Graphical Inference in Linear-Gaussian State-Space Models

Víctor Elvira, Emilie Chouzenoux

Graphical Inference in Linear-Gaussian State-Space Models

Víctor Elvira, *Senior Member, IEEE,* and Émilie Chouzenoux, *Senior Member, IEEE*

***Abstract*—State-space models (SSM) are central to describe time-varying complex systems in countless signal processing applications such as remote sensing, networks, biomedicine, and finance to name a few. Inference and prediction in SSMs are possible when the model parameters are known, which is rarely the case. The estimation of these parameters is crucial, not only for performing statistical analysis, but also for uncovering the underlying structure of complex phenomena. In this paper, we focus on the linear-Gaussian model, arguably the most celebrated SSM, and particularly in the challenging task of estimating the transition matrix that encodes the Markovian dependencies in the evolution of the multi-variate state. We introduce a novel perspective by relating this matrix to the adjacency matrix of a directed graph, also interpreted as the causal relationship among state dimensions in the Granger-causality sense. Under this perspective, we propose a new method called GraphEM based on the well sounded expectation-maximization (EM) methodology for inferring the transition matrix jointly with the smoothing/filtering of the observed data. We propose an advanced convex optimization solver relying on a consensus-based implementation of a proximal splitting strategy for solving the M-step. This approach enables an efficient and versatile processing of various sophisticated priors on the graph structure, such as parsimony constraints, while benefiting from convergence guarantees. We demonstrate the good performance and the interpretable results of GraphEM by means of two sets of numerical examples.**

***Index Terms*—State-space modeling, graphical inference, sparsity, proximal methods, primal-dual algorithms, Kalman filtering, EM algorithm.**

I. Introduction

State-space modeling is widely used to describe complex systems in applications of science and engineering [1], [2], [3]. These discrete-time models are described by a hidden (or latent) state that evolves in a Markovian manner over time through arbitrarily complicated dynamics, which allows for a realistic modeling of complex phenomena. The observations are sequentially collected and linked to the hidden states.

This modeling aims at mimicking complex dynamic systems in an accurate manner through a hidden latent process, which sometimes is of reduced dimension w.r.t. the multivariate time series. Alternatively, the state can be very high-dimensional and can be interpreted, e.g., each dimension of the state represents a physical magnitude in a set of 3D points, but observation cannot be acquired in all locations. This scenario is common in complex systems, which are usually composed of many simpler units. Interestingly, in those systems, each unit usually interacts with very few others [4]. For instance, the evolution of the atmosphere can be modeled with a hidden state that captures physical properties at millions of geographical locations, but from one time step to the next one, each location is only affected by few close locations [5], [6]. Therefore, accurate and efficient inference requires realistic modeling (e.g., high-dimensional state) combined with the incorporation of prior knowledge of the inner structure of the system (e.g., sparsity in the way the dimensions of the state interact). This paper focuses on the relevant linear-Gaussian state-space model (LG-SSM). This model allows for exact inference, when the model parameters are known, through the Kalman filter and the Rauch-Tung-Striebel (RTS) smoother [3, Chapter 8]. In nonlinear and/or non-Gaussian models, the inference is generally done via particle filtering [7], [8], e.g., the BPF [9], APF [10], IAPF [11], and OAPF [12] algorithms (see [13] for a further discussion). In all these models, the parameter estimation is generally done via particle-based methods (see for instance [14]).

Existing methods in the literature. SSMs have shown to be powerful mathematical models for time series analysis. A few alternative methods to SSMs exist, e.g., classical multivariate time series analysis models [15] or polynomial data fitting for trajectory estimation in tracking applications [16], [17] (see further discussion in [3, Chapter 1] or [18]). In the case of SSMs, and more particularly in the LG-SSM, the estimation of the model parameters is essential to tackle problems that otherwise would be unapproachable, allowing for the estimation of the mean and covariance of the hidden state through Kalman filtering and RTS smoothing. Existing methods for the estimation of model parameters in LG-SSM focus on the maximum-likelihood (ML) estimate. Two main classes of methods have been proposed in the literature [3, Chap. 12]. The first class of methods makes use of the so-called sensitivity equations [19], or on the Fisher's identity [20], [21] (see the discussion in [22, Sec. 10.2.4] for connections between both strategies), to evaluate efficiently the first and second derivatives of the likelihood function with respect to the unknown parameters. This allows to apply iterative optimizers, such as quasi-Newton [23] or Newton-Raphson [19], to obtain the ML estimate. The second class of methods relies on the expectation-minimization (EM) algorithm [24][22, Sec. 10.4][3, Sec. 12.2.3], where the maximization of the marginal likelihood is indirectly performed by iteratively maximizing (M-step) an expectation (E-step) of the

V. Elvira is with the School of Mathematics at the University of Edinburgh (UK) and The Alan Turing Institute (UK). É. Chouzenoux is with Université Paris-Saclay, Inria, CentraleSupélec, Centre de Vision Numérique (France).

V.E. and É.C. acknowledge support from the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01) and MAJIC (ANR-17-CE40-0004-01) projects. V.E. acknowledges support from the Leverhulme Research Fellowship (RF-2021-593). É.C. acknowledges support from the European Research Council Starting Grant MAJORIS ERC-2019-STG-850925.

log-likelihood. Applications of the EM strategy in the LG-SSM to various fields, e.g., finance, electrical engineering, and radar, can be found for instance in [25], [26], [27]. The main advantage of EM in this context is its simplicity in the implementation and the convergence stability, inherited from the EM machinery [28], [29]. We refer to [24, Sec. 1] for a detailed discussion of the benefits and drawbacks of each class of methods. However, none of the aforementioned methods allow to compute a maximum a posteriori (MAP) estimate of the parameters in the LG-SSM. It is possible to design naive extensions by simply incorporating a prior term on the function to maximize. However, the specific strategies cannot cope with complicated prior terms. More precisely, the methods of the first class are limited to differentiable penalty terms, preventing the use of sparsity enhancing functions and constraints, which are of high interest in this context (see the discussion below). In the case of the EM algorithm, the M-step has a closed form for very limited priors (e.g., Gaussian). It gets intractable for most priors of interest and thus the existing framework of [24] does not lead to any directly implementable algorithmic solution.

SSMs are powerful mathematical tools for forecasting and also bring interpretability about the hidden process, allowing to understand the uncovered relations in the state space. In this line, graphical modeling methods for time series have been proposed [30], [31], [32]. Such representation of multivariate sequences interactions has applications in several domains such as biology [33], [34], social network analysis [35], and neuroscience [36]. Graphical modeling often requires the introduction of sparsity priors to meet interpretability and compactness (see for instance the celebrated graphical lasso approach [37]). Spectral constraints (e.g., low rank) might also be useful to enhance clustering effects on the graphs [38]. In both cases, this yields complicated MAP formulations, involving non differentiable terms, for which available methods for LG-SSM parameter estimations cannot be applied, as discussed in the previous paragraph.

Contributions. In this paper, we propose a novel framework called GraphEM for the estimation of model parameters in the LG-SSM, using prior knowledge. While the proposed methodology can be adapted to estimate all model parameters, here we are explicit on the estimate of the transition matrix of the SSM, which is arguably the most complicated parameter to be estimated because (a) it is high-dimensional, (b) it intervenes in the auto-regressive process of the hidden state that cannot be observed, and (c) it is highly related to the inner structure of the complex system, requiring the incorporation of suitable prior knowledge. In the spirit of modeling complex systems as described above, the transition matrix is here supposed to be sparse. GraphEM brings a new perspective in state-space modeling to interpret the interactions of the state dimensions between consecutive time steps as a sparse directed graph, encoding relations among the dimensions of the hidden state. Namely, the hidden process follows an order-one auto-regressive process. We interpret the sparse transition matrix of the multi-variate process in a Granger causality manner [39]. In particular, Granger causality (also called as predictive causality) is often considered, not as a true type of causality, but just as a metric of how well one time series allows to forecast a second one [40]. From that perspective, the i, j entry in the transition matrix in LG-SSMs encodes the weight in which the j-th time series in the hidden state affects the i-th time series in the next time step, being zero if it does not have any effect. Thus, a zero in the i, j entry can be interpreted as if the j-th time series does not provide any further information to predict the i-th time series (given the other time series). In GraphEM, we allow for a variety of sparsity constraints in the transition matrix, accounting for realistic modeling in a plethora of applications. We discuss particular examples and provide simulations both in controlled scenarios and in a wireless communication problem.

GraphEM belongs to the family of EM algorithms for MAP estimation, alternating between an expectation (E)-step based on the RTS smoother that builds a majorizing function of the posterior distribution of the unknown given the data, and a sophisticated maximization (M)-step in which this function is maximized w.r.t. the unknown parameter. The proposed GraphEM algorithm involves novel methodology to incorporate realistic prior knowledge about the dynamic system, such as sparsity constraints. Specifically, the inclusion of non-Gaussian and possibly non-smooth priors requires the development of a new tailored optimization procedure in the M-step. We propose to address this challenge by resorting to a proximal primal-dual splitting methodology, that we design to suitably incorporate the desired priors. In a nutshell, the contributions of this paper are as follows:[1]

- Proposition of a novel graphical interpretation of the transition matrix within LG-SSMs based on (sparse) causal interactions among state dimensions in the Granger sense,
- Derivation of an EM-based algorithm for computing a MAP estimate of this matrix, with strong theoretical guarantees,
- Design of a convergent convex optimization procedure for an efficient implementation of the M-step, able to account for a wide class of priors on the interpreted graph,
- Presentation of two challenging numerical examples, namely a controlled scenario, and a problem of channel tracking in wireless communications. Various setups of sparse transition matrices and priors are tested, including block-sparsity penalties and nuclear norms constraints.

The rest of the paper is structured as follows. Section II describes the model, the filtering/smoothing algorithms, and the background about the EM framework. The novel GraphEM algorithm is presented in Section III, with a detailed explanation of the E-step, the proposed new optimization methodology for the M-step, a discussion on the priors, both from the application and the methodological perspectives, and a convergence theorem. The paper concludes with two numerical examples in Section IV and some concluding remarks in Section V.

[1] A limited version of this work was presented by the authors in the conference paper [41].

