Robust partial-learning in linear Gaussian systems

Valérian Némesin, and Stéphane Derrode

***Abstract*—This paper deals with unsupervised and off-line learning of parameters involved in linear Gaussian systems, *i.e.* the estimation of the transition and the noise covariances matrices of a state-space system from a finite series of observations only. In practice, these systems are the result of a physical problem for which there is a partial knowledge either on the sensors from which the observations are issued or on the state of the studied system. We therefore propose in this work an "Expectation-Maximization" learning type algorithm that takes into account constraints on parameters such as the fact that two identical sensors have the same noise characteristics, and so estimation procedure should exploit this knowledge. The algorithms are designed for the pairwise linear Gaussian system that takes into account supplementary cross-dependences between observations and hidden states *w.r.t.* the conventional linear system, while still allowing optimal filtering by means of a Kalman-like filter. The algorithm is made robust through QR decompositions and the propagation of a square-root of the covariance matrices instead of the matrices themselves. It is assessed through a series of experiments that compare the algorithms which incorporate or not partial knowledge, for short as well as for long signals.**

***Index Terms*—Dynamic linear model, Kalman filter, Pairwise Kalman filter, Expectation-Maximization, Parameter learning.**

I. Introduction

An important problem in signal processing consists in estimating a set of hidden variables $\boldsymbol{x} = \{\boldsymbol{x}_n\}_{n=[0:N]}$ from a set of observations $\boldsymbol{y} = \{\boldsymbol{y}_n\}_{n=[0:N]}$. In this work, we are interested in the model called pairwise linear Gaussian system (PLGS)

$$\underbrace{\begin{pmatrix} \boldsymbol{x}_{n+1} \\ \boldsymbol{y}_n \end{pmatrix}}_{\boldsymbol{t}_{n+1}} = \underbrace{\begin{pmatrix} \boldsymbol{F}^{x,x} & \boldsymbol{F}^{x,y} \\ \boldsymbol{F}^{y,x} & \boldsymbol{F}^{y,y} \end{pmatrix}}_{\boldsymbol{F}} \underbrace{\begin{pmatrix} \boldsymbol{x}_n \\ \boldsymbol{y}_{n-1} \end{pmatrix}}_{\boldsymbol{t}_n} + \underbrace{\begin{pmatrix} \boldsymbol{\omega}^x_{n+1} \\ \boldsymbol{\omega}^y_{n+1} \end{pmatrix}}_{\boldsymbol{\omega}_{n+1}}, \tag{1}$$

introduced in [1] and extended to triplet Markov chain models in [2], where

- $\boldsymbol{x}_n \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{y}_n \in \mathbb{R}^{n_{\boldsymbol{y}}}$ denote the states and observations respectively ($n_t = n_x + n_y$);
- the transition matrix $\boldsymbol{F}$ and the noise covariance matrix $\boldsymbol{Q}$, supposed independent of n in this work, define the parameters of the model;
- $\boldsymbol{\omega} = \{\boldsymbol{\omega}_n\}_{n=[0:N]}$ is a Gaussian process, where $\boldsymbol{\omega}_n \in \mathbb{R}^{n_{\boldsymbol{t}}}$ are mutually independent and independent of $\boldsymbol{t}_0 \sim \mathcal{N}\left(\hat{\boldsymbol{t}}_0, \boldsymbol{Q}_0\right)$ and $\boldsymbol{\omega}_n \sim \mathcal{N}(\boldsymbol{0}_{n_t}, \boldsymbol{Q})$, where $\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}^{x,x} & \boldsymbol{Q}^{x,y} \\ [\boldsymbol{Q}^{x,y}]^T & \boldsymbol{Q}^{y,y} \end{pmatrix}$.

The model (1) extends the conventional linear Gaussian system while preserving Kalman-like algorithms (*i.e.* optimal, exact and fast algorithms) for filtering and smoothing data. Indeed, there exists additional cross-dependences between successive states and successive observations that classical linear Gaussian system can not take into account ($\boldsymbol{F}^{x,y} = \boldsymbol{0}$ and $\boldsymbol{F}^{y,y} = \boldsymbol{0}$).

Let us consider an example to show where such a PLGS can be useful in practice. Consider the case of two independent sensors which measure respectively the position p^{obs} and the velocity v^{obs} of an object. Each sensor generates its own measurement noise, which are reasonably assumed to be independent. In this configuration, we expect the velocity measurement to enhance the estimation of the object's position p. So, $\boldsymbol{t}_n = (p_n, p_n^{obs}, v_n^{obs})$ and matrix $\boldsymbol{F}$ writes

$$\boldsymbol{F} = \begin{bmatrix} \begin{bmatrix} 1 \end{bmatrix} & \begin{bmatrix} T_s & 0 \end{bmatrix} \\ \begin{bmatrix} * \\ 1 \end{bmatrix} & \begin{bmatrix} * & * \\ 0 & 0 \end{bmatrix} \end{bmatrix},$$

where T_s is the sampling period. This is obviously not the transition matrix of a KF because $\boldsymbol{F}^{x,y} \neq \boldsymbol{0}$. The second line of $\boldsymbol{F}$ depends on the velocity property of the object and should be adapted to the problem of interest. Suppose now that we do not know exactly the velocity properties. Hence, we seek to estimate $\boldsymbol{F}$.

The maximum-likelihood estimation of parameters by means of EM algorithm [3], [4] is widely used in signal and image processing. The principle of EM has been applied to different models, including finite mixture models [5], [6], hidden Markov chains [7], hidden Markov random fields [8], linear Gaussian systems [9] and the PLGS [10], [11], and even non-linear systems [12]. When trying to estimate all model parameters, as in previous model, we will talk about full-learning algorithms, in contrast with partial-learning algorithms we discuss later.

Due to well-known identifiability problems in such systems, the restoration of a signal using an EM algorithm can be very erroneous since the estimated set of parameters can be far from the true one. In this work, we propose to exploit possible a priori knowledge on the physical system to constrain EM, to reduce the dimension of search space and so to improve parameter estimation efficiency and robustness. This strategy, known as partial-learning, has only been addressed in Gaussian mixture models [13], [14], in hidden Markov chains [15] and in linear systems [16]. The problem is more complex in PLGS, given the higher dimension of $\boldsymbol{F}$ and $\boldsymbol{Q}$. One crucial point is that it is not possible to force any type of constraint and to get a closed-form solution for parameter re-estimation formulas within EM recursions. However, as exposed later, it is still possible to define configurations of parameters which give a closed-form solution covering many practical cases.

