Fixed-Interval Kalman Smoothing Algorithms in singular state-space systems

Boujemaa Ait-El-Fquih and *François Desbouvries**

GET / INT / dépt. CITI and CNRS UMR 5157

9 rue Charles Fourier, 91011 Evry, France

* *corresponding author*

tel +33 1 60 76 45 27, fax +33 1 60 76 44 33,

Francois.Desbouvries@it-sudparis.eu

Abstract

Fixed-interval Bayesian smoothing in state-space systems has been addressed for a long time. However, as far as the measurement noise is concerned, only two cases have been addressed so far : the regular case, i.e. with positive definite covariance matrix; and the perfect measurement case, i.e. with zero measurement noise. In this paper we address the smoothing problem in the intermediate case where the measurement noise covariance is positive semi definite with arbitrary rank. We exploit the singularity of the model in order to transform the original state-space system into a pairwise Markov model with reduced state dimension. Finally, the a posteriori Markovianity of the reduced state enables us to propose a family of fixed-interval smoothing algorithms.

I. INTRODUCTION

Let us consider the state space system

$$\left\{ \begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{u}_n \\ \mathbf{y}_n & = & \mathbf{H}_n \mathbf{x}_n + \underbrace{\mathbf{J}_n \mathbf{w}_n}_{\mathbf{v}_n} \end{array} \right. , \tag{1}$$

in which $\mathbf{x}_n \in \mathbb{R}^{n_\mathbf{x}}$ is the state, $\mathbf{y}_n \in \mathbb{R}^{n_\mathbf{y}}$ the observation, $\mathbf{u}_n \in \mathbb{R}^{n_\mathbf{u}}$ the process noise and $\mathbf{v}_n \in \mathbb{R}^{n_\mathbf{v}}$ the measurement noise. Processes $\mathbf{u} = \{\mathbf{u}_n\}_{n \in \mathbb{N}}$ and $\mathbf{w} = \{\mathbf{w}_n\}_{n \in \mathbb{N}}$ are zero-mean, independent[1], jointly independent and independent of $\mathbf{x}_0$. So $\mathbf{x}_n$ is a Markov chain (MC) and $(\mathbf{x}_n, \mathbf{y}_n)$ is a hidden Markov chain with independent noise (HMC-IN). Fixed-interval smoothing consists in estimating $\mathbf{x}_n$ from $\mathbf{y}_{0:N}$ for $0 \leq n \leq N$. In the case where the covariance matrices $\mathbf{Q}_n$ of $\mathbf{G}_n\mathbf{u}_n$ and $\mathbf{R}_n$ of $\mathbf{v}_n = \mathbf{J}_n\mathbf{w}_n$ are positive definite ($> \mathbf{0}$), many algorithms have been derived by using such different methods as calculus of variations [1], maximum a posteriori [2] [3], orthogonal projections [4], the innovations approach [5], the two-filter form [6] [7], complementary models [8] or the Bayesian approach [9] [10] (modern surveys can also be found e.g. in [11, ch. 10] [8] or [12]).

The case where $\mathbf{R}_n$ is the null matrix ($\mathbf{R}_n = \mathbf{0}$) has received less attention so far, even though it is of interest in many applications, in particular if the measurement additive noise is colored. Let us briefly summarize the existing contributions. At least two cases can lead to this situation.

- *Unnoisy measurement case ($\mathbf{w}_n = \mathbf{0}$).* The unnoisy measurement case was first addressed by Kalman in his original paper [13], see also [14]. A continuous time BF smoother has also been derived in [15].
- *Autoregressive (AR) measurement noise.* Let

$$\mathbf{w}_n = \mathbf{A}_{n-1}\mathbf{w}_{n-1} + \xi_n \tag{2}$$

(with $\mathbf{w}_0 = \xi_0$), in which ξ_n is zero-mean, and $\mathbf{x}_0$, $\{\mathbf{u}_n\}_{n \geq 0}$ and $\{\xi_n\}_{n \geq 0}$ are independent. Let $\mathbf{x}^*_n$ be the augmented state $\mathbf{x}^*_n = [\mathbf{x}_n^T, \mathbf{w}_n^T]^T$. Then system (1) can be rewritten as the unnoisy measurement system :

$$\begin{cases} \mathbf{x}^*_{n+1} & = \begin{bmatrix} \mathbf{F}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_n \end{bmatrix} \mathbf{x}^*_n + \begin{bmatrix} \mathbf{G}_n\mathbf{u}_n \\ \xi_{n+1} \end{bmatrix} \\ \mathbf{y}_n & = [\mathbf{H}_n \; \mathbf{J}_n] \times \mathbf{x}^*_n \end{cases} . \tag{3}$$

The AR measurement noise case has been addressed in many papers, but up to our best knowledge, three algorithms only have been derived : the Kalman filter (KF) [16] [17] [18] [19] [20] [21], the Rauch-Tung-Streibel (RTS) algorithm [18] [20], and a fixed-lag smoothing (FLS) algorithm [21].

Remark 1: In some particular cases one can avoid increasing the dimension of the state. Let us assume that $\mathbf{J}_n$ is the $n_\mathbf{y} \times n_\mathbf{y}$ identity matrix $\mathbf{I}_{n_\mathbf{y}}$ and that $\mathrm{E}(\xi_n\xi_n^T) > \mathbf{0}$. Then model (1) can be

[1]The independence assumptions in this paper come from our choice to adopt the Bayesian point of view and next derive our smoothing algorithms by injecting the Gaussian assumption. Alternately, we could of course have assumed that the independent processes are uncorrelated only, and derive our algorithms as recursive linear minimum mean square error restoration procedures.

transformed into

$$\begin{cases} \mathbf{x}_n &= \mathbf{F}_{n-1}\mathbf{x}_{n-1} + \mathbf{G}_{n-1}\mathbf{u}_{n-1} \\ \widetilde{\mathbf{y}}_n &= \widetilde{\mathbf{H}}_n\mathbf{x}_{n-1} + \widetilde{\mathbf{v}}_n \end{cases} \tag{4}$$

in which $\widetilde{\mathbf{y}}_n = \mathbf{y}_n - \mathbf{A}_{n-1}\mathbf{y}_{n-1}$, $\widetilde{\mathbf{H}}_n = \mathbf{H}_n\mathbf{F}_{n-1} - \mathbf{A}_{n-1}\mathbf{H}_{n-1}$, and $\widetilde{\mathbf{v}}_n = \mathbf{H}_n\mathbf{G}_{n-1}\mathbf{u}_{n-1} + \xi_n$. Since $\{\widetilde{\mathbf{v}}_n\}_{n\geq 0}$ is independent and $E(\widetilde{\mathbf{v}}_n\widetilde{\mathbf{v}}_n^T) > \mathbf{0}$, (4) is a *regular* state-space system, with the same state $\mathbf{x}_n$ (rather than an augmented one $\mathbf{x}_n^*$), correlated process and measurement noises, and with pseudo-measure $\widetilde{\mathbf{y}}_n$. For model (4) the KF has been derived e.g. in [16] [19] [20] [18], the RTS algorithm in [20] [18], an FLS algorithm in [22] [23], and a two-filter smoother (for the continuous time case) in [24].