II. Background

A. Notation

We denote by $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$ the Euclidean norm of $\mathbf{x} \in \mathbb{R}^N$, where $\top$ states from the transpose operation and $\mathbb{R}^N$ is the N-dimensional Euclidean space. We also introduce $\|\mathbf{X}\|_F$ and $\|\mathbf{X}\|_2$, the Frobenius norm and spectral norms (i.e., largest singular value), respectively, of elements $\mathbf{X} = (X(n,\ell))_{1\leq n\leq N, 1\leq \ell \leq M} \in \mathbb{R}^{N\times M}$. $\mathbf{Id}_N$ is the identity matrix of $\mathbb{R}^N$ and tr is the trace operator. Bold symbols are used for matrix and vectors. The useful definitions of convex analysis are reminded on-the-fly throughout the paper. For these concepts, we rely on the notation in the textbook [42]. Furthermore, we introduce the shorter notation $\mathrm{ct}_{/\mathbf{A}}$, for any constant independent from a variable $\mathbf{A}$. Finally, given a sequence of elements $\{\mathbf{x}_k\}_{k=1}^K$ of length $K \geq 1$, we use the notation $\mathbf{x}_{k_1:k_2}$ to refer to the subsequence $\{\mathbf{x}_k\}_{k=k_1}^{k_2}$, for $1 \leq k_1 < k_2 \leq K$.

B. Linear state-space model

We consider the LG-SSM described, for $k = 1, \ldots, K$, as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k, \tag{1}$$
$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k, \tag{2}$$

where,

- $\{\mathbf{x}_k\}_{k=1}^K \in \mathbb{R}^{N_x}$ and $\{\mathbf{y}_k\}_{k=1}^K \in \mathbb{R}^{N_y}$, are the hidden state and the observations, respectively, at each time k,
- $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ is the transition matrix that we aim at estimating,
- $\{\mathbf{H}_k\}_{k=1}^K \in \mathbb{R}^{N_y \times N_x}$ are the observation model matrices, possibly varying with k, and are assumed to be known,
- $\{\mathbf{q}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{Q})$ is the i.i.d. state noise process, assumed to follow a zero-mean Gaussian model with known symmetric definite positive (SDP) covariance matrix $\mathbf{Q} \in \mathbb{R}^{N_x \times N_x}$,
- $\{\mathbf{r}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{R}_k)$ is the i.i.d. observation noise process, again zero-mean Gaussian with known SDP covariance matrices $\mathbf{R}_k \in \mathbb{R}^{N_y \times N_y}$.

We assume an initial state distributed such that $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with known $\boldsymbol{\mu}_0 \in \mathbb{R}^{N_x}$ and SDP $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{N_x \times N_x}$. The state and the observation noises are mutually independent and also independent of the initial state $\mathbf{x}_0$.

C. Kalman filtering and smoothing

In many applications (e.g., tracking), the goal is in the estimation of the hidden state $\{\mathbf{x}_k\}_{k=1}^K$ from observations $\{\mathbf{y}_k\}_{k=1}^K$. In the Bayesian/probabilistic setting, this translates into the computation, for every $k \in \{1, \ldots, K\}$, of the posterior distribution of $\mathbf{x}_k$. If one conditions on all observations available up to time k, $\mathbf{y}_{1:k} = \{\mathbf{y}_j\}_{j=1}^k$, then the posterior probability density function (pdf), $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, is the *filtering* distribution. Conditioning on the whole set of observations $\mathbf{y}_{1:K}$, the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ is the *smoothing* distribution.

Estimating the filtering and smoothing distributions is in general a challenging problem, since obtaining these distributions of interest is possible only in few models of interest.

Algorithm 1. *Kalman Filter*

Input. *Prior parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$; model parameters $\mathbf{A}$, $\mathbf{Q}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$; set of observations $\{\mathbf{y}_k\}_{k=1}^K$.*
Recursive step. *For $k = 1, \ldots, K$*

a) Prediction/propagation step.

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{A}\boldsymbol{\mu}_{k-1} \tag{3}$$
$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{A}\boldsymbol{\Sigma}_{k-1}\mathbf{A}^\top + \mathbf{Q} \tag{4}$$

b) Update step.

$$\boldsymbol{\nu}_k = \mathbf{H}_k \boldsymbol{\mu}_{k|k-1} \tag{5}$$
$$\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\nu}_k \tag{6}$$
$$\mathbf{S}_k = \mathbf{H}_k \boldsymbol{\Sigma}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \tag{7}$$
$$\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1} \tag{8}$$
$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k \tag{9}$$
$$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top \tag{10}$$

Output. *$\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. Then, for each $k = 1, ..., K$:*

- *state filtering pdf: $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$*
- *observation predictive pdf: $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{y}_k; \boldsymbol{\nu}_k, \mathbf{S}_k)$*

Algorithm 2. *RTS Smoother*

Input. *Filtering parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=0}^K$ from the Kalman filter; model parameters $\mathbf{A}$ and $\mathbf{Q}$.*
Initialization. *Set $\boldsymbol{\mu}_K^s = \boldsymbol{\mu}_K$ and $\boldsymbol{\Sigma}_K^s = \boldsymbol{\Sigma}_K$.*
Recursive step. *For $k = K, K-1, ..., 0$*

$$\boldsymbol{\mu}_{k+1}^- = \mathbf{A}\boldsymbol{\mu}_k \tag{11}$$
$$\boldsymbol{\Sigma}_{k+1}^- = \mathbf{A}\boldsymbol{\Sigma}_k\mathbf{A}^\top + \mathbf{Q} \tag{12}$$
$$\mathbf{G}_k = \boldsymbol{\Sigma}_k \mathbf{A}^\top \left(\boldsymbol{\Sigma}_{k+1}^-\right)^{-1} \tag{13}$$
$$\boldsymbol{\mu}_k^s = \boldsymbol{\mu}_{k|k-1} + \mathbf{G}_k \left(\boldsymbol{\mu}_{k+1}^s - \boldsymbol{\mu}_{k+1}^-\right) \tag{14}$$
$$\boldsymbol{\Sigma}_k^s = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{G}_k \left(\boldsymbol{\Sigma}_{k+1}^s - \boldsymbol{\Sigma}_{k+1}^-\right) \mathbf{G}_k^\top \tag{15}$$

Output. *$\{\boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s\}_{k=1}^K$. Then, for each $k = 1, ..., K$:*

- *state smoothing pdf: $p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$*

For instance, for the LG-SSM described in (1)-(2), it is possible to obtain the filtering and smoothing distributions, for $k = 1, \ldots, K$, in the case where the model parameters $\mathbf{A}$, $\mathbf{Q}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$ are known. Interestingly, these distributions can be obtained in an efficient sequential manner via the Kalman filter [43] and the RTS smoother [44]. Algorithm 1 describes the Kalman filter while Algorithm 2 describes the RTS smoother.

D. EM framework for parameter estimation

In this paper, we consider the more challenging setting in which some parameters of the LG-SSM are unknown, and must be estimated jointly with the hidden states inference. The problem of parameter estimation in SSM has been widely studied in the literature. Three main types of methods can be distinguished, namely (i) expectation-maximization (EM) algorithms [24], [45], [25], (ii) optimization-based methods

[23], and (iii) Monte Carlo methods [46], [47]. In the context of LG models, the EM strategy is particularly well suited, since it keeps a reduced computational cost while preserving part of the Bayesian interpretation [24]. We now describe the rationale of applying the EM strategy for the estimation of the state matrix $\mathbf{A}$ in the LG-SSM of (1)-(2). In such context, the maximum likelihood (ML) estimate of $\mathbf{A}$ is not available in a closed form [24]. Moreover, the maximum a posteriori (MAP) estimate approach is also intractable and remains unexplored to the best of our knowledge.

The MLEM algorithm builds iteratively an ML estimate of the LG-SSM parameters through the resolution of surrogate problems constructed following a majorization principle [48]. For the sake of clarity, we describe here the resulting MLEM procedure for the LG-SSM case. Note that we focus here on the estimation of $\mathbf{A}$, though the MLEM for LG-SSM was initially introduced in [24] for estimating the state/observation and covariance noise matrices. In the sequel, we will denote $(\mathbf{A}^{(i)})_{i\in\mathbb{N}} \in \mathbb{R}^{N_x\times N_x}$ the sequence of MLEM iterates, whose construction will be specified below. For every $\mathbf{x}_{0:K}$ with non zero probability, the log-likelihood function is

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}) = \log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A}) - \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}). \tag{16}$$

This function is continuously differentiable for $\mathbf{A}\in\mathbb{R}^{N_x\times N_x}$. Moreover, (16) can be easily evaluated, using its recursive form:

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}) = \sum_{k=1}^{K} \frac{1}{2}\log|2\pi\mathbf{S}_k| + \frac{1}{2}\mathbf{v}_k^\top \mathbf{S}_k^{-1}\mathbf{v}_k, \tag{17}$$

where $(\mathbf{v}_k,\mathbf{S}_k)_{1\leq k\leq K}$ are obtained by the RTS Alg. 2 run for a given transition matrix $\mathbf{A}$. The gradient and Hessian of (16) can also be evaluated with recursive formula (using, for instance, Fisher's identity [3, Chap. 12]). These properties are at the core of the optimization-based estimation methods in [19], [23], unfortunately presenting an unstable behaviour, mostly due to the non-convexity of (16). The MLEM algorithm [24] proceeds differently, by building sequential lower bounds of (16), as we describe below. Let $i\in\mathbb{N}$, associated with the current parameter estimate $\mathbf{A}^{(i)}$. We can take the expectation of (16) over all possible values of the unknown state given $\mathbf{A}^{(i)}$, by multiplying both sides of (16) by $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)})$ and integrating over all states. Since $\int \log p(\mathbf{y}_{1:K}|\mathbf{A}) p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)})d\mathbf{x}_{0:K} = \log p(\mathbf{y}_{1:K}|\mathbf{A})$ (i.e., integration of a constant quantity), then

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}) = \underbrace{\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)})\log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A})d\mathbf{x}_{0:K}}_{\triangleq q(\mathbf{A};\mathbf{A}^{(i)})} \underbrace{- \int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)})\log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A})d\mathbf{x}_{0:K}}_{\triangleq h(\mathbf{A};\mathbf{A}^{(i)})}. \tag{18}$$

The latter equation holds for any $\mathbf{A}\in\mathbb{R}^{N_x\times N_x}$, including $\mathbf{A}=\mathbf{A}^{(i)}$, i.e.,

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i)}) = q(\mathbf{A}^{(i)};\mathbf{A}^{(i)}) + h(\mathbf{A}^{(i)};\mathbf{A}^{(i)}). \tag{19}$$

Subtracting (19) from (18) yields

$$\begin{aligned}&\log p(\mathbf{y}_{1:K}|\mathbf{A}) - \log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i)})\\ &= q(\mathbf{A};\mathbf{A}^{(i)}) - q(\mathbf{A}^{(i)};\mathbf{A}^{(i)}) + h(\mathbf{A};\mathbf{A}^{(i)}) - h(\mathbf{A}^{(i)};\mathbf{A}^{(i)}).\end{aligned} \tag{20}$$