This paper extends the work done by E. Holmes [16] on linear Gaussian systems to PLGS, with two main contributions: (1) the characterization of constraints that still allow strict EM re-estimation formulas and, (2) a robust version based on the computation of triangular square root of covariance matrices instead of the matrices themselves. The remainder of the paper is organized as follows. Firstly, Section II presents constraints that still allow an EM-based partial-learning algorithm for the pairwise linear system. Then, a robust version of this EM algorithm is proposed in Section III, based on a previous work regarding full-learning estimation [11]. Examples of realistic simulations are reported in Section IV to illustrate the interest and the robustness of the partial-learning algorithm for signal restoration. Finally, main conclusions and perspectives are presented in Section V.

II. Partial learning EM-PLGS algorithm

This Section intends to present configurations of parameters for which a constrained estimation of PLGS parameters with strict EM remains possible.

V. Némesin, Aix-Marseille Université, Centrale Marseille, CNRS, Institut Fresnel, UMR 7249, 13013 Marseille, France, e-mail: *valerian.nemesin@fresnel.fr.* S. Derrode, École Centrale de Lyon, CNRS, LIRIS, UMR 5205, 69134 Lyon, France, e-mail: *stephane.derrode@ec-lyon.fr.*

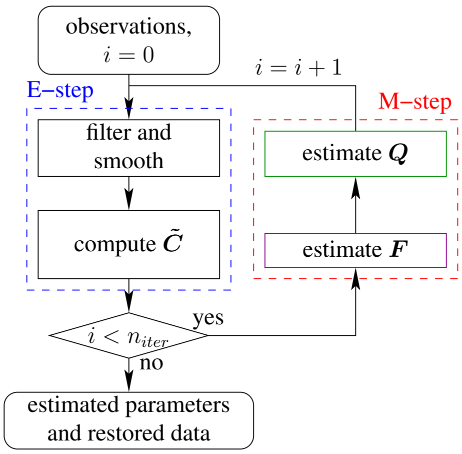

observations,
$i=0$
$i=i+1$
E−step
M−step
filter and smooth
compute $\tilde{\boldsymbol{C}}$
estimate $\boldsymbol{Q}$
estimate $\boldsymbol{F}$
$i<n_{iter}$
yes
no
estimated parameters and restored data

Fig. 1: Flowchart of the EM-PLGS algorithm.

The EM algorithm search to approach the maximum of likelihood function to estimate parameters. It is an iterative algorithm in which each iteration[1] decomposes into two steps [11]:

- the Expectation step (E-step) evaluates the auxiliary likelihood function from probability density functions of smoothing states $\boldsymbol{x}_{n|N}$

$$\begin{aligned} g_{\tilde{\boldsymbol{C}}}(\boldsymbol{Q},\boldsymbol{F}) &= -\frac{1}{2}\left[N+1\right]\log|\boldsymbol{Q}| \\ &- \frac{1}{2}\mathrm{tr}\left[\boldsymbol{Q}^{-1}\left[\boldsymbol{F}, \quad -\boldsymbol{I}\right]\tilde{\boldsymbol{C}}\begin{bmatrix}\boldsymbol{F}^T\\ -\boldsymbol{I}\end{bmatrix}\right], \end{aligned} \tag{2}$$

  where $\tilde{\boldsymbol{C}}$ is the sum of correlation matrices of the augmented smoothing states $\left[\boldsymbol{t}_{n|N},\boldsymbol{t}_{n+1|N}\right]$ (see [11, section II] for details).
- the Maximization step (M-step) re-estimates PLGS parameters by maximizing (2).

It is not possible to impose whatever constraint on $\boldsymbol{F}$ and $\boldsymbol{Q}$ while keeping strict EM learning principle. So we next present a series of constraints that preserve EM and which only impact its M-step so that the flowchart sketched in Fig. 1 remains valid. The term "constraints" is used to express a priori knowledge on the PGLS model. In order to preserve a strict EM algorithm (exact closed-form re-estimation formulas for PGLS parameters), we impose $\boldsymbol{F}$ to be processed row-wise (blocks $\boldsymbol{F}_i$), while forcing a decorrelation of corresponding sub-noise blocks $\boldsymbol{Q}_i$. For example, to constrain observations and states separately $\boldsymbol{F}^{y,t}=[\boldsymbol{F}^{y,x},\boldsymbol{F}^{y,y}]$ and $\boldsymbol{F}^{x,t}=[\boldsymbol{F}^{x,x},\boldsymbol{F}^{x,y}]$, we have to assume $\boldsymbol{Q}^{y,x}=\boldsymbol{0}$.

Remark: We must mention that the EM algorithm designed below does not work in case of singular $\boldsymbol{Q}^{y,y}$ (*e.g.* when observations are perfect).

A. *Model noise decorrelation*

In order to estimate transition matrix $\boldsymbol{F}$ using row-wise blocks, noise covariance matrix $\boldsymbol{Q}$ has to be block-diagonal, *i.e.* noise decomposes into n_{IN} independent sub-noises with covariance matrices $\{\boldsymbol{Q}_i\}_{i=1:n_{IN}}$, and $\boldsymbol{F}$ is estimated accordingly. Hence, matrices $\boldsymbol{Q}$ and $\boldsymbol{F}$ are written

$$\boldsymbol{Q} = \sum_{i=1}^{n_{IN}} \boldsymbol{P}_i^T\boldsymbol{Q}_i\boldsymbol{P}_i, \tag{3}$$

$$\boldsymbol{F} = \sum_{i=1}^{n_{IN}} \boldsymbol{P}_i^T\boldsymbol{F}_i, \tag{4}$$

where $\{\boldsymbol{P}_i\}_{i=1:n_{IN}}$ are the projectors of diagonal blocks of $\boldsymbol{Q}$. We recall that projectors are mutually orthogonal ($\forall i\neq j, \boldsymbol{P}_i^T\boldsymbol{P}_j=\boldsymbol{0}$).