Remark 2: In the above models $\mathbf{x}_0$, $\{\mathbf{u}_n\}_{n\geq 0}$ and $\{\xi_n\}_{n\geq 0}$ were assumed to be independent, which implies that $\{\mathbf{u}_n\}_{n\geq 0}$ and $\{\mathbf{w}_n\}_{n\geq 0}$ are independent too. Let us now only assume that $\mathbf{x}_0$ and $\{(\mathbf{u}_n, \xi_{n+1})\}_{n\geq 0}$ are independent, but that the subvectors $\mathbf{u}_n$ and ξ_{n+1} are correlated. These assumptions, in turn, imply that $\mathbf{u}_n$ and $\mathbf{w}_{n+p}$ are correlated for all $p > 0$, which can be of interest in some cases like in aircraft radar guidance systems [25], in which the system and measurement noises come from the same source, and so are correlated. Yet in (3) $\mathbf{x}_n^*$ remains an MC, and (4) is still an HMC-IN, which implies that the algorithms for models (3) and (4) can still be used.

Let us finally consider the general singular case in which $\mathbf{R}_n$ is a positive semi-definite matrix ($\mathbf{R}_n \geq \mathbf{0}$). Up to our best knowledge, only filtering has been addressed in this case [16]. Apart from the case where $\mathbf{J}_n$ is not a full rank matrix, the singular measurements case also occurs when some of the measurements are either unnoisy or colored :

- *Partially unnoisy measurements case.* Let $\mathbf{y}_n = [y_n^1 \cdots y_n^{n_\mathbf{y}}]^T$. If some components $\{y_n^k\}_{k\in K}$ are unnoisy then $\mathbf{R}_n \geq 0$.
- *Partially AR measurement noise.* Let some of the components be colored. Without loss of generality we thus assume that

$$\mathbf{J}_n\mathbf{w}_n = [\mathbf{J}_n^1\mathbf{J}_n^2] \begin{bmatrix} \mathbf{w}_n^1 \\ \mathbf{w}_n^2 \end{bmatrix}, \begin{bmatrix} \mathbf{w}_n^1 \\ \mathbf{w}_n^2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{n-1}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{n-1}^1 \\ \mathbf{w}_{n-1}^2 \end{bmatrix} + \underbrace{\begin{bmatrix} \xi_n^1 \\ \xi_n^2 \end{bmatrix}}_{\xi_n} \tag{5}$$

in which, again, $\mathbf{x}_0$, $\{\mathbf{u}_n\}_{n\geq 0}$ and $\{\xi_n\}_{n\geq 0}$ are independent. Then the AR noise block component can be concatenated to $\mathbf{x}_n$ in order to form an augmented state $\mathbf{x}_n^* = [\mathbf{x}_n^T, \mathbf{w}_n^1]^T$, and system (1) can be rewritten as :

$$\begin{cases} \mathbf{x}^*_{n+1} &= \begin{bmatrix} \mathbf{F}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^1_n \end{bmatrix} \mathbf{x}^*_n + \begin{bmatrix} \mathbf{G}_n \mathbf{u}_n \\ \xi^1_{n+1} \end{bmatrix} \\ \mathbf{y}_n &= [\mathbf{H}_n \ \mathbf{J}^1_n] \times \mathbf{x}^*_n + \mathbf{J}^2_n \xi^2_n \end{cases} \tag{6}$$

which again is an HMC-IN with singular measurement noise.

Remark 3: Remarks 1 and 2 still hold, and in particular (6) remains an HMC-IN with singular measurement noise if we only assume that $\mathbf{x}_0$ and $\{(\mathbf{u}_n, \xi_{n+1})\}_{n \geq 0}$ are independent.

In this paper we study the fixed-interval smoothing problem in model (1), in the case where $\mathbf{R}_n$ is an arbitrary (and possibly null) $\geq \mathbf{0}$ matrix. We exploit the singularity of $\mathbf{R}_n$ in order to transform the original state-space system into an (equivalent) stochastic dynamical system but with state dimension reduced by the nullity of $\mathbf{R}_n$. The transformed system happens to be a Pairwise Markov Chain (PMC) model, and as such the hidden process (even though it is not Markovian) is Markovian conditionnally on the observations. This key computational property finally enables us to develop fixed interval smoothing algorithms in the transformed system - and therefore, equivalently, in the original singular system.

The paper is organised as follows. Section II is devoted to the transformation of the HMC-IN model (1) into a reduced state PMC model. In section III we propose Bayesian fixed-interval smoothing algorithms in a general (i.e., not necessarily linear and Gaussian) PMC model. Following the Bayesian point of view for deriving minimum mean-square error (MMSE) algorithms for state-space systems (see e.g. [26]), in section IV we inject the Gaussian assumption into the algorithms of section III, which eventually yields our fixed interval smoothing algorithms for singular measurement noise state-space systems. In section V we perform some simulations, and finally ection VI concludes the paper.

II. State-space transform

We address the fixed-interval smoothing problem in the singular measurements case, i.e. in system (1) in the case where $\mathbf{R}_n \geq \mathbf{0}$. Some of the classical fixed-interval smoothing algorithms (see e.g. [27] for a recent review), originally designed for regular state-space systems, still hold in the singular case, but some others cannot be used any longer. In this paper we thus develop an alternative technique. Let r = rank$(\mathbf{R}_n) \in \{0, 1, \ldots, n_{\mathbf{y}} - 1\}$. Following [16], we transform the state space model (1) in order to reduce by $m = n_{\mathbf{y}} - r$ the order of state $\mathbf{x}_n$. In sections III and IV it will remain to design smoothing algorithms for this reduced-order linear stochastic system.

A. State-space transform

Since $\mathbf{R}_n$ has m zero eigenvalues, there exists a square non-singular matrix $\mathbf{M}_n$ satisfying

$$\mathbf{M}_n\mathbf{R}_n\mathbf{M}_n^T = \begin{bmatrix} \mathbf{0}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}, \tag{7}$$

in which $\mathbf{0}_m$ denotes the $m \times m$ null matrix. Let $\overline{\mathbf{y}}_n = \mathbf{M}_n\mathbf{y}_n$ and $\overline{\mathbf{v}}_n = \mathbf{M}_n\mathbf{v}_n$; then from (1) we get

$$\underbrace{\begin{bmatrix} \overline{\mathbf{y}}_n^p \\ \overline{\mathbf{y}}_n^r \end{bmatrix}}_{\overline{\mathbf{y}}_n} = \underbrace{\begin{bmatrix} \overline{\mathbf{H}}_n^p \\ \overline{\mathbf{H}}_n^r \end{bmatrix}}_{\overline{\mathbf{H}}_n = \mathbf{M}_n\mathbf{H}_n} \mathbf{x}_n + \underbrace{\begin{bmatrix} \mathbf{0}_{m\times 1} \\ \overline{\mathbf{v}}_n^r \end{bmatrix}}_{\overline{\mathbf{v}}_n}, \tag{8}$$

and we see that $\overline{\mathbf{y}}_n$ is divided into a perfect part $(\overline{\mathbf{y}}_n^p)_{m\times 1}$ (the unnoisy part) and a regular one $(\overline{\mathbf{y}}_n^r)_{r\times 1}$. Since m linear functionals of the state $\mathbf{x}_n$ are known once $\overline{\mathbf{y}}_n$ is known, there is no need to estimate them, and this is why one can reduce by m the order of the state-space system, as we now see. Let us assume that

$$n_{\mathbf{x}} \geq m, \tag{9}$$

and that in (8)

$$\text{rank}(\overline{\mathbf{H}}_n^p)_{m\times n_{\mathbf{x}}} = m. \tag{10}$$