Since the entropy is upper-bounded by the cross-entropy w.r.t. any other pdf (*Gibb's inequality*),

$$h(\mathbf{A};\mathbf{A}^{(i)}) \geq h(\mathbf{A}^{(i)};\mathbf{A}^{(i)}), \tag{21}$$

where the equality holds if and only if $\mathbf{A}=\mathbf{A}^{(i)}$. We can thus conclude that

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}) - \log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i)}) \geq q(\mathbf{A};\mathbf{A}^{(i)}) - q(\mathbf{A}^{(i)};\mathbf{A}^{(i)}), \tag{22}$$

that is,

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}) \geq q(\mathbf{A};\mathbf{A}^{(i)}) + \mathrm{ct}_{/\mathbf{A}}. \tag{23}$$

Again, the equality holds in (23) if and only if $\mathbf{A}=\mathbf{A}^{(i)}$. Inequality (23) is the cornerstone of the MLEM algorithm, which follows a majoration-minimization (MM) principle [49]. At each iteration $i\in\mathbb{N}$ of the MLEM method, the E-step computes the following expectation:

$$q(\mathbf{A};\mathbf{A}^{(i)}) = \int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)})\log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A})\mathrm{d}\mathbf{x}_{0:K}, \tag{24}$$

satisfying (23). The M-step aims at maximizing $q(\mathbf{A};\mathbf{A}^{(i)})$ with respect to $\mathbf{A}$, yielding $\mathbf{A}^{(i+1)}$. Thus, by construction,

$$\log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i+1)}) \geq q(\mathbf{A}^{(i+1)};\mathbf{A}^{(i)}) + \mathrm{ct}_{/\mathbf{A}} \tag{25}$$

$$\geq q(\mathbf{A}^{(i)};\mathbf{A}^{(i)}) + \mathrm{ct}_{/\mathbf{A}} \tag{26}$$

$$= \log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i)}). \tag{27}$$

The MLEM guarantees the increase of the log-likelihood loss $\log p(\mathbf{y}_{1:K}|\mathbf{A}^{(i)})$ along iterations, which is equivalent to an increase of the ML loss [48], [28]. As shown in [24], the integral in (24) can be expressed as byproducts of the RTS smoother. This leads to the construction of an MLEM method to derive estimates of the parameters of an LG-SSM, jointly with the hidden states inference task. However, the aforementioned work did not include any prior knowledge on the parameters. Moreover, although the convergence of generic EM schemes has been established in [29], the required assumptions are not met in the case of the MLEM scheme for LG-SSM from [24], mostly due to the intricate recursive form of the ML loss. Finally, the derivations in [24] were restricted to the case of constant matrices $\mathbf{R}$ and $\mathbf{H}$ in the observation model equation (2).

III. The GraphEM algorithm

In this section, we present a generalized version of this EM approach, able to encompass time-varying observation model as well as to yield a MAP estimate of LG-SSM transition matrix, for a large class of priors. We explicit both the E and M steps, and introduce a novel efficient iterative solver for performing the latter with assessed convergence guarantees. We show the convergence of the resulting EM-based approach under reasonable assumptions.

A. Summary of GraphEM

In this section, we present a general framework for the estimation of the transition matrix $\mathbf{A}$ of the state model in Eq. (1) under suitable prior assumption. This allows to integrate useful sparsity and spectral constraints on $\mathbf{A}$, with the aim of promoting the interpretability and the stability of the inferred LG-SSM. These constraints are encoded in the prior distribution $p(\mathbf{A})$, as we discuss in Section III-D. GraphEM aims at providing the maximum a posteriori (MAP) estimator of $\mathbf{A}$. More specifically, let us denote the posterior of the unknown parameter, $p(\mathbf{A}|\mathbf{y}_{1:K})$, where the hidden states have been marginalized. It is direct to show, using Bayes rule and the (strictly increasing) logarithmic function, that the maximum of $p(\mathbf{A}|\mathbf{y}_{1:K}) \propto p(\mathbf{A})p(\mathbf{y}_{1:K}|\mathbf{A})$ coincides with the minimum of the loss function

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{L}_K(\mathbf{A}) \triangleq \mathcal{L}_0(\mathbf{A}) + \mathcal{L}_{1:K}(\mathbf{A}), \tag{28}$$

where we denote the regularization function as

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{L}_0(\mathbf{A}) \triangleq -\log p(\mathbf{A}), \tag{29}$$

and the neg-log-likelihood as

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}) \triangleq -\log p(\mathbf{y}_{1:K}|\mathbf{A}), \tag{30}$$

with $\log p(\mathbf{y}_{1:K}|\mathbf{A})$ defined in Eq. (16). As commented above, it is not straightforward to find a minimizer of (28), even for the case without regularization function.

The proposed algorithm, GraphEM, is summarized in Algorithm 3. GraphEM is a type of expectation-maximization (EM) method that runs for several iterations, alternating between the expectation (E)-step and the maximization (M)-step. The E-step can be seen as a generalization of the one in MLEM from [24], to the case of time-varying observation matrices $\{\mathbf{R}_k\}_{k=1}^K$ and $\{\mathbf{H}_k\}_{k=1}^K$. Moreover, it also accounts for a prior term on $\mathbf{A}$ (see Section III-D), so as to reach a MAP estimate of the transition matrix $\mathbf{A}$. The M-step thus becomes much more intricate than in the aforementioned MLEM. In particular, no close form is longer available for the update of the transition matrix. To overcome this challenge, we propose an iterative solver with sound convergence guarantees, relying on modern tools from convex analysis (see Section III-C).

At each iteration $i \in \mathbb{N}$, the expectation function $q(\mathbf{A}; \mathbf{A}^{(i)})$ given in (24) is first computed in the E-step. This function is created by running the Kalman filter followed by the RTS smoother with the state matrix set to the estimate of the previous iteration, i.e., equals to $\mathbf{A}^{(i)}$. We then construct

$$\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i)}) \triangleq -q(\mathbf{A}; \mathbf{A}^{(i)}) + \mathcal{L}_0(\mathbf{A}) + \mathrm{ct}_{/\mathbf{A}}, \tag{31}$$

a majorizing approximation of the MAP loss function (28). Then, a new estimate of the transition matrix, $\mathbf{A}^{(i+1)}$, is obtained in a corrected M-step, as the minimizer of the regularized surrogate in (31). As we will show in Section III-E, GraphEM aims at providing ultimately an estimate of the maximum of $p(\mathbf{A}|\mathbf{y}_{1:K})$, i.e., the MAP estimate of $\mathbf{A}$. The last iteration of GraphEM also provides, as a byproduct, the filtering and smoothing distribution, given this last version of the transition matrix.

Algorithm 3. *GraphEM algorithm*

Inputs. *Prior parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$; model parameters $\mathbf{Q}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$; set of observations $\{\mathbf{y}_k\}_{k=1}^K$, and prior $p(\mathbf{A})$. Precisions $(\varepsilon, \xi) > 0$.*
Initialization. *Set $\mathbf{A}^{(0)} \in \mathbb{R}^{N_x \times N_x}$.*
Recursive step. *For $i = 0, 1, \ldots$:*

(E step) Run the Kalman filter and RTS smoother using transition matrix $\mathbf{A}^{(i)}$.
Calculate $(\boldsymbol{\Psi}, \boldsymbol{\Delta}, \Phi)$ using (46)-(47)-(48).
Build function $\mathbf{A} \mapsto \mathcal{Q}(\mathbf{A}, \mathbf{A}^{(i)})$ using (31).
(M step) Run Algorithm 4 with precision ξ to solve $\mathbf{A}^{(i+1)} = \mathrm{argmin}_{\mathbf{A}} \mathcal{Q}(\mathbf{A}, \mathbf{A}^{(i)})$.

*If $\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F \leq \varepsilon \|\mathbf{A}^{(i)}\|_F$, **stop the recursion**.*
Output. *State filtering/smoothing pdfs along with MAP estimate of the transition matrix.*

B. Explicit E-step

In this section, we derive the explicit E-step for the case of unknown $\mathbf{A}$. Let us first define the log-likelihood of the observations and states (that we recall, are not observed) that, due to the Markovian structure of the state space model in Eq. (1), takes this form:

$$\log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\mathbf{A}) = \log p(\mathbf{x}_0) + \sum_{k=1}^{K} \log p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{A}) + \sum_{k=1}^{K} \log p(\mathbf{y}_k|\mathbf{x}_k). \tag{32}$$

Following Section II-D, we must compute the expectation function $q(\mathbf{A}; \mathbf{A}^{(i)})$, given in (24), i.e., the log-likelihood of the observations and states integrated against the smoothing posterior $p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\mathbf{A})$, in such a way the states are marginalized and, therefore, the resulting function depends only on the model parameters. Function $\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i)})$ used in the Alg. 3 is then deduced easily from (31).

In the case of the LG-SSM in Eqs. (1)-(2), we now show that there exists a closed-form expression for the integral (24). Our approach uses the outputs of the RTS smoother and generalizes [3, Theo. 12.4]. In a nutshell, we will demonstrate that (a) the three log-quantities in Eq. (32) are quadratic, and (b) the resulting integral (24) is tractable. Our proof lies in that the Kalman filter in Alg. 1 provides an exact form $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, for every $k = 1, \ldots, K$. The sequence of smoothing distributions (conditioned to the whole set of observations), can be also computed exactly by the RTS smoother in Alg. 2, yielding $p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k^{\mathrm{s}}, \boldsymbol{\Sigma}_k^{\mathrm{s}})$, for every $k = 1, \ldots, K$.