Under these hypotheses, auxiliary likelihood function in (2) decomposes into n_{IN} independent sub-functions according to

$$g_{\tilde{\boldsymbol{C}}}(\boldsymbol{Q},\boldsymbol{F}) = \sum_{i=1}^{n_{IN}} g_{\tilde{\boldsymbol{C}}_i}(\boldsymbol{Q}_i,\boldsymbol{F}_i), \tag{5}$$

where

$$\tilde{\boldsymbol{C}}_i = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{P}_i\end{bmatrix}\tilde{\boldsymbol{C}}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{P}_i^T\end{bmatrix}, \tag{6}$$

which can be maximized separately. Let see now how to maximize each auxiliary sub-functions $g_{\tilde{\boldsymbol{C}}_i}(\boldsymbol{Q}_i,\boldsymbol{F}_i)$ with respect to $\boldsymbol{F}_i$ and to $\boldsymbol{Q}_i$ in next subsections.

B. *Constraining row-wise blocks of $\boldsymbol{F}$*

Each matrix $\boldsymbol{F}_i$ can always be written

$$\boldsymbol{F}_i = \boldsymbol{F}_i^0 + \boldsymbol{F}_i^b, \tag{7}$$

where $\boldsymbol{F}_i^0$ is a known matrix and $\boldsymbol{F}_i^b$ is a matrix to be estimated. The estimation of $\boldsymbol{F}_i^b$ is obtained by maximizing the auxiliary sub-function in (5). Maximizing $g_{\tilde{\boldsymbol{C}}_i}(\boldsymbol{Q}_i,\boldsymbol{F}_i)$ is equivalent to maximize $g_{\tilde{\boldsymbol{C}}_i^b}(\boldsymbol{Q}_i,\boldsymbol{F}_i^b)$ with

$$\tilde{\boldsymbol{C}}_i^b = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{F}_i^0 & \boldsymbol{I}\end{bmatrix}\tilde{\boldsymbol{C}}_i\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{F}_i^0 & \boldsymbol{I}\end{bmatrix}^T. \tag{8}$$

Closed-up form solutions were obtained for the four following matrix shapes:

(0) $\boldsymbol{F}_i^b=\boldsymbol{0}$, nothing has to be done.

(1) $\boldsymbol{F}_i^b=\boldsymbol{G}_i$, where $\boldsymbol{G}_i$ is a $(n_i$ by $n_t)$-matrix to be fully estimated. In this case, matrix $\hat{\boldsymbol{F}}_i^b$ which maximizes the auxiliary sub-function $g_{\tilde{\boldsymbol{C}}_i^b}(\boldsymbol{Q}_i,\boldsymbol{F}_i^b)$ is given by

$$\hat{\boldsymbol{F}}_i^b = \tilde{\boldsymbol{C}}_i^{b,[1,0]}\left[\tilde{\boldsymbol{C}}_i^{b,[0,0]}\right]^{-1}, \tag{9}$$

where $\tilde{\boldsymbol{C}}_i^{b,[0,0]}$, $\tilde{\boldsymbol{C}}_i^{b,[1,0]}$ correspond to the matrix views of $\tilde{\boldsymbol{C}}_i^b$ with respective sizes $(n_t$ by $n_t)$ and $(n_i$ by $n_t)$ according to

$$\tilde{\boldsymbol{C}}_i^b = \begin{bmatrix}\tilde{\boldsymbol{C}}_i^{b,[0,0]} & \tilde{\boldsymbol{C}}_i^{b,[0,1]}\\ \tilde{\boldsymbol{C}}_i^{b,[1,0]} & \tilde{\boldsymbol{C}}_i^{b,[1,1]}\end{bmatrix}.$$

(2) $\boldsymbol{F}_i^b = \boldsymbol{G}_i\boldsymbol{M}_i$, where $\boldsymbol{G}_i$ is a $(n_i$ by $n_i^G)$-matrix to be estimated and $\boldsymbol{M}_i$ is a known and full-rank $(n_i^G$ by $n_t)$-matrix *i.e.* $rk(\boldsymbol{M}_i)=n_i^G$. In this case, matrix $\hat{\boldsymbol{F}}_i^b$ which maximizes the auxiliary sub-function $g_{\tilde{\boldsymbol{C}}_i^b}(\boldsymbol{Q}_i,\boldsymbol{F}_i^b)$ is given by

$$\hat{\boldsymbol{F}}_i^b = \tilde{\boldsymbol{C}}_i^{b,[1,0]}\boldsymbol{M}_i\left[\boldsymbol{M}_i\tilde{\boldsymbol{C}}_i^{b,[0,0]}\boldsymbol{M}_i^T\right]^{-1}\boldsymbol{M}_i. \tag{10}$$

Examples:

- Full-learning in PLGS is obtained by setting $\boldsymbol{G}_1 = \begin{bmatrix}\boldsymbol{F}^{x,x} & \boldsymbol{F}^{x,y}\\ \boldsymbol{F}^{y,x} & \boldsymbol{F}^{y,y}\end{bmatrix}$, $\boldsymbol{M}_1=\boldsymbol{I}$ and $\boldsymbol{F}_1^0=\boldsymbol{0}$, in which case eq. (4) writes $\boldsymbol{F}=\boldsymbol{F}_1$ and $\tilde{\boldsymbol{C}}_i^b=\tilde{\boldsymbol{C}}_i$. So that, eq. (10) is, up to notation, exactly the one given in [10].

[1] The index j of EM iteration is intentionally forgotten to improve readability.

- The classical linear Gaussian system is another example of constrained system where $\boldsymbol{G}_1 = \begin{bmatrix} \boldsymbol{F}^{x,x} \\ \boldsymbol{F}^{y,x} \end{bmatrix}$, $\boldsymbol{M}_1 = \begin{bmatrix} \boldsymbol{I}, & \boldsymbol{0} \end{bmatrix}$ and $\boldsymbol{F}_1^0 = \boldsymbol{0}$, in which case eq. (4) also writes $\boldsymbol{F} = \boldsymbol{F}_1$ and $\tilde{\boldsymbol{C}}_i^b = \tilde{\boldsymbol{C}}_i$. So that, eq. (10) is, up to notation, exactly the one given in [17, chapter 6, page 343].