Then one can choose a $(n_{\mathbf{x}} - m) \times n_{\mathbf{x}}$ matrix $\mathbf{U}_n$ in such a way that the transform

$$\underbrace{\begin{bmatrix} (\mathbf{U}_n)_{(n_{\mathbf{x}}-m)\times n_{\mathbf{x}}} \\ (\overline{\mathbf{H}}_n^p)_{m\times n_{\mathbf{x}}} \end{bmatrix}}_{\mathbf{T}_n} \mathbf{x}_n = \begin{bmatrix} \overline{\mathbf{x}}_n \\ \overline{\mathbf{y}}_n^p \end{bmatrix} \tag{11}$$

is reversible, and finally $\mathbf{T}_n$ and $\mathbf{M}_n$ enable us to transform the original linear state-space model (1) into the equivalent state-space system

$$\underbrace{\begin{bmatrix} \overline{\mathbf{x}}_{n+1} \\ \overline{\mathbf{y}}_{n+1}^p \end{bmatrix}}_{\mathbf{T}_{n+1}\mathbf{x}_{n+1}} = \mathbf{T}_{n+1}\mathbf{F}_n\mathbf{T}_n^{-1} \underbrace{\begin{bmatrix} \overline{\mathbf{x}}_n \\ \overline{\mathbf{y}}_n^p \end{bmatrix}}_{\mathbf{T}_n\mathbf{x}_n} + \mathbf{T}_{n+1}\mathbf{G}_n\mathbf{u}_n. \tag{12}$$

On the other hand from (8) and the first equation of (1) we have

$$\begin{aligned} \overline{\mathbf{y}}_{n+1}^r &= \overline{\mathbf{H}}_{n+1}^r\mathbf{x}_{n+1} + \overline{\mathbf{v}}_{n+1}^r \\ &= \overline{\mathbf{H}}_{n+1}^r\mathbf{F}_n\mathbf{T}_n^{-1} \begin{bmatrix} \overline{\mathbf{x}}_n \\ \overline{\mathbf{y}}_n^p \end{bmatrix} + \overline{\mathbf{H}}_{n+1}^r\mathbf{G}_n\mathbf{u}_n + \overline{\mathbf{v}}_{n+1}^r. \end{aligned} \tag{13}$$

Gathering (12) and (13), we eventually get the reduced-order linear dynamic stochastic system :

$$\underbrace{\begin{bmatrix}\overline{\mathbf{x}}_{n+1}\\ \overline{\mathbf{y}}_{n+1}\end{bmatrix}}_{\mathbf{z}_{n+1}}=\underbrace{\begin{bmatrix}\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} & \overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{y}}}\\ \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}} & \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{y}}}\end{bmatrix}}_{\overline{\mathcal{F}}_n}\begin{bmatrix}\overline{\mathbf{x}}_n\\ \overline{\mathbf{y}}_n\end{bmatrix}+\underbrace{\begin{bmatrix}\mathbf{U}_{n+1} & \mathbf{0}\\ \overline{\mathbf{H}}_{n+1} & \widetilde{\mathbf{I}}_r\end{bmatrix}\begin{bmatrix}\mathbf{G}_n\mathbf{u}_n\\ \overline{\mathbf{v}}_{n+1}^r\end{bmatrix}}_{\overline{\mathbf{w}}_n}, \tag{14}$$

with

$$\overline{\mathcal{F}}_n = \underbrace{\begin{bmatrix}\mathbf{T}_{n+1}\\ \overline{\mathbf{H}}_{n+1}^r\end{bmatrix}}_{(n_{\mathbf{x}}+r)\times n_{\mathbf{x}}}\underbrace{[\mathbf{F}_n\mathbf{T}_n^{-1} \ , \ \mathbf{0}]}_{n_{\mathbf{x}}\times(n_{\mathbf{x}}+r)}, \tag{15}$$

$$\widetilde{\mathbf{I}}_r = \begin{bmatrix}\mathbf{0}_{m\times r}\\ \mathbf{I}_r\end{bmatrix}. \tag{16}$$

Note that if $r=0$ then equations (13)-(16) are useless, and (14) reduces to (12). Now, (12) also coincides with (14) with $\overline{\mathbf{y}}_n=\overline{\mathbf{y}}_n^p=\mathbf{y}_n$ (since we can take $\mathbf{M}_n=\mathbf{I}_{n_{\mathbf{y}}}$), $\overline{\mathcal{F}}_n=\mathbf{T}_{n+1}\mathbf{F}_n\mathbf{T}_n^{-1}$, and $\overline{w}_n=\mathbf{T}_{n+1}\mathbf{G}_n\mathbf{u}_n$. As a consequence the algorithms of sections III and IV, designed for model (14), also hold (up to these adjustments) in the case $r=0$.

B. Markovianity of the reduced state model (14)

The noise of (14), $\{\overline{\mathbf{w}}_n\}_n$, is zero mean and independent since $\mathbf{u}$ and $\{\overline{\mathbf{v}}_n^r\}_n$ are independent and jointly independent. Moreover, from (15) and (11) we get

$$\overline{\mathcal{F}}_0\mathbf{z}_0=\begin{bmatrix}\mathbf{T}_1\\ \overline{\mathbf{H}}_1^r\end{bmatrix}\mathbf{F}_0\mathbf{x}_0.$$

Then $\overline{\mathbf{w}}$ is independent of $\overline{\mathcal{F}}_0\mathbf{z}_0$ (because $\overline{\mathbf{w}}$ is independent of $\mathbf{x}_0$). For these reasons, the process $\{\mathbf{z}_n\}_n$ is a Markov chain (MC), so model (14) defines a so-called PMC. PMC models have been first introduced in the discrete state case and have been applied in the context of image segmentation [28]. KF for continuous state PMC has also been addressed, see [29] [30], and parameter estimation via the EM algorithm is available too [31]. The term "pairwise" here emphasizes the fact that even though $\mathbf{z}_n=(\overline{\mathbf{x}}_n,\overline{\mathbf{y}}_n)$ is an MC, the marginal process $\overline{\mathbf{x}}$ in model (14) is indeed not Markovian. However, in a PMC the conditional distribution $p(\overline{\mathbf{x}}|\overline{\mathbf{y}})$ is Markovian; this property enables the developpement of efficient Bayesian restoration algorithms, and in particular, in the context of this paper, of fixed interval smoothing algorithms.

III. BAYESIAN FIXED-INTERVAL SMOOTHING ALGORITHMS IN PMC

For notational simplicity let us set $\overline{\mathbf{x}}_n = \mathbf{x}_n$, $\overline{\mathbf{y}}_n = \mathbf{y}_n$ and $\mathbf{z}_n = [\mathbf{x}_n^T, \mathbf{y}_n^T]^T$. Let us start by the general case, i.e. the case of the non-linear and/or non Gaussian PMC model :

$$\mathbf{z}_{n+1} = g(\mathbf{z}_n, \overline{\mathbf{w}}_n). \tag{17}$$

The aim of this section is to propose fixed-interval Bayesian smoothing algorithms for model (17), i.e. we want to compute the smoothing probability density function (pdf) $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ for all n, $0 \leq n \leq N$.