Note that the computation of (24) requires the marginalization of the three terms in (32). However, since M-step aims at minimizing $\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i)})$ w.r.t. $\mathbf{A}$, only terms of (24) depending on variable $\mathbf{A}$ are in practice needed for the update, i.e., the second term of (32):

$$f(\mathbf{x}_{1:K},\mathbf{A}) \triangleq \sum_{k=1}^{K} \log p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{A}) \tag{33}$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\left((\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1}) + \log|2\pi\mathbf{Q}|\right). \tag{34}$$

Then, skipping the constant terms independent from $\mathbf{A}$, Eq. (24) can be rewritten as

$$q(\mathbf{A};\mathbf{A}^{(i)}) = \int f(\mathbf{x}_{1:K},\mathbf{A}) p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)}) d\mathbf{x}_{0:K} + \mathrm{ct}_{/\mathbf{A}}, \tag{35}$$

$$= \int \left(-\frac{1}{2}\sum_{k=1}^{K}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})\right) \times p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)}) d\mathbf{x}_{0:K} + \mathrm{ct}_{/\mathbf{A}} \tag{36}$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\int (\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1}) \times p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A}^{(i)}) d\mathbf{x}_{0:K} + \mathrm{ct}_{/\mathbf{A}}. \tag{37}$$

Then, we marginalize part of the variables to obtain

$$q(\mathbf{A};\mathbf{A}^{(i)}) = -\frac{1}{2}\sum_{k=1}^{K}\int (\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1}) \times p(\mathbf{x}_{k:k-1}|\mathbf{y}_{1:K},\mathbf{A}^{(i)}) d\mathbf{x}_{k:k-1} + \mathrm{ct}_{/\mathbf{A}} \tag{38}$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\int (\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1}) \times \mathcal{N}(\mathbf{x}_{k:k-1}|\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1},\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1}) d\mathbf{x}_{k:k-1} + \mathrm{ct}_{/\mathbf{A}}, \tag{39}$$

where $\mathcal{N}(\mathbf{x}_{k:k-1}|\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1},\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1})$ denotes the joint smoothing distribution of two consecutive states $\mathbf{x}_{k:k-1} = [\mathbf{x}_k;\mathbf{x}_{k-1}] \in \mathbb{R}^{2N_x}$. The latter is Gaussian with mean

$$\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1} = [\boldsymbol{\mu}^{\mathrm{s}}_k;\boldsymbol{\mu}^{\mathrm{s}}_{k-1}], \tag{40}$$

and covariance

$$\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1} = [\boldsymbol{\Sigma}^{\mathrm{s}}_k, \boldsymbol{\Sigma}^{\mathrm{s}}_k\mathbf{G}^\top_{k-1}; \mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_k, \boldsymbol{\Sigma}^{\mathrm{s}}_{k-1}]. \tag{41}$$

The matrix $\mathbf{G}_k = \boldsymbol{\Sigma}_k\mathbf{A}^{(i)\top}\left(\mathbf{A}^{(i)}\boldsymbol{\Sigma}_k\mathbf{A}^{(i)\top}+\mathbf{Q}\right)$ follows from the derivation of the RTS smoother via manipulations of Gaussian pdfs (see for instance [3, Theorem 8.2]). Then, by defining $\widetilde{\mathbf{A}} = [\mathbf{Id}_{N_x}, -\mathbf{A}]$, Eq. (39) turns

$$q(\mathbf{A};\mathbf{A}^{(i)}) = -\frac{1}{2}\sum_{k=1}^{K}\int \left(\widetilde{\mathbf{A}}\mathbf{x}_{k:k-1}\right)^\top \mathbf{Q}^{-1}\left(\widetilde{\mathbf{A}}\mathbf{x}_{k:k-1}\right) \times \mathcal{N}(\mathbf{x}_{k:k-1}|\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1},\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1}) d\mathbf{x}_{k:k-1} + \mathrm{ct}_{/\mathbf{A}} \tag{42}$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\int \mathbf{x}^\top_{k:k-1}(\widetilde{\mathbf{A}}^\top\mathbf{Q}^{-1}\widetilde{\mathbf{A}})\mathbf{x}_{k:k-1} \times \mathcal{N}(\mathbf{x}_{k:k-1}|\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1},\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1}) d\mathbf{x}_{k:k-1} + \mathrm{ct}_{/\mathbf{A}}. \tag{43}$$

We now apply equality (69) in Appendix A (with $\mathbf{X} \equiv \mathbf{x}_{k:k-1}$, $\boldsymbol{\mu} \equiv 0$, $\boldsymbol{\Sigma}^{-1} \equiv \widetilde{\mathbf{A}}^\top\mathbf{Q}^{-1}\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{x}} \equiv \boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}$ and $\widetilde{\mathbf{P}} \equiv \boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1}$) to the integral term in (43) (equality (44)(a)) and then the result (76) in Appendix B (equality (44)(b)):

$$\int \mathbf{x}^\top_{k:k-1}\widetilde{\mathbf{A}}^\top\mathbf{Q}^{-1}\widetilde{\mathbf{A}}\mathbf{x}_{k:k-1}\mathcal{N}(\mathbf{x}_{k:k-1}|\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1},\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1})d\mathbf{x}_{k:k-1}$$
$$\overset{(a)}{=} \mathrm{tr}\left(\widetilde{\mathbf{A}}^\top\mathbf{Q}^{-1}\widetilde{\mathbf{A}}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1}+\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1})^\top)\right),$$
$$\overset{(b)}{=} \mathrm{tr}\Big(\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_k+\boldsymbol{\mu}^{\mathrm{s}}_k(\boldsymbol{\mu}^{\mathrm{s}}_k)^\top - \mathbf{A}(\mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_k+\boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_k)^\top)$$
$$-(\boldsymbol{\Sigma}^{\mathrm{s}}_k\mathbf{G}^\top_{k-1}+\boldsymbol{\mu}^{\mathrm{s}}_k(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top + \mathbf{A}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k-1}+\boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top\Big)\Big). \tag{44}$$

Finally, summing (44) for k from 1 to K, using the additivity property of the trace, and plugging the result into (43), yields

$$q(\mathbf{A};\mathbf{A}^{(i)}) = -\frac{1}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Psi}-\boldsymbol{\Delta}\mathbf{A}-\mathbf{A}\boldsymbol{\Delta}^\top+\mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right)+\mathrm{ct}_{/\mathbf{A}}, \tag{45}$$

with

$$\boldsymbol{\Psi} = \sum_{k=1}^{K}\left(\boldsymbol{\Sigma}^{\mathrm{s}}_k+\boldsymbol{\mu}^{\mathrm{s}}_k(\boldsymbol{\mu}^{\mathrm{s}}_k)^\top\right), \tag{46}$$

$$\boldsymbol{\Delta} = \sum_{k=1}^{K}\left(\boldsymbol{\Sigma}^{\mathrm{s}}_k\mathbf{G}^\top_{k-1}+\boldsymbol{\mu}^{\mathrm{s}}_k(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top\right), \tag{47}$$

$$\boldsymbol{\Phi} = \sum_{k=1}^{K}\left(\boldsymbol{\Sigma}^{\mathrm{s}}_{k-1}+\boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top\right). \tag{48}$$

The terms $(\boldsymbol{\Psi},\boldsymbol{\Delta},\boldsymbol{\Phi})$ depend, in an implicit manner, of $\mathbf{A}^{(i)}$, that is the value of the transition matrix used when running the E-step (i.e., Kalman/RTS iterates). We omitted this dependency for the sake of readability.

Then, using (23), we deduce that (31), where function q given in (45), majorizes the MAP loss function $\mathcal{L}_K$ in Eq. (28) for every $\mathbf{A} \in \mathbb{R}^{N_x\times N_x}$. As a special case, when no prior is included and the noise covariance and observation matrices do not vary with k, we retrieve the result [3, Theo.12.4].

C. Computation in the M-step

The M-step at iteration $i \in \mathbb{N}$ amounts to minimizing function $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i)})$ given in (31). Following the computations of the E-step, and particularly the result in (45), we can express this function in a generic form:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x}) \quad \mathcal{Q}(\mathbf{A};\mathbf{A}^{(i)}) = \sum_{m=1}^{M} f_m(\mathbf{A}), \tag{49}$$

where

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})$$
$$f_1(\mathbf{A}) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Psi}-\boldsymbol{\Delta}\mathbf{A}^\top-\mathbf{A}\boldsymbol{\Delta}^\top+\mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right), \tag{50}$$

and $\sum_{m=2}^{M} f_m(\mathbf{A}) = \mathcal{L}_0(\mathbf{A})$ is the regularization term. We recall that, for every $\mathbf{A} \in \mathbb{R}^{N_x\times N_x}$, $f_1(\mathbf{A}) = -q(\mathbf{A},\mathbf{A}^{(i)}) + \mathrm{ct}_{/\mathbf{A}}$ (i.e., up to a constant independent from $\mathbf{A}$) while $\mathcal{L}_0(\mathbf{A}) = -\log(p(\mathbf{A}))$. The assumed sum structure for $\mathcal{L}_0$ allows us to account for factorizing priors.

Function f_1 in (50) is quadratic and convex on $\mathbb{R}^{N_x\times N_x}$. We furthermore assume that each $\{f_m\}_{m=2}^{M}$ involved in the regularization term is proper (i.e., with non empty domain), convex, and lower semi-continuous on $\mathbb{R}^{N_x\times N_x}$, and such that the set of minimizers of (49) is non-empty. Function (49) consequently reads as a sum of a quadratic function, and convex possibly non smooth terms. This paves the way for the application of primal-dual proximal approach for its minimization. Primal-dual proximal splitting (PDPS) algorithms [50] rely on the fundamental tool called the proximity operator,

whose definition is stated as follows. For a proper, lower semi-continuous and convex function $f : \mathbb{R}^{N_x \times N_x} \mapsto (-\infty, +\infty]$, the proximity operator[2] of f at $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ is defined as [51]

$$\text{prox}_f(\widetilde{\mathbf{A}}) = \text{argmin}_{\mathbf{A}} \left(f(\mathbf{A}) + \frac{1}{2}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 \right). \quad (51)$$

Given this tool, a generic PDPS method can iteratively minimize (49) by processing sequentially the terms $\{f_m\}_{m=1}^M$, either through their gradient or their proximity operator. The convergence of the sequence to a minimizer of (49) is then guaranteed, under specific rules on the algorithm hyperparameters (e.g., the stepsize). A large number of algorithms can be built from this generic strategy, with different practical efficiency, depending on several factors such as the order of the updates, the way to process linear operators, the stepsize rules, the use or not of randomized block updates, etc. [52], [53], [54].