(3) $\boldsymbol{F}_i^b = \sum_{j=1}^{n_i^\lambda} \lambda_i^j \boldsymbol{U}_i^j$ where $\left\{\boldsymbol{U}_i^j\right\}_{j=1:n_i^\lambda}$ are known and independent matrices, and $\boldsymbol{\Lambda}_i = \left(\lambda_i^j\right)_{j=1:n_i^\lambda}$ is the vector of parameters to be estimated. Furthermore, matrix $\boldsymbol{Q}_i$ defined in (3), has to be proportional to a known symmetrical positive-definite matrix denoted $\boldsymbol{Q}_i^0$. In this case, the vector $\hat{\boldsymbol{\Lambda}}_i$ which maximizes the auxiliary sub-function $g_{\tilde{\boldsymbol{C}}_i^b}(\boldsymbol{Q}_i, \boldsymbol{F}_i^b)$ is given by

$$\hat{\boldsymbol{\Lambda}}_i = \boldsymbol{A}_i^{-1} \boldsymbol{B}_i, \tag{11}$$

where

$$\boldsymbol{A}_i = \sum_{k=1}^{n_i^\lambda} \sum_{l=1}^{n_i^\lambda} \operatorname{tr}\left[\left[\boldsymbol{Q}_i^0\right]^{-1} \boldsymbol{U}_i^k \tilde{\boldsymbol{C}}_i^{b,[0,0]} \boldsymbol{U}_i^l\right] \boldsymbol{E}_{k,l}, \tag{12}$$

with $\boldsymbol{E}_{k,l}$, the $(k,l)^{th}$-matrix canonical element and where

$$\boldsymbol{B}_i = \sum_{k=1}^{n_i^\lambda} \operatorname{tr}\left[\left[\boldsymbol{Q}_i^0\right]^{-1} \boldsymbol{U}_i^k \tilde{\boldsymbol{C}}_i^{b,[0,1]}\right] \boldsymbol{e}_k, \tag{13}$$

with $\boldsymbol{e}_k$, the k^{th} vector canonical element.

Example: Let consider the following PLGS which allows to estimate the gain between observations and hidden states when the transition matrix is partially known:

- For $\boldsymbol{F}$: $\boldsymbol{F}^{x,x} = 0.5$, $\boldsymbol{F}^{x,y} = 0.5$, $\boldsymbol{F}^{y,y} = 0$ and $\boldsymbol{F}^{y,x} = \lambda$ where λ has to be estimated. Hence, $\boldsymbol{F} = [\boldsymbol{F}_1^T, \boldsymbol{F}_2^T]^T$ with $\boldsymbol{F}_1 = [\boldsymbol{F}^{x,x}, \boldsymbol{F}^{x,y}] = [0.5, 0.5]$ and $\boldsymbol{F}_2 = [\boldsymbol{F}^{y,x}, \boldsymbol{F}^{y,y}] = [\lambda, 0]$.
- For $\boldsymbol{Q}$: $\boldsymbol{Q}^{x,x} = \boldsymbol{Q}_1 = \sigma^2$, $\boldsymbol{Q}^{y,y} = \boldsymbol{Q}_2 = \gamma^2$ and $\boldsymbol{Q}^{x,y} = 0$, where γ and σ have to be estimated.

In this example, blocks of $\boldsymbol{F}$ and $\boldsymbol{Q}$ have respective projectors $\boldsymbol{P}_1 = [1, 0]$, $\boldsymbol{P}_2 = [0, 1]$. Block $\boldsymbol{F}_1$ is type-(0) and block $\boldsymbol{F}_2$ is type-(3) with $n_2^\lambda = 1$, $\boldsymbol{F}_2^0 = [0, 0]$, and $\boldsymbol{U}_2^1 = [1, 0]$.

Finally, the estimation of $\boldsymbol{F}$ writes

$$\hat{\boldsymbol{F}} = \sum_{i=1}^{n_{IN}} \boldsymbol{P}_i^T \hat{\boldsymbol{F}}_i. \tag{14}$$

We now assume that $\boldsymbol{F}$ has been estimated, so we next write $\hat{\boldsymbol{F}}$ instead of $\boldsymbol{F}$.

C. Constraining blocks of $\boldsymbol{Q}$

Recalling that the noise covariance matrix $\boldsymbol{Q}$ is block-diagonal, each block $\boldsymbol{Q}_i$ ($i \in [1 : n_{QB})$ is assumed to be unitary-equivalent to a block-diagonal matrix $\boldsymbol{Q}_i^M$, *i.e.* $\boldsymbol{Q}_i = \boldsymbol{M}_i \boldsymbol{Q}_i^M \boldsymbol{M}_i^T$, where $\boldsymbol{M}_i$ is a known unitary matrix. Hence, $\boldsymbol{Q}$ is unitary-equivalent to a block-diagonal matrix denoted $\boldsymbol{Q}^M$, *i.e.* $\boldsymbol{Q} = \boldsymbol{M} \boldsymbol{Q}^M \boldsymbol{M}^T$, where

$$\boldsymbol{M} = \sum_{i=1}^{n_{IN}} \boldsymbol{M}_i \boldsymbol{P}_i^T.$$

Then, auxiliary likelihood function can be expressed using $\boldsymbol{Q}^M$ by $g_{\tilde{\boldsymbol{C}}}(\boldsymbol{Q}, \hat{\boldsymbol{F}}) = g_{\tilde{\boldsymbol{C}}^M}(\boldsymbol{Q}^M, \widehat{\boldsymbol{F}^M})$, where

$$\widehat{\boldsymbol{F}^M} = \boldsymbol{M}^T \hat{\boldsymbol{F}}, \tag{15}$$

$$\tilde{\boldsymbol{C}}^M = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^T \end{bmatrix} \tilde{\boldsymbol{C}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}. \tag{16}$$

So, $\boldsymbol{Q}^M$ writes

$$\boldsymbol{Q}^M = \sum_{i=1}^{n_{QB}} \left[\boldsymbol{P}_i^M\right]^T \boldsymbol{Q}_i^M \boldsymbol{P}_i^M, \tag{17}$$

where $\left\{\boldsymbol{P}_i^M\right\}_{i=1:n_{QB}}$ are the projectors of diagonal blocks $\boldsymbol{Q}_i^M$. Hence, auxiliary likelihood function decomposes into n_{QB} independent sub-functions according to

$$g_{\tilde{\boldsymbol{C}}}(\boldsymbol{Q}, \hat{\boldsymbol{F}}) = \sum_{i=1}^{n_{QB}} g_{\tilde{\boldsymbol{C}}_i^M}(\boldsymbol{Q}_i^M, \widehat{\boldsymbol{F}_i^M}), \tag{18}$$

where

$$\widehat{\boldsymbol{F}_i^M} = \boldsymbol{P}_i^M \widehat{\boldsymbol{F}^M}, \tag{19}$$

$$\tilde{\boldsymbol{C}}_i^M = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_i^M \end{bmatrix} \tilde{\boldsymbol{C}}^M \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_i^M \end{bmatrix}^T. \tag{20}$$

Blocks are supposed to be independent, so each sub-function can be maximized individually. Closed-up form solutions were obtained for the four following matrix shapes:

(O) $\boldsymbol{Q}_i^M = \boldsymbol{Q}_i^0$, no estimation is required.