Recently most of the existing MMSE smoothing algorithms (as well as some new alternatives) for classical state-space systems (and therefore for HMC with continuous state) have been gathered and classified into a commun unifying framework [27]. That same classification has also been extended to the context of non-symmetrical triplet Markov chains (TMC) [32]. Let us adapt this classification to the context of model (17). All the algorithms described in propositions 2 to 4 combine one or two densities out of the set $\alpha_n \stackrel{\text{def}}{=} p(\mathbf{x}_n|\mathbf{y}_{0:n})$, $\beta_n \stackrel{\text{def}}{=} p(\mathbf{y}_{n+1:N}|\mathbf{z}_n)$, $\gamma_n \stackrel{\text{def}}{=} p(\mathbf{x}_n|\mathbf{y}_{n:N})$ and $\eta_n \stackrel{\text{def}}{=} p(\mathbf{y}_{0:n-1}|\mathbf{z}_n)$. So we begin with the following proposition :

Proposition 1: $\alpha_n = p(\mathbf{x}_n|\mathbf{y}_{0:n})$ and $\widetilde{\alpha}_n = p(\mathbf{x}_n|\mathbf{y}_{0:n+1})$ can be computed recursively (in the forward direction, i.e. for increasing values of n) as

$$\begin{cases} \widetilde{\alpha}_n & = \frac{p(\mathbf{y}_{n+1}|\mathbf{z}_n)\times\alpha_n}{p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n})=\int p(\mathbf{y}_{n+1}|\mathbf{z}_n)\alpha_n d\mathbf{x}_n} \\ \alpha_{n+1} & = \int p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) \times \widetilde{\alpha}_n d\mathbf{x}_n \end{cases} ; \tag{18}$$

$\beta_n = p(\mathbf{y}_{n+1:N}|\mathbf{z}_n)$ and $\widetilde{\beta}_n = p(\mathbf{y}_{n+2:N}|\mathbf{z}_n, \mathbf{y}_{n+1})$ can be computed recursively (in the backward direction, i.e. for decreasing values of n) as

$$\begin{cases} \widetilde{\beta}_n & = \int p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) \times \beta_{n+1}\, d\mathbf{x}_{n+1} \\ \beta_n & = p(\mathbf{y}_{n+1}|\mathbf{z}_n)\, \widetilde{\beta}_n \end{cases} ; \tag{19}$$

$\eta_n = p(\mathbf{y}_{0:n-1}|\mathbf{z}_n)$ and $\widetilde{\eta}_n = p(\mathbf{y}_{0:n-2}|\mathbf{z}_n, \mathbf{y}_{n-1})$ can be computed recursively (in the forward direction) as

$$\begin{cases} \widetilde{\eta}_{n+1} & = \int p(\mathbf{x}_n|\mathbf{z}_{n+1}, \mathbf{y}_n) \times \eta_n\, d\mathbf{x}_n \\ \eta_{n+1} & = p(\mathbf{y}_n|\mathbf{z}_{n+1}) \times \widetilde{\eta}_{n+1} \end{cases} ; \tag{20}$$

and $\gamma_n = p(\mathbf{x}_n|\mathbf{y}_{n:N})$ and $\widetilde{\gamma}_n = p(\mathbf{x}_n|\mathbf{y}_{n-1:N})$ can be computed recursively (in the backward direction) as

$$\begin{cases} \widetilde{\gamma}_{n+1} & = \frac{p(\mathbf{y}_n|\mathbf{z}_{n+1})\times\gamma_{n+1}}{p(\mathbf{y}_n|\mathbf{y}_{n+1:N})=\int p(\mathbf{y}_n|\mathbf{z}_{n+1})\gamma_{n+1} d\mathbf{x}_{n+1}} \\ \gamma_n & = \int p(\mathbf{x}_n|\mathbf{z}_{n+1}, \mathbf{y}_n) \times \widetilde{\gamma}_{n+1} d\mathbf{x}_{n+1} \end{cases} . \tag{21}$$

We now turn to the computation of the smoothing density $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ itself. We have the following propositions.

Proposition 2: $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ can be computed in the backward direction by

$$p(\mathbf{x}_n|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_{n+1}|\mathbf{y}_{0:N}) p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:N}) d\mathbf{x}_{n+1}, \tag{22}$$

with

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:N}) &\propto p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) \times \widetilde{\alpha}_n & (23) \\ &\propto p(\mathbf{x}_n|\mathbf{z}_{n+1}, \mathbf{y}_n) \times \eta_n & (24) \\ &\propto \frac{p(\mathbf{x}_n|\mathbf{z}_{n+1}, \mathbf{y}_n) \times \alpha_n}{p(\mathbf{x}_n|\mathbf{y}_n)} & (25) \\ &\propto p(\mathbf{z}_{n+1}|\mathbf{z}_n) \times \eta_n \times p(\mathbf{x}_n|\mathbf{y}_n). & (26) \end{aligned}$$

Proposition 3: $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ can be computed in the forward direction by

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_n|\mathbf{y}_{0:N}) p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{0:N}) d\mathbf{x}_n, \tag{27}$$

with

$$\begin{aligned} p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{0:N}) &\propto p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) \times \beta_{n+1} & (28) \\ &\propto p(\mathbf{x}_n|\mathbf{z}_{n+1}, \mathbf{y}_n) \times \widetilde{\gamma}_{n+1} & (29) \\ &\propto \frac{p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) \times \gamma_{n+1}}{p(\mathbf{x}_{n+1}|\mathbf{y}_{n+1})} & (30) \\ &\propto p(\mathbf{z}_n|\mathbf{z}_{n+1}) \times \beta_{n+1} \times p(\mathbf{x}_{n+1}|\mathbf{y}_{n+1}). & (31) \end{aligned}$$

Proposition 4: The smoothing pdf $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ can be computed as

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:N}) &\propto \alpha_n \times \beta_n & (32) \\ &\propto \gamma_n \times \eta_n & (33) \\ &\propto \frac{\alpha_n \times \gamma_n}{p(\mathbf{x}_n|\mathbf{y}_n)} & (34) \\ &\propto \eta_n \times \beta_n \times p(\mathbf{x}_n|\mathbf{y}_n). & (35) \end{aligned}$$

IV. MMSE FIXED-INTERVAL SMOOTHING ALGORITHMS.

Let us now come back to the linear PMC (14). The general algorithms of Propositions 2 to 4 reduce to MMSE fixed-interval smoothing algorithms if we further inject the Gaussian assumption. So from now on we also assume that $\mathbf{z}_0$ and $\overline{\mathbf{w}}_n$ are Gaussian for all n, which in turn holds if the original state-space model (1) is Gaussian, i.e. if $\mathbf{x}_0$, $\mathbf{u}_n$ and $\mathbf{v}_n$ are Gaussian for all n. Let us set

$$\mathbf{x}_0 \sim \mathcal{N}(\widehat{\mathbf{x}}_0, \mathbf{P}_0), \overline{\mathbf{w}}_n \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \overline{\mathbf{Q}}_n & \overline{\mathbf{S}}_n \\ \overline{\mathbf{S}}_n^T & \overline{\mathbf{R}}_n \end{bmatrix}). \tag{36}$$

In this case, all the pdfs in §III are Gaussian. Let us set

$$p(\mathbf{x}_n|\mathbf{y}_{i:j}) \sim \mathcal{N}(\widehat{\mathbf{x}}_{n|i:j}, \mathbf{P}_{n|i:j}), \tag{37}$$

for all n, i, j with $0 \leq n \leq N$ and $0 \leq i \leq j \leq N$.