On the one hand, following the comparative analysis from [55], [56], we will prefer an algorithm that activates each terms via their proximity operator. Function f_1 is quadratic and with a close form for its proximity operator. Indeed, for every $\vartheta > 0$, for every $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$,

$$\text{prox}_{\vartheta f_1}(\mathbf{A}) = \text{lyapunov}\left(\vartheta \mathbf{Q}^{-1}, \mathbf{\Phi}^{-1}, \mathbf{A}\mathbf{\Phi}^{-1} + \vartheta \mathbf{Q}^{-1}\mathbf{\Delta}\mathbf{\Phi}^{-1}\right), \quad (52)$$

where $A = \text{lyapunov}(X, Y, Z)$ provides the solution to the Lyapunov equation $XA + AY = Z$ [51]. If $\mathbf{Q} = \sigma_{\mathbf{Q}}^2 \mathbf{Id}_{\mathrm{N_x}}$ for some $\sigma_Q > 0$, then (52) simplifies into

$$\text{prox}_{\vartheta f_1}(\mathbf{A}) = \left(\frac{\vartheta}{\sigma_{\mathbf{Q}}^2}\mathbf{\Delta} + \mathbf{A} \right) \left(\frac{\vartheta}{\sigma_{\mathbf{Q}}^2}\mathbf{\Phi} + \mathbf{Id}_{\mathrm{N_x}} \right)^{-1}. \quad (53)$$

On the other hand, it might be beneficial to impose the M-step update to satisfy certain structural properties, such as sparsity, regardless the precision level of its implementation. For the two aforementioned reasons, we opt for the monotone+skew (MS) algorithm from [57], described in Algorithm 4.. More precisely, we assume without loss of generality that f_M is our sparsity-enhancing term. We then propose an implementation of the approach of [57] where we particularize f_M while all the remaining terms $\{f_m\}_{1 \leq m \leq M-1}$ are processed in a consensus-based manner [58], [50]. In this way, the output of Alg. 4 inherits the structure of the proximity operator of f_M. For instance, if f_M is the ℓ_1 norm then the output of Alg. 4 is sparse by construction [59], [60], whatever the value of the precision parameter ξ. Algorithm 4 has two other parameters besides the precision level, namely the stepsizes (γ, λ) whose choice is dictated by the convergence analysis. Under the range settings of Alg. 4, we can establish the following Proposition 1.

Proposition 1. *Assume that, for every $m \in \{1, \ldots, M\}$, function f_m is convex, proper, and lower semicontinuous on $\mathbb{R}^{N_x \times N_x}$. Then, the sequences $\{\mathbf{A}_n^M\}_{n \in \mathbb{N}}$ and $\{\mathbf{Z}_n^M\}_{n \in \mathbb{N}}$ converge to a minimizer of* (49)*.*

Proof. The proof relies on applying the consensus-based splitting from [50, Sec. III] to $\sum_{m=1}^{M-1} f_m$. Let us introduce $\mathbf{L} = [\mathbf{Id}_{N_x}, \ldots, \mathbf{Id}_{N_x}]^\top \in \mathbb{R}^{(M-1)N_x \times N_x}$ and $g : \mathbb{R}^{(M-1)N_x \times N_x} \to (-\infty, +\infty]$ such that, for every $\mathbf{V} = [\mathbf{V}_1^\top, \ldots, \mathbf{V}_{M-1}^\top]^\top \in \mathbb{R}^{(M-1)N_x \times N_x}$, $g(\mathbf{V}) = \sum_{m=1}^{M-1} f_m(\mathbf{V}_m)$. Then, minimizing (49) is equivalent to minimize

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad g(\mathbf{L}\mathbf{A}) + f_M(\mathbf{A}). \quad (54)$$

By construction, $\|\mathbf{L}\|_2 = M - 1$. Moreover, for every $\mathbf{V} = [\mathbf{V}_1^\top, \ldots, \mathbf{V}_{M-1}^\top]^\top \in \mathbb{R}^{(M-1)N_x \times N_x}$, $\mathbf{L}^\top \mathbf{V} = \sum_{m=1}^{M-1} \mathbf{V}_m$ and $\text{prox}_g(\mathbf{V}) = [\text{prox}_{f_1}(\mathbf{V}_1)^\top, \ldots, \text{prox}_{g_{M-1}}(\mathbf{V}_{M-1})^\top]^\top$. Then, the proposed Alg. 4 identifies with [57, Eq. (4.8)] and applying [57, Prop. 4.2] concludes the proof. □

Algorithm 4. *MS algorithm for GraphEM M-step*

Inputs. $\mathbf{A}^{(i)}, \mathbf{\Psi}, \mathbf{\Delta}, \Phi, \mathbf{Q}$*, and prior* $p(\mathbf{A})$*. Precision* $\xi > 0$*.*

1) ***Setting.*** *Set stepsizes* $\lambda \in (0, 1/M)$*,* $\gamma \in [\lambda, (1-\lambda)/(M-1)]$*.*
2) ***Initialization.*** *For every* $m \in \{1, \ldots, M\}$*,* $\mathbf{V}_0^m = \mathbf{A}^{(i)}$*.*
3) ***Recursive step.*** *For* $n = 1, 2, \ldots$*:*

$$\begin{array}{l} \mathbf{W}_n^m = \mathbf{V}_n^m + \gamma \mathbf{V}_n^M \, (\forall m \in \{1, \ldots, M-1\}) \\ \mathbf{W}_n^M = \mathbf{V}_n^M - \gamma \sum_{m=1}^{M-1} \mathbf{V}_n^m \\ \mathbf{A}_n^m = \mathbf{W}_n^m - \gamma \, \text{prox}_{f_m/\gamma}(\mathbf{W}_n^m) \, (\forall m \in \{1, \ldots, M-1\}) \\ \mathbf{A}_n^M = \text{prox}_{\gamma f_M}(\mathbf{W}_n^M) \\ \mathbf{Z}_n^m = \mathbf{A}_n^m + \gamma \mathbf{A}_n^M \, (\forall m \in \{1, \ldots, M-1\}) \\ \mathbf{Z}_n^M = \mathbf{A}_n^M - \gamma \sum_{m=1}^{M-1} \mathbf{A}_n^m \\ \mathbf{V}_{n+1}^m = \mathbf{V}_n^m - \mathbf{W}_n^m + \mathbf{Z}_n^m \, (\forall m \in \{1, \ldots, M\}). \end{array} \quad (56)$$

If $|\mathcal{Q}(\mathbf{A}_n^M, \mathbf{A}^{(i)}) - \mathcal{Q}(\mathbf{A}_{n-1}^M, \mathbf{A}^{(i)})| \leq \xi$*,* ***stop the recursion.***

Output. *Transition matrix update,* $\mathbf{A}^{(i+1)} = \mathbf{A}_n^M$*.*

Typical choices for setting the hyper-parameters in Alg. 4, satisfying the range assumptions and adopted in our experiments, are

$$\lambda = \frac{0.9}{M}, \quad \gamma = \frac{1 - \lambda}{M - 1}. \quad (55)$$

In practice, the algorithm is stopped as soon as (49) stabilizes. Note that a warm start initialization strategy is employed in Alg. 4. The so-called dual variables $\{\mathbf{V}_0^m\}_{m=1}^M$ are set to the previous estimate of the state matrix, that is $\mathbf{A}^{(i)}$. This was observed to yield considerable reduction of required iterations to reach our stopping criterion, when compared to a cold start (setting initial dual variables to zero, for instance). Our implementation of MS processes separately function f_M, and the other terms $(f_m)_{1 \leq m \leq M-1}$. In particular, the elements of the converging sequence $\{\mathbf{A}_n^M\}_{n \in \mathbb{N}}$ of Alg. 4 are outputs of proximity operator for f_M, and thus are sparse for suitable choice of this regularization term. This feature is not present in most standard implementations of primal-dual proximal splitting techniques.

D. Choice of the prior

Let us now explicit choices for the prior $p(\mathbf{A})$, and thus for the regularization function $\mathcal{L}_0$, that are encompassed by our study. We will focus on a hybrid form for the regularization function, such that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \mathcal{L}_0(\mathbf{A}) = \sum_{m=2}^{M} f_m(\mathbf{A}). \quad (57)$$

[2] See also http://proximity-operator.net/

For the sake of readability, we denote f a possible regularization function, keeping in mind that $\mathcal{L}_0$ might combine various terms, to promote various properties in the state matrix $\mathbf{A}$. All these terms would be then processed in a parallel manner in Alg. 4, through their proximity operator.

Let us first discuss a particularly useful choice for f covered by our study. An important matter is to make sure that the LG-SSM resulting from the parameter identification phase (here, the EM procedure) presents good structural properties. In particular, one may require that the first order auto-regressive model inherent to the state process in LG-SSM is stable, in order to avoid any numerical divergence for large values of K. The stability is directly related to the spectral properties of matrix $\mathbf{A}$ in (1). As a result, a sufficient condition for the LG-SSM to be stable (i.e., not diverging with $K \to \infty$) is to be parameterized by an $\mathbf{A}$ parameter with singular values less than one [61], [62]. This condition can be incorporated within our framework, by defining

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad f(\mathbf{A}) = \begin{cases} 0 & \text{if } \mathbf{A} \in \mathcal{S} \\ +\infty & \text{elsewhere} \end{cases} \tag{58}$$

$$\triangleq \iota_{\mathcal{S}}(\mathbf{A}). \tag{59}$$

Hereabove, $\mathcal{S} \subset \mathbb{R}^{N_x \times N_x}$ is related to the stability condition on the SSM:

$$\mathcal{S} = \{\mathbf{A} \in \mathbb{R}^{N_x \times N_x} | \|\mathbf{A}\|_2 \leq \delta < 1\}, \tag{60}$$

for $\delta \in (0, 1)$ (typically close to one). The proximity operator of (58) is simply the projection onto $\mathcal{S}$. Such projection has a closed form [42], that we explicit in Table II. We also provide in Table II two other meaningful examples for $\mathcal{S}$, along with the expression for the associated projection. In particular, the range constraint can be used to impose the sign of some entries of $\mathbf{A}$, i.e., to impose the arrows direction in the estimated graph (under our interpretation).

We now continue our discussion by presenting another family of possible penalty terms in $\mathcal{L}_0$. For each presented example for f, we provide the expression for $\mathrm{prox}_{\vartheta f}$ where ϑ is a positive scaling parameter. This is the aim of Table I. We focus on two particular choices presented in the table, namely the Laplace and block-Laplace priors. Both choices enhance sparsity of matrix $\mathbf{A}$, with the latter being a generalization of the former. The introduction of sparsity promoting prior is in general desirable when doing parameter identification, and is key under our novel approach. As any regularization, it aims at reducing over-fitting problems that could arise for low values of K and thus increases the generalization capacity of the model. Even more, it promotes matrices $\mathbf{A}$ with few non-zero entries, which highly helps for the interpretability of the resulting SSM. Each non-zero entry can actually be understood as a statistical dependence (correlation in this case), between two state dimensions in two consecutive time steps. One can thus interpret $\mathbf{A}$ as the adjacency matrix of a directed graph (since the entries of $\mathbf{A}$ are signed) mapping the entries of the hidden state vector from time $k-1$ to those of time k. The GraphEM approach proposed in this work aims at recovering this graph, and if possible, promoting an interpretable structure. This is done by incorporating a prior of sparsity on $\mathbf{A}$, thanks to appropriate choice for f. An immediate idea would be to define f as the ℓ_0 norm of $\mathbf{A}$, that counts the number of non-zero entries of the matrix. However, this function is non-convex, non continuous, and it is associated to a improper law $p(\mathbf{A})$, which is undesirable. Instead, one prefers to choose for $p(\mathbf{A})$, the proper, log-concave Laplace distribution, leading to the so-called Lasso regularization [63] reading $f(\mathbf{A}) = \kappa \ell_1(\mathbf{A})$ with $\kappa > 0$ a regularization weight. The larger κ, the stronger sparsity of $\mathbf{A}$, with the extreme case of a null $\mathbf{A}$ for sufficiently large κ. The ℓ_1 norm has been used in numerous works of signal processing and machine learning [64], [65], including graph signal processing [37], [66]. It has a simple closed form proximity operator, namely the soft thresholding operator [60], that we recall in Table I.