(A) $\boldsymbol{Q}_i^M$ has to be fully estimated. In this case, the covariance matrix $\widehat{\boldsymbol{Q}_i^M}$, which maximizes the auxiliary sub-function, is given by

$$\widehat{\boldsymbol{Q}_i^M} = \frac{1}{N+1}\left[\left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right] \tilde{\boldsymbol{C}}_i^M \left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right]^T\right]. \tag{21}$$

(B) $\boldsymbol{Q}_i^M = \lambda_i \boldsymbol{Q}_i^0$, where λ_i is a scalar to be estimated and $\boldsymbol{Q}_i^0$ a known matrix. In this case, the scalar $\hat{\lambda}_i$, which maximizes the auxiliary sub-function, is given by

$$\begin{aligned} \hat{\lambda}_i = \; & \frac{1}{(N+1)n_i^M} tr\left[\, \left[\boldsymbol{Q}_i^0\right]^{-1} \right. \\ & \left. \left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right] \tilde{\boldsymbol{C}}_i^M \left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right]^T \right]. \end{aligned} \tag{22}$$

(C) $\boldsymbol{Q}_i^M = \sum_{j=1}^{n_i^B} \left[\boldsymbol{P}_i^j\right]^T \boldsymbol{M}_i^j \boldsymbol{R}_i^M \left[\boldsymbol{M}_i^j\right]^T \boldsymbol{P}_i^j$, where $\left\{\boldsymbol{M}_i^j\right\}_{j=1:n_i^B}$ are known invertible matrices, $\left\{\boldsymbol{P}_i^j\right\}_{j=1:n_i^B}$ are orthogonal projectors and $\boldsymbol{R}_i^M$ is a covariance matrix to be estimated. Matrix $\widehat{\boldsymbol{R}_i^M}$, which maximizes the auxiliary sub-function, is given by

$$\begin{aligned} \widehat{\boldsymbol{R}_i^M} = \; & \frac{1}{(N+1)n_i^B} \sum_{j=1}^{n_i^B} \Big[\\ & \left[\boldsymbol{M}_i^j\right]^{-1} \boldsymbol{P}_i^j \left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right] \tilde{\boldsymbol{C}}_i^M \\ & \left[\widehat{\boldsymbol{F}_i^M}, \quad -\boldsymbol{I}\right]^T \left[\boldsymbol{P}_i^j\right]^T \left[\boldsymbol{M}_i^j\right]^{-T} \Big]. \end{aligned} \tag{23}$$

Example: Let assume a system where two identical sensors measure a physical quantity. It is legitimate to want to fix the same measurement noise covariance $\boldsymbol{Q}^s$ for both sensors. So, the measurement covariance matrix is a type-(C) matrix, with shape

$$\boldsymbol{Q}^{y,y} = \boldsymbol{Q}_2^M = \begin{bmatrix} \boldsymbol{Q}^s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}^s \end{bmatrix},$$

with

$$\boldsymbol{P}_2^1 = \begin{bmatrix} \boldsymbol{I}, & \boldsymbol{0} \end{bmatrix}, \boldsymbol{P}_2^2 = \begin{bmatrix} \boldsymbol{0}, & \boldsymbol{I} \end{bmatrix},$$
$$\boldsymbol{M}_2^1 = \boldsymbol{M}_2^2 = \boldsymbol{I} \text{ and } \boldsymbol{R}_2 = \boldsymbol{Q}^s.$$

In this section, we presented an EM algorithm that takes into account constraints on matrices $\boldsymbol{F}$ and $\boldsymbol{Q}$ to allow partial-learning using strict EM principle. Despite constraints may seem restrictive, this algorithm is very general as illustrated in the Section IV. Whatever, this kind of algorithm suffers from numerical instabilities [11]. Next section is devoted to design a robust version of this partial-learning algorithm.

III. Robust algorithm

Following [18], we now propose a robust version of the partial-learning algorithm which makes use of only a triangular square-root $\boldsymbol{S}$ for each positive-definite matrix $\boldsymbol{P}$ ($\boldsymbol{P} = \boldsymbol{S}^T \boldsymbol{S}$) instead of $\boldsymbol{P}$ itself. Robust algorithm is almost the same as the one presented in [11] except for the M-step which is modified to take into account the constraints for partial-learning. The only difference lies in the way to compute $\hat{\boldsymbol{Q}}^{1/2}$ from matrices $\hat{\boldsymbol{F}}$ and $\tilde{\boldsymbol{C}}^{\frac{1}{2}}$ [11, Section III].

The algorithm decomposes into four steps[2]:

(1) First, a square root of $\tilde{\boldsymbol{C}}^M$ (16) is given by QR decomposition

$$\left[\tilde{\boldsymbol{C}}^M\right]^{\frac{1}{2}} = \boldsymbol{Z}^* \tilde{\boldsymbol{C}}^{\frac{1}{2}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}. \tag{24}$$

(2) Then each block $\boldsymbol{Q}_i^M$ is estimated using the following algorithm

a) A square root of $\tilde{\boldsymbol{C}}_i^M$ (20) is obtained by QR decomposition

$$\left[\tilde{\boldsymbol{C}}_i^M\right]^{\frac{1}{2}} = \boldsymbol{Z}^* \left[\tilde{\boldsymbol{C}}^M\right]^{\frac{1}{2}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_i^M \end{bmatrix}^T. \tag{25}$$

b) $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}}$ is computed depending on its block-type.