The general algorithms of Propositions 2 to 4 compute $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ from α_n (or η_n) and/or γ_n (or β_n). In the Gaussian case, this amounts to computing the parameters of $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ from those of α_n (or η_n) and/or γ_n (or β_n). More precisely, (18) to (35) reduce to equations which compute $\arg\max_{\mathbf{x}_n} p(\mathbf{x}_n|\mathbf{y}_{0:N})$ (i.e., $\widehat{\mathbf{x}}_{n|0:N}$), and the associated covariance matrix, from $\arg\max_{\mathbf{x}_n} \alpha_n = \widehat{\mathbf{x}}_{n|0:n}$ (or $\arg\max_{\mathbf{x}_n} \eta_n$) and/or $\arg\max_{\mathbf{x}_n} \gamma_n = \widehat{\mathbf{x}}_{n|n:N}$ (or $\arg\max_{\mathbf{x}_n} \beta_n$), as well as the associated covariance matrice(s). In practice, these equations can be derived by systematically applying some simple results for Gaussian variables; each one of the twelve general algorithms in Propositions 2, 3 and 4 then reduces to a particular Kalman smoothing algorithm; some of them are PMC extensions of such classical Kalman-like smoothing algorithms as, e.g., the RTS algorithm [3], the two-filter algorithm [6] [7], or the general two-filter algorithm [11, Theorem 10.4.1] (see [27] for details).

A. *A worked example : the PMC RTS algorithm*

This section is devoted to the implementation of $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ (equations (23)-(22) in case of a Gaussian PMC model (i.e., we assume that (14) and (36) hold).

First, the algorithm requires the propagation of $\widetilde{\alpha}_n$ in the forward direction. In the Gaussian case, the propagation of the Gaussian densities α_n and $\widetilde{\alpha}_n$ via algorithm (18) reduces to the PMC KF algorithm [33, eqs. (13.56) and (13.57)] which propagates their parameters ($\widehat{\mathbf{x}}_{n|0:n}$, $\mathbf{P}_{n|0:n}$) and ($\widehat{\mathbf{x}}_{n|0:n+1}$, $\mathbf{P}_{n|0:n+1}$). For convenience of the reader, the PMC KF recalled in the following proposition.

Proposition 5: PMC-KF algorithm. Let (14) and (36) hold. Then $\widehat{\mathbf{x}}_{n+1|0:n+1}$ and $\mathbf{P}_{n+1|0:n+1}$ can be computed from $\widehat{\mathbf{x}}_{n|0:n}$ and $\mathbf{P}_{n|0:n}$ via:

$$\begin{aligned}
\widetilde{\mathbf{y}}_{n+1|0:n} &= \mathbf{y}_{n+1} - \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}} \widehat{\mathbf{x}}_{n|0:n} - \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{y}}} \mathbf{y}_n, && (38)\\
\mathbf{P}^{\mathbf{y}}_{n+1|0:n} &= \overline{\mathbf{R}}_n + \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}} \mathbf{P}_{n|0:n} (\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}})^T, && (39)\\
\mathbf{K}_{n|0:n+1} &= \mathbf{P}_{n|0:n} (\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}})^T (\mathbf{P}^{\mathbf{y}}_{n+1|0:n})^{-1}, && (40)\\
\widehat{\mathbf{x}}_{n|0:n+1} &= \widehat{\mathbf{x}}_{n|0:n} + \mathbf{K}_{n|0:n+1} \widetilde{\mathbf{y}}_{n+1|0:n}, && (41)\\
\mathbf{P}_{n|0:n+1} &= \mathbf{P}_{n|0:n} - \mathbf{K}_{n|0:n+1} \mathbf{P}^{\mathbf{y}}_{n+1|0:n} \mathbf{K}^T_{n|0:n+1}, && (42)\\
\widehat{\mathbf{x}}_{n+1|0:n+1} &= [\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} - \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}] \widehat{\mathbf{x}}_{n|0:n+1} + \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \mathbf{y}_{n+1} + [\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{y}}} - \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{y}}}] \mathbf{y}_n, && (43)\\
\mathbf{P}_{n+1|0:n+1} &= [\overline{\mathbf{Q}}_n - \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \overline{\mathbf{S}}_n^T] + [\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} - \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}] \mathbf{P}_{n|0:n+1} [\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} - \overline{\mathbf{S}}_n \overline{\mathbf{R}}_n^{-1} \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}]^T && (44)
\end{aligned}$$

It remains to compute the parameters $\widehat{\mathbf{x}}_{n|0:N}$ and $\mathbf{P}_{n|0:N}$ of $p(\mathbf{x}_n|\mathbf{y}_{0:N})$. In the Gaussian case, (23)-(22) reduce to the following equations (see §VII-B for a proof) :

Proposition 6: PMC-RTS algorithm. Let (14) and (36) hold. Let

$$\mathbf{K}_{n|0:N} = \mathbf{P}_{n|0:n+1}[\overline{\mathcal{F}}_n^{\mathbf{x},\mathbf{x}} - \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{x}}]^T \mathbf{P}_{n+1|0:n+1}^{-1}. \tag{45}$$

Then

$$\widehat{\mathbf{x}}_{n|0:N} = \widehat{\mathbf{x}}_{n|0:n+1} + \mathbf{K}_{n|0:N}[\widehat{\mathbf{x}}_{n+1|0:N} - \widehat{\mathbf{x}}_{n+1|0:n+1}], \tag{46}$$

$$\mathbf{P}_{n|0:N} = \mathbf{P}_{n|0:n+1} - \mathbf{K}_{n|0:N}[\mathbf{P}_{n+1|0:n+1} - \mathbf{P}_{n+1|0:N}]\mathbf{K}_{n|0:N}^T. \tag{47}$$

In summary, $(\widehat{\mathbf{x}}_{n|0:N}, \mathbf{P}_{n|0:N})$ can be propagated in the backward direction via (45)-(47) provided $(\widehat{\mathbf{x}}_{n|0:n+1}, \mathbf{P}_{n|0:n+1})$ and $(\widehat{\mathbf{x}}_{n+1|0:n+1}, \mathbf{P}_{n+1|0:n+1})$ are known; these, in turn, can be computed recursively (in the forward direction) by the PMC KF algorithm recalled in Proposition 5.