In certain scenarios, one might have some prior knowledge about some structured sparsity in $\mathbf{A}$. Otherwise stated, one might want to cancel (or not) some blocks of $\mathbf{A}$ in a simultaneous manner, because the entries of these blocks are connected. For instance, they could correspond to real/imaginary part of the same complex quantity (see example in the experimental section). This paves the way for using a more sophisticated prior, where the Laplace distribution is now promoted on each block of $\mathbf{A}$. More formally, let $B \geq 1$ a divisor of N_x^2, defining the number of blocks. Each $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ can be rewritten equivalently as a set of B vectors $(\mathbf{a}(b))_{1 \leq b \leq B}$ of size N_x^2/B. The block-Laplace prior amounts to computing the ℓ_2 norms of each of these vectors, and then summing the B resulting values, to obtain $f(\mathbf{A})$. This can also be rewritten $f(\mathbf{A}) = \kappa \ell_{2,1}(\mathbf{A})$, by using the mixed norm notation from [67], and introducing the regularization weight $\kappa > 0$. Mixed norms have been widely used in machine learning under various combinations [63]. For our particular choice, the proximity operator remains simple, and is provided in Table I. It is worth noticing that both proximity operators for Laplace and block-Laplace involve a threshold of the entries of their input. We list two other examples of priors in the table, the former being the result of a Gaussian prior distribution, while the latter combines Laplace and Gaussian, and is also known as *Lasso with elastic-net* [68].

E. Convergence result

We now show the convergence of GraphEM as in Algorithm 3. We refer to [42] for definitions of functional analysis.

Theorem 1. *Assume that the MAP loss function* (28) *is coercive on* $\mathbb{R}^{N_x \times N_x}$ *and that the prior term* $\mathcal{L}_0$ *is proper, convex, and lower semicontinuous on* $\mathbb{R}^{N_x \times N_x}$. *We furthermore assume that the relative interior of the domain of* $\mathcal{L}_0$ *contains the level set* $\mathcal{E} = \{\mathbf{A} \in \mathbb{R}^{N_x \times N_x} | \mathcal{L}_K(\mathbf{A}) \leq \mathcal{L}_K(\mathbf{A}^{(0)})\}$. *If the M-step in GraphEM is solved exactly i.e., for every* $i \in \mathbb{N}$,

$$\mathbf{A}^{(i+1)} = \mathit{argmin}_{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}} \mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i)}), \tag{61}$$

with $\mathbf{A}^{(0)} \in \mathbb{R}^{N_x \times N_x}$, *then the following statements hold:*

(i) *The sequence* $\left(\mathcal{L}_K(\mathbf{A}^{(i)})\right)_{i \in \mathbb{N}}$ *is a decreasing sequence converging to a finite limite* $\mathcal{L}^*$.

TABLE I
EXAMPLE OF PRIORS, EXPRESSIONS FOR THE RESULTING REGULARIZATION AND ITS PROXIMITY OPERATOR WITH SCALE PARAMETER $\vartheta > 0$.

Prior	$f(\mathbf{A})$	$\mathrm{prox}_{\vartheta f}(\mathbf{A})$
Laplace	$\|\mathbf{A}\|_1 = \sum_{n=1}^{Nx}\sum_{\ell=1}^{N_x} \|A(n,\ell)\|$	$(\mathrm{sign}(A(n,\ell)) \times \max(0, \|A(n,\ell)\| - \vartheta))_{1\leq n,\ell\leq N_x}$
Block-Laplace	$\|\mathbf{A}\|_{2,1} = \sum_{b=1}^{B} \|\mathbf{a}(b)\|_2$	$\left((1 - \frac{\vartheta}{\max(\|\mathbf{a}(b)\|_2, \vartheta)})\, \mathbf{a}(b)\right)_{1\leq b\leq B}$
Gaussian	$\frac{1}{2}\|\mathbf{A}\|_F^2 = \frac{1}{2}\sum_{n=1}^{Nx}\sum_{\ell=1}^{N_x}(A(n,\ell))^2$	$\left(\frac{A(n,\ell)}{1+\vartheta}\right)_{1\leq n,\ell\leq N_x}$
Laplace + Gaussian	$\|\mathbf{A}\|_1 + \frac{1}{2}\|\mathbf{A}\|_F^2$	$\left(\mathrm{sign}\left(\frac{A(n,\ell)}{1+\vartheta}\right) \times \max\left(0, \left\|\frac{A(n,\ell)}{1+\vartheta}\right\| - \frac{\vartheta}{1+\vartheta}\right)\right)_{1\leq n,\ell\leq N_x}$

TABLE II
EXAMPLE OF CONVEX CONSTRAINED SETS AND ASSOCIATED PROJECTION OPERATORS. $\delta > 0$ AND $a_{\min} \leq a_{\max}$ ARE HYPER-PARAMETERS. WE USE THE SINGULAR VALUE DECOMPOSITION $\mathbf{A} = \mathbf{U}^\top \mathrm{DIAG}(\mathbf{s})\mathbf{V}$.

Constraint	$\mathcal{S}$	$\mathrm{Proj}_{\mathcal{S}}(\mathbf{A})$
Bounded spectrum	$\|\mathbf{A}\|_2 \leq \delta$	$\mathbf{U}^\top \mathrm{Diag}\left((\mathrm{sign}(s_n)\min(\|s_n\|, \delta))_{1\leq n\leq N_x}\right)\mathbf{V}$
Range	$(\forall (n,\ell) \in \{1,\ldots,N_x\}^2)\ A(n,\ell) \in [a_{\min}, a_{\max}]$	$(\min(\max(a_{\min}, A(n,\ell)), a_{\max}))_{1\leq n,\ell\leq N_x}$
Bounded energy	$\|\mathbf{A}\|_F \leq \delta$	$\left((1 - \frac{\delta}{\max(\|\mathbf{A}\|_F, \delta)})\, A(n,\ell)\right)_{1\leq n,\ell\leq N_x}$

(ii) *The sequence of iterates* $\left(\mathbf{A}^{(i)}\right)_{i\in\mathbb{N}}$ *has a cluster point (i.e., one can extracts a converging subsequence).*
(iii) *Let* $\mathbf{A}^*$ *a cluster point (i.e., the limit of a converging subsequence) of* $\left(\mathbf{A}^{(i)}\right)_{i\in\mathbb{N}}$ *. Then,* $\mathcal{L}_K(\mathbf{A}^*) = \mathcal{L}^*$*, and* $\mathbf{A}^*$ *is a critical point of* $\mathcal{L}_K$*, i.e.,* $\nabla\mathcal{L}_{1:K}(\mathbf{A}^*) \in \partial\mathcal{L}_0(\mathbf{A}^*)$.

Proof. Our proof consists in showing that the conditions of [66, Th. 5] are met. First, let us remark that the GraphEM exact formulation (61) is well defined, since for every $\mathbf{A}$, $\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i)})$ is a coercive lower-semicontinuous function. It indeed majorizes $\mathcal{L}_K$ which is coercive by assumption. Moreover, according to [19], the likelihood function $\mathcal{L}_{1:K}$ is continuously differentiable on $\mathbb{R}^{N_x\times N_x}$. In particular, it is continuously differentiable on the level set $\mathcal{E}$. The rest of the proof follows using the same arguments as in [66, Th. 5], and the subdifferential calculation from [42, Corollary 16.48(ii)]. □

First, it should be noted that this result focuses on the exact form of Alg. 3, when the M-step is assumed to be solved exactly. Extending Theorem 1(i)-(ii) to the case of an inexact resolution of the M-step would be straightforward, but it is not the case for Theorem 1(iii). According to our Proposition 1, the sequence produced by our M-step inner solver in Alg. 4 converges to an optimal value. In practice, we did not observe any numerical instabilities of the algorithm as soon as a sufficient precision was imposed in the M-step resolution. Second, let us notice that our assumptions on the penalty term $\mathcal{L}_0$ are compliant with those made in Section III-C and with the examples discussed in Section III-D. The case of a null penalty (i.e., $\mathcal{L}_0 \equiv 0$) is also covered by our assumptions. In such case, Alg. 3 becomes equivalent to the MLEM algorithm from [24]. Up to our knowledge, our convergence result is new even for this simple setting. Finally, we must emphasize that, due to the intricate form of the likelihood function and of the presence of a possibly non-differentiable penalty term, it appeared not possible to apply the standard convergence analysis for EM methods from [29].

IV. NUMERICAL SIMULATIONS

A. Synthetic data

We start our experimental section by illustrating the performance of GraphEM in a controlled scenario involving synthetic data. Time series $\{\mathbf{y}_k, \mathbf{x}_k\}_{k=1}^K$ are simulated using (1)-(2), with settings $K = 10^3$, $\mathbf{Q} = \sigma_{\mathbf{Q}}^2\mathbf{Id}_{N_x}$, $\mathbf{R}_k = \sigma_{\mathbf{R}}^2\mathbf{Id}_{N_y}$ for every $k \in \{1,\ldots,K\}$, $\mathbf{P}_0 = \sigma_{\mathbf{P}}^2\mathbf{Id}_{N_x}$ with $(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$ some predefined values. We consider the scenario where $\mathbf{H}_k = \mathbf{Id}_{N_x}$ for every $k \in \{1,\ldots,K\}$, that is there is a one-to-one correspondence between states and observations, and thus $N_x = N_y$. This choice presents the advantage of avoiding any identifiability issues that may arise from an ill-conditioned observation matrix, and thus to fully focus on the graph inference problem, i.e., the estimation of matrix $\mathbf{A}$. Since we are dealing with synthetic data, the ground truth matrix $\mathbf{A}$ can be predefined. In our experiments, we rely on block-diagonal matrices $\mathbf{A}$, made of J blocks with dimensions $\{B_j\}_{1\leq j\leq J}$, so that $N_y = \sum_{j=1}^J B_j$. The diagonal blocks of $\mathbf{A}$ are randomly set as matrices of auto-regressive processes of order one, AR(1), satisfying the stability assumption (i.e., spectral norm less than one). This procedure leads to the construction of four datasets summarized in Table IV. Having the groundtruth available allows us to rely on quality assessment metrics of the estimated $\mathbf{A}$. Here, we retain the relative mean square error (RMSE) in the estimation of the transition matrix, as well as the precision, recall, specificity, accuracy, and F1 score for detecting the non-zero entries of $\mathbf{A}$ (that is, the graph edges positions). A threshold value of 10^{-10} on the absolute entries of matrix $\mathbf{A}$ is used for the detection hypothesis.