For a type-(O) block, $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}} = \left[\boldsymbol{Q}_i^0\right]^{\frac{1}{2}}$.

For a type-(A) block, $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}}$ is computed according to QR decomposition

$$\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}} = \frac{1}{\sqrt{N+1}} \boldsymbol{Z}^* \left[\tilde{\boldsymbol{C}}_i^M\right]^{\frac{1}{2}} \left[\boldsymbol{F}_i^M, \quad -\boldsymbol{I}\right]^T. \tag{26}$$

For a type-(B) block, $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}}$ is computed according to (22).

For a type-(C) block, $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}}$ is computed according to the following method:

- A square root of $\boldsymbol{R}_i^M$ (23) is computed according to QR algorithm: $\left[\boldsymbol{R}_i^M\right]^{\frac{1}{2}} = \frac{1}{(N+1)n_i^B} \left[\boldsymbol{W}_i^{n_i^B}\right]^{\frac{1}{2}}$ where

$$\begin{aligned} \boldsymbol{W}_i^k \quad = \quad & \sum_{j=1}^{k} \Big[\left[\boldsymbol{M}_i^j\right]^{-T} \boldsymbol{P}_i^j \left[\boldsymbol{F}_i^M, \quad -\boldsymbol{I}\right] \\ & \tilde{\boldsymbol{C}}_i^M \left[\boldsymbol{F}_i^M, \quad -\boldsymbol{I}\right]^T \left[\boldsymbol{P}_i^j\right]^T \left[\boldsymbol{M}_i^j\right]^{-1} \Big]. \end{aligned} \tag{27}$$

(a) For $k = 0$, $\left[\boldsymbol{W}_i^0\right]^{\frac{1}{2}} = \boldsymbol{0}$

(b) For $k = 1$ to n_i^B, $\left[\boldsymbol{W}_i^k\right]^{\frac{1}{2}}$ is computed from QR decomposition

$$\begin{bmatrix} \left[\boldsymbol{W}_i^k\right]^{\frac{1}{2}} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Z}^* \begin{bmatrix} \left[\boldsymbol{W}_i^{k-1}\right]^{\frac{1}{2}} \\ \boldsymbol{K}^i \end{bmatrix}$$

where

$$\begin{aligned} \boldsymbol{K}^i \quad = \quad & \left[\tilde{\boldsymbol{C}}_i^M\right]^{\frac{1}{2}} \left[\boldsymbol{F}_i^M, \quad -\boldsymbol{I}\right]^T \\ & \left[\boldsymbol{P}_i^j\right]^T \left[\boldsymbol{M}_i^j\right]^{-1}. \end{aligned}$$

- Then, $\left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}}$ is obtained by QR decomposition:

$$\begin{aligned} \left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}} \quad = \quad & \boldsymbol{Z}^* \sum_{j=1}^{n_i^B} \Big[\left[\boldsymbol{P}_i^j\right]^T \\ & \left[\boldsymbol{R}_i^M\right]^{\frac{1}{2}} \left[\boldsymbol{M}_i^j\right]^T \boldsymbol{P}_i^j \Big]. \end{aligned} \tag{28}$$

(3) Hence, we get a square root of $\widehat{\boldsymbol{Q}^M}$ according to

$$\left[\widehat{\boldsymbol{Q}^M}\right]^{\frac{1}{2}} = \sum_{i=1}^{n_{QB}} \left[\boldsymbol{P}_i^M\right]^T \left[\widehat{\boldsymbol{Q}_i^M}\right]^{\frac{1}{2}} \boldsymbol{P}_i^M. \tag{29}$$

(4) Finally, a square root of $\hat{\boldsymbol{Q}}$ is obtained according to the last QR decomposition

$$\left[\hat{\boldsymbol{Q}}\right]^{\frac{1}{2}} = \boldsymbol{Z}^* \underbrace{\left[\widehat{\boldsymbol{Q}^M}\right]^{\frac{1}{2}} \boldsymbol{M}^T}_{\left[\widehat{\boldsymbol{Q}^+}\right]^{\frac{1}{2}}}. \tag{30}$$

The algorithm described in this Section is robust to the propagation of the numerous covariances matrices involved in the learning algorithm. Experimental section just after exclusively makes use of this algorithm.

TABLE I: Parameters of experiment in Section IV-A. Row 1: true parameters, Row 2: initial values for EM; Rows 3 to 6: estimated parameters for the four filters described in the text ($N = 1000$ samples).

	$\boldsymbol{F}$	$\boldsymbol{Q}$
θ	$\begin{pmatrix} 0.5 & -0.5 \\ 1.0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix}$
$\theta^{(0)}$	$\begin{pmatrix} 1.0 & 0 \\ 1.0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$
$\hat{\theta}_{\text{ELGS}}$	$\begin{pmatrix} 0.62 & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.14 & 0.37 \\ 0.37 & 1.28 \end{pmatrix}$
$\hat{\theta}_{\text{FPLGS}}$	$\begin{pmatrix} 0.38 & -0.64 \\ 0.72 & 0.11 \end{pmatrix}$	$\begin{pmatrix} 0.54 & -0.27 \\ -0.27 & 0.83 \end{pmatrix}$
$\hat{\theta}_{\text{EPLGS}}$	$\begin{pmatrix} 0.49 & -0.51 \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.10 \\ -0.10 & 0.83 \end{pmatrix}$
$\hat{\theta}_{\text{CPLGS}}$	$\begin{pmatrix} 0.50 & -0.50 \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.099 & \mathbf{0} \\ \mathbf{0} & 0.99 \end{pmatrix}$

IV. Experiments

To illustrate the potential of unsupervised partial-learning in PLGS, we propose two series of experiments which are representative of different learning situations. Please note that initial expectation $\hat{\boldsymbol{t}}_0$ and covariance matrix $\boldsymbol{Q}_0$ are not estimated.