B. An alternate algorithme : the PMC Bryson and Frazier (BF) algorithm

Let us finally mention the BF algorithm [1], which cannot actually be derived from Bayesian considerations, even though it is closely related to the RTS algorithm. The BF algorithm was introduced for the first time (in the continuous time case) as a solution of a deterministic least mean square problem by using the variational approach [1] (see also [11, chap. 10 & 6] and [19, pp. 223-25]). The discrete time version of this algorithm then appeared in various publications like e.g. [34] [3] [18] [11]. The link between the RTS and the BF algorithms has been established for the first time in [3]. As in the classical HMC framework, the PMC RTS algorithm can also be implemented by an algorithm which extends to PMC the BF algorithm (see §VII-C for a proof) :

Proposition 7: Let (14) and (36) hold. Let

$$\begin{aligned}
\widehat{\mathbf{y}}_{n+1|0:n} &= \overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{x}}\widehat{\mathbf{x}}_{n|0:n} + \overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{y}}\mathbf{y}_n \\
\mathbf{P}_{n+1|0:n}^{\mathbf{y}} &= \overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{x}}\mathbf{P}_{n|0:n}(\overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{x}})^T + \overline{\mathbf{R}}_n \\
\mathbf{K}_n^{\lambda} &= \overline{\mathcal{F}}_n^{\mathbf{x},\mathbf{x}}[\mathbf{I} - \mathbf{P}_{n|0:n}(\overline{\mathcal{F}}_n^{\mathbf{y},\mathbf{x}})^T(\mathbf{P}_{n+1|0:n}^{\mathbf{y}})^{-1}\overline{\mathcal{F}}_n^{\mathbf{x},\mathbf{x}}].
\end{aligned}$$

Then ($\widehat{\mathbf{x}}_{n|0:N}$ and $\mathbf{P}_{n|0:N}$) can be computed as

$$\widehat{\mathbf{x}}_{n|0:N} = \widehat{\mathbf{x}}_{n|0:n+1} + \mathbf{P}_{n|0:n}(\mathbf{K}_n^{\lambda})^T\lambda_{n+1}, \tag{48}$$

$$\mathbf{P}_{n|0:N} = \mathbf{P}_{n|0:n+1} - \mathbf{P}_{n|0:n}(\mathbf{K}_n^{\lambda})^T\Lambda_{n+1}\mathbf{K}_n^{\lambda}\mathbf{P}_{n|0:n}, \tag{49}$$

in which the parameters

$$\lambda_n = \mathbf{P}_{n|0:n}^{-1}[\widehat{\mathbf{x}}_{n|0:N} - \widehat{\mathbf{x}}_{n|0:n}] \quad (50)$$

$$\Lambda_n = \mathbf{P}_{n|0:n}^{-1}[\mathbf{P}_{n|0:n} - \mathbf{P}_{n|0:N}]\mathbf{P}_{n|0:n}^{-1}, \quad (51)$$

are computed recursively (with $\lambda_N = \mathbf{0}$ and $\Lambda_N = \mathbf{0}$) as follows :

$$\lambda_n = (\mathbf{K}_n^{\lambda})^T \lambda_{n+1} + (\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}})^T (\mathbf{P}_{n+1|0:n}^{\mathbf{y}})^{-1}[\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|0:n}], \quad (52)$$

$$\Lambda_n = (\mathbf{K}_n^{\lambda})^T \Lambda_{n+1}\mathbf{K}_n^{\lambda} + (\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}})^T (\mathbf{P}_{n+1|0:n}^{\mathbf{y}})^{-1}\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}. \quad (53)$$

As we can see, the PMC BF algorithm is still a two pass one. In the forward pass, the filtering and one-step smoothing parameters are computed by the PMC KF. One then propagates in the backward pass the new variables λ_n and Λ_n via (52) and (53). The smoothing parameters of interest are finally computed as a combination of the forward and the backward quantities, see (48)-(49).

V. Numerical simulations

Let us now perform some simulations. We consider the following model :

$$\mathbf{x}_{n+1} = \mathbf{F}_n\mathbf{x}_n + \mathbf{u}_n \quad (54)$$

$$\mathbf{y}_n = \underbrace{\begin{bmatrix} -1.9 & 1 & 1.2 \\ 5 & 1 & -90 \\ 3 & 1.7 & 1 \end{bmatrix}}_{\mathbf{H}_n} \mathbf{x}_n + \underbrace{\begin{bmatrix} 0.7071 \\ 0 \\ 0.7071 \end{bmatrix}}_{\mathbf{J}_n} w_n \quad (55)$$

with $F_n = 0.98\mathbf{I}_3$, and $\mathbf{u}_n$ and v_n are independent, jointly independent and independent of $\mathbf{x}_0$. We assume that $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_n)$, $w_n \sim \mathcal{N}(0,1)$ and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}_3, 0.01 \times \mathbf{I}_3)$. So the covariance $\mathbf{R}_n$ of the measurement noise $\mathbf{v}_n = \mathbf{J}_n \times w_n$ is a rank 1 matrix :

$$\mathbf{J}_n v_n \sim \mathcal{N}(\mathbf{0}, \underbrace{\begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}}_{\mathbf{R}_n}).$$

A FAIRE *Following* §*II, we set* $\mathbf{M} = xxx$, *and next* $\mathbf{T}_n = xxx$. *So model* (54) *is transformed to* $\cdots$ *(écrire l'équivalent de (14)-(16), en précisant les matrices, la dimension de* $\overline{\mathbf{x}}$, *et comment on revient de* $(\hat{\overline{\mathbf{x}}}, \mathbf{y})$ *à* $\hat{\mathbf{x}}$. *Je pense que ça peut intéresser les gens.)*

We now compare the PMC-KF and the PMC-RTS algorithms in the reduced state-order model.

A. *Comparison between the PMC-KF and the PMC-RTS*

A ETTOFFER - LIS MON MAIL

In this section we consider the case of $\mathbf{Q}_n = 0.01 \times \widetilde{\mathbf{Q}}_n$, and we focus on the performance of the PMC-RTS compared with the PMC-KF, after the reduced state transformation. The figure 1 illustrates the tracking of the first component of $\mathbf{x}_n$ (i.e., x_n^1), and in the figure 2 we plot the MSE associated to x_n^1. All the results are averaged over 100 realizations.

As we can see, the PMC-RTS smoothing algorithm performs the PMC-KF filtering one. This coincides with the truth since the smoothing estimator takes in account all the measurements $\overline{\mathbf{y}}_{0:N}$ in each time n, while the filtering estimator uses only the present and the past measurements $\overline{\mathbf{y}}_{0:n}$. Finally, remark that the results given by these 2 algorithms coincide in the final time $n = N = 50$; this can be explained by the fact that the smoothing becomes the filtering at $n = N$.

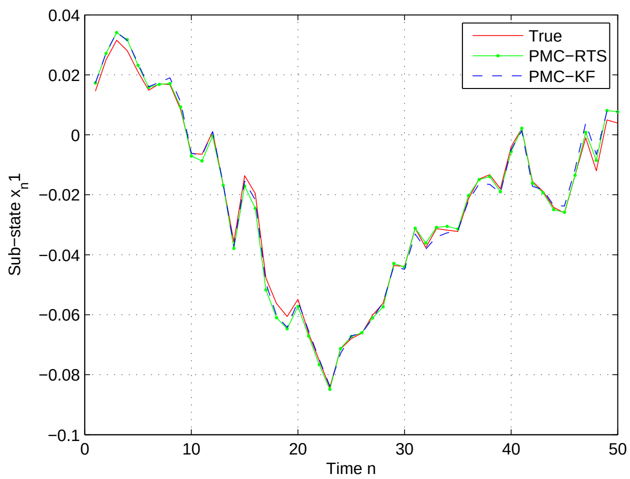

Fig. 1. Tracking of the first component of $\mathbf{x}_n$.