For each dataset, we ran GraphEM algorithm using a stable AR(1) matrix as initial estimate, and precision parameters

TABLE III
RESULTS FOR GRAPHEM, STABLEEM, ORACLEEM, MLEM, PGC AND CGC, ALONG WITH AVERAGED COMPUTING TIMES.

	method	RMSE	accur.	prec.	recall	spec.	F1	Time (s.)
A	GraphEM	**0.081789**	**0.90988**	0.999	0.73037	0.99963	**0.84361**	2.3063
	StableEM	0.1405	0.3333	0.3333	**1**	0	0.5	2.3506
	MLEM	0.148	0.3333	0.3333	**1**	0	0.5	1.6059
	PGC	-	0.8765	0.9474	0.6667	0.9815	0.7826	0.1312
	CGC	-	0.8765	**1**	0.6293	**1**	0.7727	0.1366
	OracleEM	0.0879	1	1	1	1	1	0.9572
B	GraphEM	**0.080687**	**0.90691**	**1**	0.72074	**1**	**0.83753**	2.1448
	StableEM	0.15042	0.3333	0.3333	**1**	0	0.5	3.0263
	MLEM	0.15203	0.3333	0.3333	**1**	0	0.5	1.5291
	PGC	-	0.8889	**1**	0.6667	**1**	0.8	0.0606
	CGC	-	0.8889	**1**	0.6667	**1**	0.8	0.0631
	OracleEM	0.076122	1	1	1	1	1	1.1179
C	GraphEM	**0.12624**	**0.91695**	0.97392	0.70676	0.99298	**0.81878**	5.0027
	StableEM	0.23253	0.2656	0.2656	**1**	0	0.4198	5.6791
	MLEM	0.2448	0.2656	0.2656	**1**	0	0.4198	5.6557
	PGC	-	0.9023	**0.9778**	0.6471	**0.9949**	0.7788	0.4095
	CGC	-	0.8555	0.9697	0.4706	**0.9949**	0.6337	0.4175
	OracleEM	0.1214	1	1	1	1	1	2.5504
D	GraphEM	**0.12347**	**0.91648**	**0.98866**	0.69382	**0.99702**	**0.81514**	2.9988
	StableEM	0.22897	0.2656	0.2656	**1**	0	0.4198	4.9561
	MLEM	0.2416	0.2656	0.2656	**1**	0	0.4198	2.5501
	PGC	-	0.8906	0.9	0.6618	0.9734	0.7627	0.2881
	CGC	-	0.8477	0.9394	0.4559	0.9894	0.6139	0.2948
	OracleEM	0.11925	1	1	1	1	1	2.2367

$(\varepsilon, \xi) = (10^{-3}, 10^{-4})$. The regularization $\mathcal{L}_0 = f_2 + f_3$ with $f_2 = \iota_{\mathcal{S}}$ indicator function of the stable matrix set (60) with $\delta = 0.99$, and $f_3 = \kappa\ell_1$ with weight parameter $\kappa > 0$. Such choice satisfies the required assumptions for the convergence of the MS algorithm for the M-step. Parameter κ is set empirically through a rough grid search maximizing the accuracy score. As for comparison, we also provide the results obtained when (i) no regularization is employed, thus leading to the ML estimator (MLEM), (ii) MLEM is modified so as to account for an oracle knowledge of the position of zero entries of $\mathbf{A}$ (OracleEM), and (iii) only stability constraint is imposed (StableEM). In each of these cases, a similar EM-based procedure than GraphEM is used, with simplified computations for the M-step. The results from OracleEM are separated from the others, as it requires the ground truth knowledge of the graph support, not available in practical situations. In addition to these EM-based methods, we provide comparisons with two Granger-causality approaches [69] for graphical modeling, namely pairwise Granger Causality (PGC) and conditional Granger Causality (CGC). Both methods provide a binary information about the identification (or not) of an edge in the graph, by relying on conditional dependency analysis. PGC explores the $N_x(N_x-1)$ possible dependencies among two nodes, at each time independently from the rest. CGC additionally accounts, for each pair of nodes, for the information of the other $N_x - 2$ signals, in order to evaluate whether one node brings information to the other while the rest of signals are observed. As PGC and CGC do not provide a weighted graph estimation, no RMSE score is computed in those case.

The results, averaged on 50 realizations, are presented in Table III. Nor MLEM neither StableEM promote sparsity in the graph which explains their poor results in terms of edge detection. Still, StableEM presents a slightly better RMSE score,

TABLE IV
DESCRIPTION OF DATASETS

Dataset	$(B_j)_{1\leq j\leq J}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	$(3,3,3)$	$(1, 1, 10^{-4})$
C	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	$(3,5,5,3)$	$(1, 1, 10^{-4})$

showing the advantage of integrating the stability constraint as a prior during the estimation procedure. GraphEM provides very good RMSE score on all examples. It is remarkable that these scores are comparable, and sometimes even better, than those obtained with OracleEM. This shows that our construction for the regularization function, gathering both sparsity and stability terms, is well suited to reach a high quality estimate for the state matrix. Moreover, the retained ℓ_1 penalty does not appear here to yield any bias in the estimated graph weights, as it can be sometimes noticed in Lasso regression [64]. This can be probably explained by the proposed combination of an ℓ_1 term and the stability spectral constraint. Regarding the graph structure, we can observe that GraphEM has also better detection scores, when compared with both PGC and CGC. We observe that GraphEM is consistently superior in accuracy and F1. These metrics are relevant since they take into account both the true positive and negative connections. Both StableEM and MLEM present a recall metric equal to one, which corresponds to an estimate of the transition matrix without any null entries. An opposite effect is observed by PGC and CGC in specificity, since both methods can overestimate the amount of zeros (no connections), particularly the latter [34]. We remind that OracleEM should not be compared within this metric, since it has access to the edges position and thus has perfect edge detection scores. We also provide in the

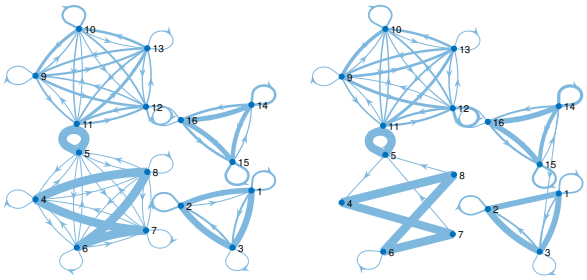

Fig. 1. True graph (left) and GraphEM estimate (right) for dataset C.

last column of Table III, the averaged computing times over 50 realizations for each methods, for Matlab 2021a codes running on a 11th Gen Intel(R) Core(TM) i7-1185G7 3.00GHz with 32 Go RAM. PGC and CGC require the lowest computing times. These two methods are based on simpler auto-regressive processes (without latent sates), which explains their poorer performance w.r.t. GraphEM. The other methods share computing times with similar order of magnitude. OracleEM is the fastest method among the EM-based ones, simply because it works in the favorable setting when the edge positions of the graph are assumed to be known, thus reducing the size of the search space. GraphEM is very competitive, compared to its non-regularized counterpart MLEM, thanks to the proposed efficient proximal splitting M-step resolution, while reaching better quantitative results than MLEM by far. Interestingly, for a given dataset size, one can notice a trend of a lower computational times when solving the inference problem for an higher noise level (see dataset A vs B, dataset C vs D), whatever the algorithm employed. This might be related to the peaky likelihood phenomenon described in [70], namely the larger noise variance, the easier it is to explore the posterior.

We also display an example of graph reconstruction for dataset C in Fig. 1, illustrating the ability of GraphEM to recover the graph structure and signed weights. Finally, we show on Fig. 2 a comparison between MLEM and GraphEM, in terms of evolution of the loss function (28) and the RMSE score, along the iterations of both algorithms. One can notice that both methods reach convergence very fast, in about a dozen of iterations. As EM-based approaches, they both guarantee the decrease of the loss function. Here, we should precise that slight oscillations might be observed for GraphEM as it solves a constrained minimization problem. The projection steps might break the monotonicity of the loss decrease, but this is not jeopardizing the convergence properties of the EM approach, and in practice the loss is rather stable. We notice the different behavior of the RMSE curves for both methods. The non regularized MLEM shows the typical noise amplification effect, decreasing first the RMSE and then increasing it. In contrast, the introduction of a suitable regularization strategy in GraphEM makes it avoid such undesirable phenomenon, and the RMSE evolution follows a stable decrease until converging to its final small value.

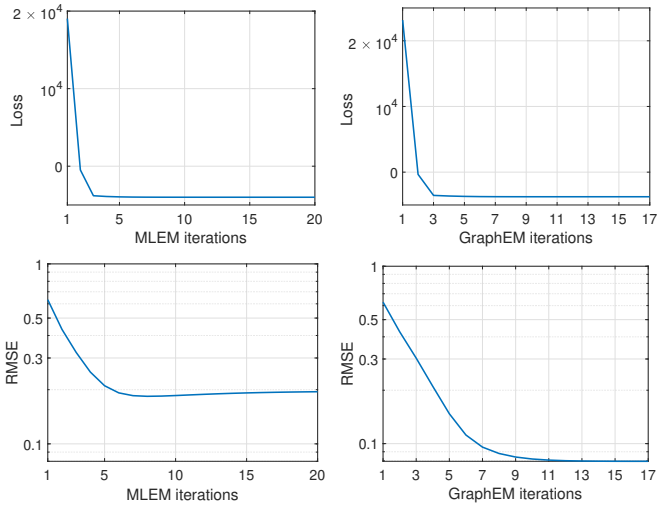

Fig. 2. Evolution of loss function (top) and RMSE score (bottom), for MLEM (left) and graphEM (right), when ran on a realization from dataset A.