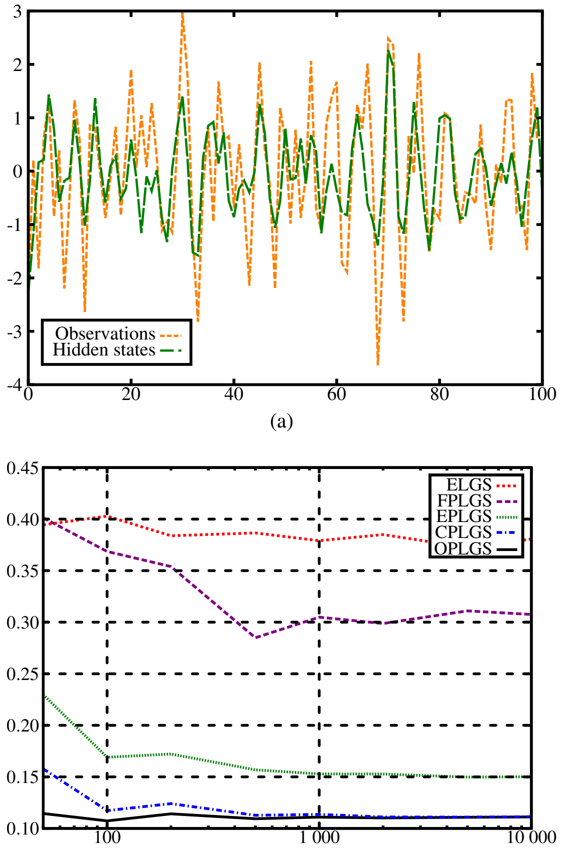
A. Simulated PLGS scalar signal

In this first experiment, all series of PLGS-signals were simulated according to parameters reported in first row of Table I. Values for $\hat{\boldsymbol{t}}_0$ and $\boldsymbol{Q}_0$ were respectively set to $[0, 0]$ and to $\boldsymbol{I}$. Then, the observations were restored by

0. the optimal PLGS (OPLGS),
1. an equivalent[3] LGS (ELGS),

[2] All unitary matrices denoted $\boldsymbol{Z}^*$ that appears in QR decompositions are never used.

[3] We used the method described in [11] in order to select an equivalent filter in which $\boldsymbol{F}^{y,x} = \boldsymbol{I}$ and $\boldsymbol{F}^{y,y} = \boldsymbol{0}$.



(a)

(b)

Fig. 2: Mean Square Error (MSE) for simulated signals with varying number of samples, for the five filters experimented in Section IV-A. A trajectory is reported in (a).

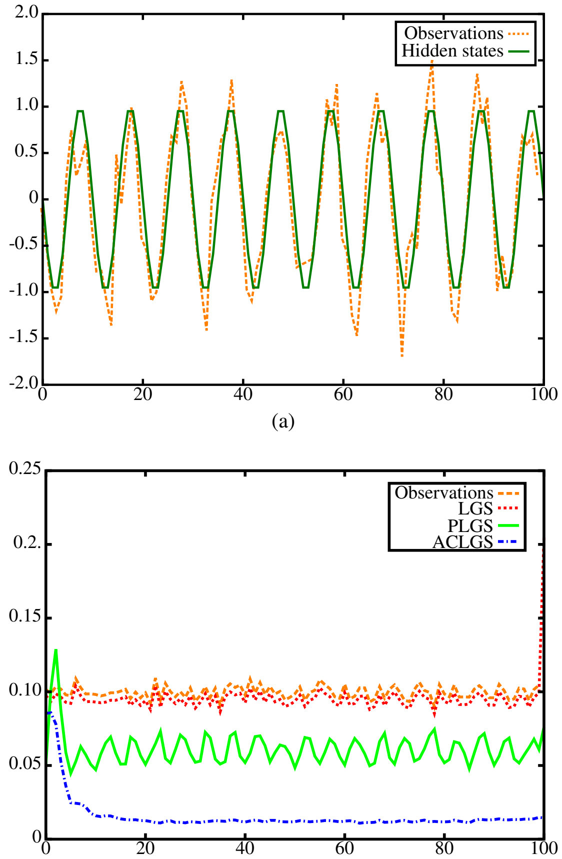
2. a free PLGS (FPLGS), that is to say full-learning PLGS
3. an equivalent[3] PLGS (EPLGS),
4. and a constrained PLGS (CPLGS).

For the CPLGS, we imposed

- a two-block decomposition of transition matrix $\boldsymbol{F}$,
- a type-(0) constraint on $\boldsymbol{F}^{y,t} = [1, 0]$,
- a type-(2) constraint on $\boldsymbol{F}^{x,t} = \lambda[1, 1] + [1, 0]$ with λ to be estimated,
- a type-(B) constraint on $\boldsymbol{Q}$: $\boldsymbol{Q} = \gamma \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$ with γ to be estimated [4].

Initial EM parameters are given in row two, and final mean estimation of 100 experiments in rows three to six for ELGS, FPLGS, EPLGS and CPLGS respectively for a $N = 1000$ samples signal. An example of trajectory is reported in Fig. 2a. Mean MSE for increasing signal sizes for the four filters are reported in Fig. 2b.

This experiments allows to draw three main conclusions: (1) As expected, an ELGS is not able to restore correctly PLGS data. Moreover, FPLGS does not give expected results due to the identifiability problem of PLGS. (2) Although we get nice estimation by EPLGS, CPLGS produces better one for short as well as long signals. (3) For signals that exceed 1000 samples, CPLGS has the same performances than the optimal filter!



(a)

(b)

Fig. 3: Instantaneous MSE for simulated signals with for $N = 100$ samples for the three filters defined in Section IV-B (b). An example of trajectory is reported in (a).

TABLE II: Parameters of experiment in Section IV-B. Rows 1 and 2: initial values for EM; Rows 3 to 5: estimated parameters for the three filters described in the text.