VI. Conclusion

In this paper we addressed the fixed-interval smoothing problem in state-space systems with singular measurement noise, i.e. in the case where the covariance matrix of the measurement noise is either null or $\geq \mathbf{0}$ with arbitrary rank. This case is of interest in a number of situations, including the case where some

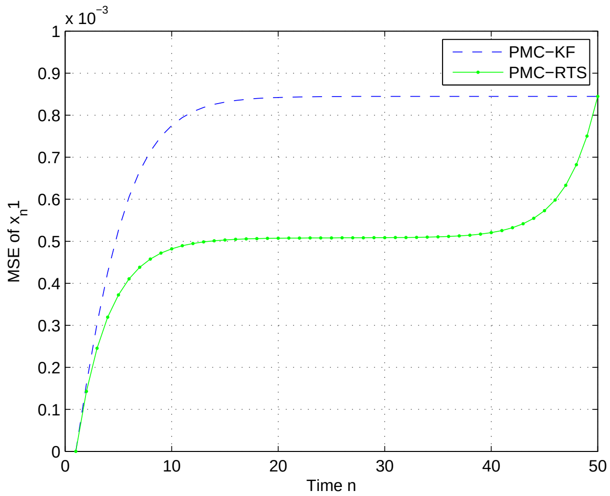

Fig. 2. The associated MSE of the tracking of the first component of $\mathbf{x}_n$.

of the sensors are affected by colored noise. We first transformed the original (singular) HMC model into an equivalent PMC model with regular noise and state dimension reduced by the nullity of that covariance matrix. Though the transformed system is no longer an HMC (in particular the hidden state is no longer Markovian), it enables Bayesian restoration because the state remains Markovian conditionally on the observations. We finally proposed a set of Bayesian fixed-interval smoothing algorithms, which reduce in the Gaussian case to Kalman-like smoothing algorithms for singular systems.

VII. Annex

A. *Some useful results for Gaussian variables.*

The proof of of Proposition 6 is based on the following two classical results on Gaussian variables.

Proposition 8: Let $p(\mathbf{x}) \sim \mathcal{N}(\widehat{\mathbf{x}}, \mathbf{P}_x)$ and $p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{P}_{y|x})$. Then

$$p(\mathbf{x}, \mathbf{y}) \sim \mathcal{N}(\begin{bmatrix} \widehat{\mathbf{x}} \\ \mathbf{A}\widehat{\mathbf{x}} + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_x & \mathbf{P}_x\mathbf{A}^T \\ \mathbf{A}\mathbf{P}_x & \mathbf{A}\mathbf{P}_x\mathbf{A}^T + \mathbf{P}_{y|x} \end{bmatrix}).$$

Proposition 9: Let $p(\mathbf{x}, \mathbf{y}) \sim \mathcal{N}(\begin{bmatrix} \widehat{\mathbf{x}} \\ \widehat{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_x & \mathbf{P}_{x,y} \\ \mathbf{P}_{y,x} & \mathbf{P}_y \end{bmatrix})$. Then

$$p(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\widehat{\mathbf{x}} + \mathbf{P}_{x,y}\mathbf{P}_y^{-1}(\mathbf{y} - \widehat{\mathbf{y}}), \mathbf{P}_x - \mathbf{P}_{x,y}\mathbf{P}_y^{-1}\mathbf{P}_{y,x}).$$

B. Proof of Proposition 6.

Let us address the calculation of (23)-(22). First, from (14) we have

$$p(\underbrace{(\mathbf{x}_{n+1}, \mathbf{y}_{n+1})}_{\mathbf{z}_{n+1}}|\mathbf{z}_n) \sim \mathcal{N}(\begin{bmatrix} \overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} & \overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{y}}} \\ \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}} & \overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{y}}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{x}}_n \\ \overline{\mathbf{y}}_n \end{bmatrix}, \begin{bmatrix} \overline{\mathbf{Q}}_n & \overline{\mathbf{S}}_n \\ (\overline{\mathbf{S}}_n)^T & \overline{\mathbf{R}}_n \end{bmatrix}), \tag{56}$$

and by using proposition 9 we get

$$\begin{aligned} p(\mathbf{x}_{n+1}|\mathbf{z}_n, \mathbf{y}_{n+1}) &= p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{0:n+1}) \\ &\sim \mathcal{N}(\underbrace{[\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} - \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}]}_{\mathbf{A}_n} \mathbf{x}_n + \underbrace{[\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{y}}} - \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{y}}}]}_{\mathbf{B}_n} \mathbf{y}_n \\ &\quad + \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\mathbf{y}_{n+1}, \underbrace{[\overline{\mathbf{Q}}_n - \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-}1(\overline{\mathbf{S}}_n)^T]}_{\mathbf{C}_n}). \end{aligned} \tag{57}$$

(the first equality holds because $(\mathbf{x}_n, \mathbf{y}_n)$ is an MC). On the other hand,

$$\widetilde{\alpha}_n = p(\mathbf{x}_n|\mathbf{y}_{0:n+1}) \sim \mathcal{N}(\widehat{\mathbf{x}}_{n|0:n+1}, \mathbf{P}_{n|0:n+1}). \tag{58}$$

Applying Proposition 8 to (58) and (57) we get

$$p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{0:n+1}) \sim \mathcal{N}(\begin{bmatrix} \widehat{\mathbf{x}}_{n|0:n+1} \\ \mathbf{A}_n\widehat{\mathbf{x}}_{n|0:n+1} + \mathbf{B}_n\mathbf{y}_n + \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\mathbf{y}_{n+1} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{n|0:n+1} & \mathbf{P}_{n|0:n+1}\mathbf{A}_n^T \\ \mathbf{A}_n\mathbf{P}_{n|0:n+1} & \mathbf{A}_n\mathbf{P}_{n|0:n+1}\mathbf{A}_n^T + \mathbf{C}_n \end{bmatrix}). \tag{59}$$

Applying Proposition 9 in (59), and observing that conditionnally on $(\mathbf{x}_{n+1}, \mathbf{y}_{0:n+1})$, $\mathbf{x}_n$ and $\mathbf{y}_{n+2:N}$ are independent, we get

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:N}) &\sim (\widehat{\mathbf{x}}_{n|0:n+1} + \underbrace{\mathbf{P}_{n|0:n+1}\mathbf{A}_n^T\mathbf{P}_{n+1|0:n+1}^{-1}}_{\mathbf{K}_{n|0:N}}(\mathbf{x}_{n+1} - \widehat{\mathbf{x}}_{n+1|0:n+1}), \\ &\quad \mathbf{P}_{n|0:n+1} - \mathbf{P}_{n|0:n+1}\mathbf{A}_n^T\mathbf{P}_{n+1|0:n+1}^{-1}\mathbf{A}_n\mathbf{P}_{n|0:n+1}). \end{aligned} \tag{60}$$

from which we get formula (45) for the Kalman smoothing gain $\mathbf{K}_{n|0:N}$. Injecting (60) in (22) and using Proposition 8 again we eventually get (46) and (47).

