{E}\left[\boldsymbol{Z}_{n+1}^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]$$

we have

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{Z}_{n+1}\boldsymbol{Z}_{n+1}^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right] \\
&= \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n\boldsymbol{Z}_n^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{N}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{B}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{B}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) + \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{N}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1})
\end{aligned} \tag{47}$$

and, using (15)

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{Z}_{n+1}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\mathbb{E}\left[\boldsymbol{Z}_{n+1}^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right] \\
&= \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right] \\
&\times E\left[\boldsymbol{Z}_n^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{A}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{N}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1})\mathbb{E}\left[\boldsymbol{Z}_n^{\mathsf{T}}\right|\left.\boldsymbol{r}_n^{n+1}, \boldsymbol{y}_1^n\right]\boldsymbol{A}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}) \\
&+ \boldsymbol{N}_{n+1}(\boldsymbol{r}_n^{n+1})\boldsymbol{N}_{n+1}^{\mathsf{T}}(\boldsymbol{r}_n^{n+1}).
\end{aligned} \tag{48}$$

Substracting (47) and (48) gives (16).

References

[1] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[2] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, Aug. 2003.

[3] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. New York, NY, USA: Springer-Verlag, 2005.

[5] A. Doucet and A. Johansen, *A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later*. London, U.K.: Oxford Univ. Press, 2011.

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

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

[8] N. Abbassi, D. Benboudjema, and W. Pieczynski, "Kalman filtering approximations in triplet Markov Gaussian switching models," in *Proc. IEEE Workshop Statist. Signal Process.*, Nice, France, 28–30 Jun. 2011, pp. 77–80.

[9] N. Abbassi, D. Benboudjema, S. Derrode, and W. Pieczynski, "Optimal filter approximations in conditionally Gaussian pairwise Markov switching models," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1104–1109, Apr. 2015.

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

[11] H. Caillol, A. Hillion, and W. Pieczynski, "Fuzzy random fields and unsupervised image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 4, pp. 801–810, Jul. 1993.

[12] C. Carincotte, S. Derrode, and S. Bourennane, "Unsupervised change detection on SAR images using fuzzy hidden Markov chains," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 432–441, Feb. 2006.

[13] F. Salzenstein, C. Collet, S. L. Cam, and M. Hatt, "Non-stationary fuzzy Markov chain," *Pattern Recog. Lett.*, vol. 28, no. 16, pp. 2201–2208, 2007.

[14] F. Salzenstein and C. Collet, "Fuzzy Markov random fields versus chains for multispectral image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1753–1767, Nov. 2006.

[15] T. Crivelli, B. Cernuschi-Frías, P. Bouthemy, and J. Yao, "Motion textures: Modeling, classification, and segmentation using mixed-state Markov random fields," *SIAM J. Imag. Sci.*, vol. 6, no. 4, pp. 2484–2520, 2013.

[16] T. Crivelli, P. Bouthemy, B. Cernuschi-Frías, and J. Yao, "Simultaneous motion detection and background reconstruction with a conditional mixed-state Markov random field," *Int. J. Comput. Vis.*, vol. 94, no. 3, pp. 295–316, 2011.

[17] X. Xie, D. Yue, and C. Peng, "Multi-instant observer design of discrete-time fuzzy systems: A ranking-based switching approach," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1281–1292, Oct. 2017.

[18] F. Li, P. Shi, C. Lim, and L. Wu, "Fault detection filtering for nonhomogeneous Markovian jump systems via a fuzzy approach," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 131–141, Feb. 2018.

[19] P. Skoglar, U. Orguner, D. Törnqvist, and F. Gustafsson, "Pedestrian tracking with an infrared sensor using road network information," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, Feb. 2012, Art. no. 26.

[20] Z. Bouyahia, H. Haddad, N. Jabeur, and S. Derrode, "Real-time traffic data smoothing from GPS sparse measures using fuzzy switching linear models," in *Proc. 14th Int. Conf. Mobile Syst. Pervasive Comput.*, Leuven, Belgium, Jul. 24–26, 2017, pp. 143–150.

[21] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Nov. 2012, version 20121115. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274

[22] A. Gamal-Eldin, F. Salzenstein, and C. Collet, "Hidden fuzzy Markov chain model with K discrete classes," in *Proc. 10th Int. Conf. Inf. Sci., Signal Process. Appl.*, Kuala Lumpur, Malaysia, May 10–13, 2010, pp. 109–112.

[23] F. Zheng, S. Derrode, and W. Pieczynski, "Parameter estimation in switching Markov systems and unsupervised smoothing," *IEEE Trans. Autom. Control*, vol. 64, no. 4, pp. 1761–1767, Apr. 2019, 10.1109/TAC.2018.2863651.