	$\boldsymbol{F}$	$\boldsymbol{Q}$
$\theta^{(0)}_{LGS/PLGS}$	$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\theta^{(0)}_{ACLGS}$	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
$\hat{\theta}_{LGS}$	$\begin{pmatrix} 0.68 & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.32 & \mathbf{0} \\ \mathbf{0} & 0.01 \end{pmatrix}$
$\hat{\theta}_{PLGS}$	$\begin{pmatrix} 1.48 & -0.83 \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.04 & 0.07 \\ 0.07 & 0.15 \end{pmatrix}$
$\hat{\theta}_{ACLGS}$	$\begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} \\ -0.38 & 0.61 & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{pmatrix}$	$\begin{pmatrix} 0.002 & \mathbf{0} & -0.003 \\ \mathbf{0} & 0.001 & \mathbf{0} \\ -0.003 & \mathbf{0} & 0.091 \end{pmatrix}$

B. *Sinusoidal signal*

In order to evaluate partial-learning EM-PLGS algorithm, we chosed to restore a noisy signal $y(t) = p(t) + b(t)$, where $p(t) = cos(0.2\pi t)$ is a pseudo-sinusoid and $b(t)$ is a white Gaussian noise with variance 0.1. Noisy signal is sampled between moments $t = 0s$ and $t = 99s$ with a sampling period $T_S = 1s$. Hence, previous model becomes

$$\underbrace{y_n}_{y(nT_S)} = p(nT_S) + \underbrace{\omega_n^y}_{b(nT_S)}$$

[4] Constraint on $\boldsymbol{Q}$ is compatible with constraints on $\boldsymbol{F}$ since we forced decorrelation between process and measurement noises.

Three filters were used to restore the signal

1. A classical LGS in which there are no correlation between measurement and process noise covariances, and with $\boldsymbol{F}^{y,x} = \boldsymbol{I}$ and $\boldsymbol{F}^{y,y} = \boldsymbol{0}$.
2. A PLGS in which $\boldsymbol{F}^{y,x} = \boldsymbol{I}$ and $\boldsymbol{F}^{y,y} = \boldsymbol{0}$.
3. An augmented and constrained LGS (ACLGS). For that, we decomposed $\boldsymbol{Q}$ and $\boldsymbol{F}$ into, respectively, two blocks $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$ and $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ with projectors $\boldsymbol{P}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\boldsymbol{P}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$. Block $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are type-(A) or free blocks. We imposed a type-(0) constraint on block $\boldsymbol{F}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$, and a type-(2) constraint on block $\boldsymbol{F}_2 = [a, b] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ with a and b to be estimated.

Table II reports EM initial parameters in rows 1 and 2 and estimated parameters in rows 3 to 5. An example of generated signal is reported in Fig. 3a, whereas instantaneous mean square restoration errors for $1,000$ experiments are reported in Fig. 3b.

This experiments allow to draw three main conclusions:

(1) Classical LGS can not restore noisy sinusoidal data.

(2) Although PLGS outperforms restoration by LGS, augmented and constrained LGS gives the best restoration. The difference between PLGS and ACLGS is that PLGS introduces measurement noise in hidden states, which degrades a little bit the restoration of this kind of signals.

(3) An augmented LGS seems to be the optimal model for restoring sinusoidal data, as the low process noise covariance matrix and the low MSE suggest.

V. Conclusion

The main interest of the Pairwise Linear Gaussian System (PLGS) relies on the possibility for the user to finely tune the amount of interactions between successive observations and states. In turn, this increases the number of parameters to be estimated if no a priori knowledge is available. In practice, due to the underlying physics of the model, some knowledge parameters are often partially known, whether they are fixed or constrained by other parameters. In this context, we have proposed an EM-based approach for unsupervised partial estimation of parameters in a PLGS, by extending the work in [16] regarding classical linear systems. Since the latter is a particular PLGS, our algorithm is directly applicable to robust partial-learning of linear Gaussian system parameters.

We have evaluated our algorithm in different configurations including the difficult case of very short signals (*i.e.* $N = 100$ samples). All experiments confirm the robustness of our approach. As expected, the constrained PLGS-EM algorithm provides a better quality estimation than the full-learning one. It is worth noting that, in some cases, the classical linear system can outperform the PLGS when using an augmented state-space, yet at the expense of a larger number of parameters to estimate.

A perspective to this work is to relax strict EM principle, allowing GEM principle instead [3], so that to increase the possibilities to constraint the model with other kind of a priori knowledge. Especially of interest, will be to impose that the model we want to estimate is stationary.

Acknowledgment

Authors would like to thank DGA (French *Direction Générale de l'Armement*) and CNRS for financial support.

References

[1] R. S. Lipster and A. N. Shiryaev, *Statistics of Random Processes: I. General theory*, 2nd ed. springer, 2001.

[2] F. Desbouvries and W. Pieczynski, "Triplet Markov models and Kalman filtering," *Comptes Rendus de l'Académie des Sciences - Mathématique - Série I*, vol. 336, no. 8, pp. 667–670, 2003.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[4] G. McLachlan and T. Krishnan, *The EM algorithm and extensions.* John Wiley & Sons, New York, Second Edition, 2008.

[5] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation*, vol. 8, pp. 129–151, 1995.

[6] G. McLachlan and D. Peel, *Finite mixture models.* John Wiley & Sons, New York, Willey Series in Probability and Statistics, 2000.

[7] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[8] J. Zhang, "The mean field theory in EM procedures for blind Markov random field image restoration," *IEEE Trans. on Image Processing*, vol. 2, no. 1, pp. 27–40, 1993.

[9] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[10] B. Ait-el-Fquih and F. Desbouvries, "Unsupervised signal restoration in partially observed Markov chains," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'06)*, Toulouse, France, 2006.

[11] V. Némesin and S. Derrode, "Robust blind pairwise Kalman algorithms using QR decompositions," *Signal Processing, IEEE Transactions on*, vol. 61, no. 1, pp. 5–9, 2013.

[12] Z. Ghahramani and S. T. Roweis, "Learning nonlinear dynamical systems using an EM algorithm," in *Advances in Neural Information Processing Systems 11.* MIT Press, 1998, pp. 599–605.

[13] M. Welling and M. Weber, "A constrained EM algorithm for independent component analysis," *Neural Computation*, vol. 13, pp. 677–689, 2001.

[14] S. Ingrassia and R. Rocci, "Monotone constrained EM algorithms for multinormal mixture models," in *Data Analysis, Classification and the Forward Search*, ser. Studies in Classification, Data Analysis, and Knowledge Organization. Springer Berlin Heidelberg, 2006, pp. 111–118.

[15] S. Roweis, "Constrained hidden Markov models," in *Advances in Neural Information Processing Systems 12 (NIPS'99).* MIT Press, 1999, pp. 782–788.

[16] E. Eli Holmes, "Derivation of the EM algorithm for constrained and unconstrained MARSS models," Northwest Fisheries Science Center, Tech. Rep., 2013.

[17] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications, second edition.* Springer, 2006.

[18] G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering.* Magellan Book Company, 1993.