C. Proof of proposition 7.

Injecting (45) in (46) and (47) leads respectively to:

$$\widehat{\mathbf{x}}_{n|0:N} = \widehat{\mathbf{x}}_{n|0:n+1} + \mathbf{P}_{n|0:n+1}\underbrace{[\overline{\mathcal{F}}_n^{\overline{\mathbf{x}},\overline{\mathbf{x}}} - \overline{\mathbf{S}}_n(\overline{\mathbf{R}}_n)^{-1}\overline{\mathcal{F}}_n^{\overline{\mathbf{y}},\overline{\mathbf{x}}}]}_{\mathbf{A}_n}{}^T\lambda_{n+1}, \tag{61}$$

$$\mathbf{P}_{n|0:N} = \mathbf{P}_{n|0:n+1} - \mathbf{P}_{n|0:n+1}\mathbf{A}_n^T\Lambda_{n+1}\mathbf{A}_n\mathbf{P}_{n|0:n+1}. \tag{62}$$

On the other hand, from (42) and (40) we get

$$\mathbf{P}_{n|0:n+1}\mathbf{A}_n^T = \mathbf{P}_{n|0:n}(\mathbf{K}_n^\lambda)^T. \tag{63}$$

Injecting (63) in (61) and (62) leads respectively to (48) and (49).

It remains to show (52) and (53). Injection (48) in (50) and (49) in (51) leads respectively to:

$$\lambda_n = (\mathbf{K}_n^\lambda)^T \lambda_{n+1} + \mathbf{P}_{n|0:n}^{-1}[\widehat{\mathbf{x}}_{n|0:n+1} - \widehat{\mathbf{x}}_{n|0:n}] \tag{64}$$

$$\Lambda_n = (\mathbf{K}_n^\lambda)^T \Lambda_{n+1}\mathbf{K}_n^\lambda + \mathbf{P}_{n|0:n}^{-1}[\mathbf{P}_{n|0:n} - \mathbf{P}_{n|0:n+1}]\mathbf{P}_{n|0:n}^{-1}. \tag{65}$$

Finally injecting (41) and (40) in (64) we get (52), and injecting (42) and (40) in (65) we get (53).

REFERENCES

[1] A.E. Bryson and M. Frazier, "Smoothing for linear and non-linear dynamic systems," *Tech. Rep., Aero. Syst. Dvi. Wrigth-Patterson Air Force Base*, pp. 353–364, 1963.

[2] H. E. Rauch, "Solutions to smoothing problem," *IEEE Transactions on Automatic Control*, vol. 8, pp. 371–372, October 1963.

[3] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[4] J. S. Meditch, "Orthogonal projection and discrete optimal linear smoothing," *SIAM J. Contr*, vol. 5, pp. 74–89, January 1967.

[5] T. Kailath and P. Frost, "An innovations approach to least squares estimation, part II : Linear smoothing in additive white noise," *IEEE Transactions on Automatic Control*, vol. 13, pp. 655–660, December 1968.

[6] D. Q. Mayne, "A solution to the smoothing problem for linear dynamical systems," *Automatica*, vol. 4, pp. 73–92, 1966.

[7] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 7, pp. 387–90, Aug. 1969.

[8] H. L. Weinert, *Fixed interval smoothing for state-space models*, The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, 2001.

[9] R. C. K. Lee, *Optimal Estimation: Identification and Control*, M.I.T. Press, Cambridge, MA, 1964.

[10] M. Askar and H. Derin, "A recursive algorithm for the Bayes solution of the smoothing problem," *IEEE Transactions on Automatic Control*, vol. 26, pp. 558–560, 1981.

[11] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*, Prentice Hall Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, 2000.

[12] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag, 2005.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng., Trans. ASME, Series D*, vol. 82, no. 1, pp. 35–45, 1960.

[14] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, 1989.

[15] R. Mehra and A. E. Jr. Bryson, "Linear smoothing using measurements containing correlated noise with an application to inertial navigation," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 496–503, October 1968.

[16] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice Hall, Englewood Cliffs, New Jersey, 1979.

[17] P. S. Maybeck, *Stochastic Models, Estimation and Control*, vol. 1, New York: Academic Press, 1979.

[18] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, Hemisphere, New York, 1975.

[19] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, vol. 64 of *Mathematics in Science and Engineering*, Academic Press, San Diego, 1970.

[20] A. E. Bryson Jr. and L. J. Henrikson, "Estimation using sampled data containing sequentially correlated noise," *Journal of Spacecraft and Rockets*, vol. 5, pp. 662–65, June 1968.

[21] G. Doblinger, "Smoothing of AR signals using an adaptive Kalman filter," in *Proceedings of the European Signal Processing Conference (EUSIPCO 98)*, 1998, vol. 2, pp. 781–784.

[22] V. Shukla, "Estimation in linear time-delay system with coloured observation noise," *IEEE Electronics Letters*, vol. 9, no. 1, pp. 1–2, January 1973.

[23] K. Biswas and A. Mahalanabis, "Optimal fixed lag smoothing for time delayed system with colored noise," *IEEE Transactions on Automatic Control*, vol. 17, no. 3, pp. 387– 88, June 1972.

[24] G. T. Spyros, "On optimum distributed-parameter filtering and fixed-interval smoothing for colored noise," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 448– 58, August 1972.

[25] C.K. Chui and G. Chen, *Kalman filtering with real-time applications*, Berlin, DE: Springer, 1999.

[26] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Transactions on Automatic Control*, vol. 9, pp. 333–339, October 1964.

[27] B. Ait-El-Fquih and F. Desbouvries, "On Bayesian Fixed-Interval Smoothing Algorithms," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2437–42, November 2008.

[28] W. Pieczynski, "Pairwise Markov chains," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 634–39, May 2003.

[29] W. Pieczynski and F. Desbouvries, "Kalman filtering using pairwise Gaussian models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 03)*, Hong-Kong, 2003.

[30] F. Desbouvries and W. Pieczynski, "Particle filtering in pairwise and triplet Markov chains," in *Proc. IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, Grado-Gorizia, Italy, June 8-11 2003.

[31] B. Ait-El-Fquih and F. Desbouvries, "Unsupervised signal restoration in partially observed Markov chains," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 06)*, Toulouse, France, May 15-19 2006.

[32] B. Ait el Fquih and F. Desbouvries, "Exact and approximate Bayesian smoothing algorithms in partially observed Markov chains," in *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW'06)*, Cambridge, UK, September 13-15 2006.

[33] R. S. Lipster and A. N. Shiryaev, *Statistics of Random Processes, Vol. 2 : Applications*, chapter 13 : "Conditionally Gaussian Sequences : Filtering and Related Problems", Springer Verlag, Berlin, 2001.

[34] H. Cox, "On the Estimation of State Variables and Parameters for Noisy Dynamic Systems," *IEEE Transactions on Automatic Control*, vol. 9, no. 1, pp. 5–12, January 1964.