B. Wireless channel tracking

We consider a multi-input multi-output (MIMO) wireless communication system with fading [71], where the (unknown) channel between the transmitter (TX) and the receiver (RX) must be tracked. The MIMO system is $L \times L$, although a different number of transmit and receive antennas is readily possible. At each time step, N_{pil} L-dimensional complex pilots, $\mathbf{p}_k^{(i)}$, $i = 1, \ldots, N_{\text{pil}}$, with $N_{\text{pil}} \geq 1$ and 64-QAM symbols in each component, are transmitted between TX and RX through the complex channel $\mathbf{C}_k \in \mathbb{C}^{L \times L}$. Therefore, the MIMO system with fading is modeled as

$$\mathbf{z}_k^{(i)} = \mathbf{C}_k \mathbf{p}_k^{(i)} + \mathbf{n}_k^{(i)}, \tag{62}$$

with $k = 1, \ldots, K$ and $i = 1, \ldots, N_{\text{pil}}$, where $\mathbf{n}_k^{(i)} \in \mathbb{C}^L$, is distributed complex-normally with isotropic covariance that yields an $E_b/N_0 = 38$dBs. In order to express (62) as the (real-valued) observation model in (2), we define $\mathbf{x}_k = [\text{Real}(\text{vec}(\mathbf{C}_k)); \text{Imag}(\text{vec}(\mathbf{C}_k))] \in \mathbb{R}^{2L^2}$ (i.e., $N_x = 2L^2$) as the vectorized version of the complex channel. The real-valued observation vector corresponding to each pilot $i \in \{1, \ldots, N_{\text{pil}}\}$ is defined as $\mathbf{y}_k^{(i)} = [\text{Real}(\mathbf{z}_k); \text{Imag}(\mathbf{z}_k)]$, so that $\mathbf{y}_k = [\mathbf{y}_k^{(1)}, \ldots, \mathbf{y}_k^{(N_{\text{pil}})}]^\top \in \mathbb{R}^{N_y}$ with $N_y = 2LN_{\text{pil}}$. The observation matrix $\mathbf{H}_k \in \mathbb{R}^{2LN_{\text{pil}} \times 2L^2}$ is a sparse matrix constructed from the real and imaginary components of all pilots, in such a way that the real-valued observation model in (2) is equivalent to (62). The prior pdf of each entry of $\mathbf{x}_0$ is a standard normal distribution. We consider isotropic covariances $\mathbf{Q} = \sigma_{\mathbf{Q}}^2 \mathbf{Id}_{N_x}$ and $\mathbf{R}_k = \sigma_{\mathbf{R}}^2 \mathbf{Id}_{N_y}$, for every $k \in \{1, \ldots, K\}$, with $\sigma_{\mathbf{Q}} = 0.2$ and $\sigma_{\mathbf{R}} = 0.2$. We transmit $N_{\text{pil}} = 4$ at each time step, with $L = 4$ transmit and receive antennas, hence $N_x = 32$ and $N_y = 32$.

The goal is estimating $\mathbf{A} \in \mathbb{R}^{32 \times 32}$ by introducing sparse constraints motivated by the physical model, in such a way we can then do tracking of the channel with the estimated transition matrix. We consider two datasets, obtained from two ground truth matrices $\mathbf{A}$. Dataset E relies on the tri-diagonal

transition matrix:

$$(\forall (i,j) \in \{1,\dots,32\}^2) \quad A(i,j) = \begin{cases} a & \text{if} \quad i=j \text{ or } i=j+16 \text{ or } i+16=j, \\ 0 & \text{otherwise,} \end{cases} \tag{63}$$

with $a = 0.495$ set so that $\mathbf{A}$ belongs to the stability set (60) with $\delta = 0.99$. Dataset F uses the ground truth matrix $\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{B} \\ \mathbf{B} & \mathbf{B} \end{bmatrix}$ with $\mathbf{B} \in \mathbb{R}^{16\times 16}$ a block diagonal matrix of 3 blocks with respective dimensions $(4, 8, 4)$. As in the example in subsection IV-A, randomly selected AR(1) matrices belonging to the stability set $\mathcal{S}$, are used to build the blocks of $\mathbf{B}$.

In both cases, observed data are simulated using (2) with $K = 200$ time steps. We compare the MLEM approach with the GraphEM algorithm, for the estimation of $\mathbf{A}$ from these data. In this example, we aim at exploring the robustness w.r.t. the regularization parameter in GraphEM when imposing both a stability and sparsity constraint, namely $\mathcal{L}_0 = f_2 + f_3$ with f_2 set as in our previous example, and $f_3 = \kappa \ell_{2,1}$ with weight parameter $\kappa > 0$. The $\ell_{2,1}$ norm, as introduced in subsection III-D, is a block-sparsity enhancing penalty. We preferred it to the ℓ_1 norm in that example, as it better accounts for correlations between entries of $\mathbf{A}$ related to the same states in the complex domain. More precisely, following our notations from subsection III-D, we set $B = L^4$ blocks, so that, for every $b \in \{1,\dots,B\}$, and every $\mathbf{A} \in \mathbb{R}^{2L^2\times 2L^2}$, we consider the b-th block of it as

$$\mathbf{a}(b) = \big[A(i,j), A(i+L^2,j), A(i,j+L^2), A(i+L^2,j+L^2)\big]^\top \in \mathbb{R}^4,$$

with $(i,j) \in \{1,\dots,L^2\}^2$ the index pair corresponding to the matrix position associated with the lexicographic index b.

To that end, $\ell_{2,1}(\mathbf{A})$ pairs the real and imaginary parts of the state at current and previous time state. Similar block-sparsity prior was used in [72], [73] for processing complex-valued images. On both datasets, we run GraphEM algorithm with various weights κ selected with the range $(0, 400]$, i.e., in a significantly wide range. We show two performance metrics to evidence the robustness and successful performance of GraphEM. First, in Fig. 3 (top), we show the relative mean square error (RMSE) in the estimation of the matrix $\mathbf{A}$ with respect to κ, either for dataset E (left) and dataset F (right). We then design a more sophisticated BER analysis where we will track the channel and perform linear detection. Therefore, instead of plugging the true $\mathbf{A}$ to track the channel, we set the matrix estimates corresponding to MLEM or GraphEM, with different values of κ. More precisely, at each time step, we track the channel $\mathbf{C}_k$ in the same model described above. The difference is that now we run the Kalman filter setting the estimated $\mathbf{A}$ from each corresponding algorithm. For each channel use, we transmit $N_{\text{pil}} = 4$ pilots (for tracking purposes) and 500 (unknown) symbols for evaluating the BER performance under the estimated $\mathbf{A}$ matrix of each algorithm. We decode the transmitted symbols by using the MMSE detector [74]. We run this testing phase for 10^4 time steps, for ensuring a sufficiently averaged BER metric [75]. The BER, as a function of the parameter κ, is shown in Fig. 3 (bottom) for both MLEM and GraphEM approaches running on both dataset E (left) and dataset F (right). In all the four plots, we can see that GraphEM outperforms the MLEM approach, obtaining the best performance for a value around $\kappa = 100$. We can also see, that the performance is good for a wide range of κ values with an asymmetric behavior: larger values of κ still retain the advantage of using GraphEM in this example. This shows the stability of GraphEM model to the setting of κ.

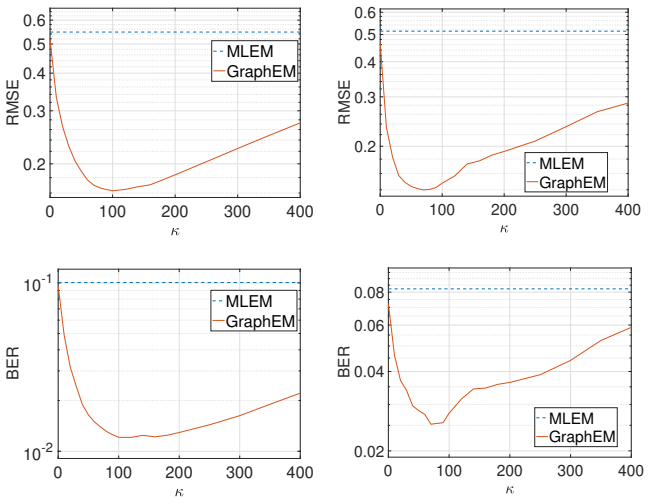

Fig. 3. **Wireless channel tracking.** Performance of GraphEM algorithm in terms of (top) the relative mean square error (RMSE) in the estimation of $\mathbf{A}$ and (bottom) bit error rate (BER) as function of κ, the weight associated to the $\ell_{2,1}$ norm, on dataset E (left) and F (right).

V. Conclusion

In this paper, we have proposed a novel methodological framework, called GraphEM, to estimate parameters in the linear-Gaussian state-space model (LG-SSM) by introducing available application-dependent prior knowledge. While the methodology is generic to allow for the MAP estimate of all LG-SSM model parameters, we develop further our method for the estimation of the transition matrix. Our novel approach interprets this matrix as the adjacency matrix of a directed graph, encoding the Markovian dependencies in the evolution of the multi-variate state. This interpretation has some ties with Granger causality. We then propose GraphEM for estimating this matrix jointly with the inference of the sequence of hidden states. GraphEM is a convergent expectation-maximization (EM) methodology which incorporates a novel consensus-based implementation of a primal-dual proximal convex optimization solver for the M-step, enabling an efficient incorporation of sophisticated priors on the graph. Numerical results illustrate the great performance of the method. The novel interpretation, the solid theoretical guarantees, and the good performance of GraphEM pave the way for novel advances. For example, we have considered in our numerical examples several penalties on the graph structure, such as stability of the hidden process and block-sparsity enhancing priors that allow for simple and interpretable graphs. The versatility of our method allows to introduce other priors to target specific applications.

APPENDIX A
USEFUL EXPECTATIONS INVOLVING NORMAL PDFS

Let us consider a random vector $\mathbf{X}$ in dimension $n \geq 1$, following a multivariate Gaussian distribution with mean $\widetilde{\mathbf{x}}$ and covariance matrix $\widetilde{\mathbf{P}}$:

$$\mathbf{X} \sim \mathcal{N}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}), \tag{64}$$

with $\widetilde{\mathbf{x}} \in \mathbb{R}^n$, $\widetilde{\mathbf{P}} \in \mathbb{R}^{n\times n}$ symmetric positive definite. We are interested in computing

$$\mathbb{E}\{(\mathbf{X}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})\}, \tag{65}$$

for some $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ symmetric positive definite. We have

$$\begin{aligned}
&\mathbb{E}\{(\mathbf{X}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})\} \\
&\quad = \mathbb{E}\{(\mathbf{X}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\widetilde{\mathbf{x}})\} + (\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\widetilde{\mathbf{x}}) && (66)\\
&\quad = \mathrm{tr}\left(\mathbb{E}\{(\mathbf{X}-\widetilde{\mathbf{x}})(\mathbf{X}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}\}\right) + (\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\widetilde{\mathbf{x}}) && (67)\\
&\quad = \mathrm{tr}\left(\mathbb{E}\{(\mathbf{X}-\widetilde{\mathbf{x}})(\mathbf{X}-\widetilde{\mathbf{x}})^\top\}\boldsymbol{\Sigma}^{-1}\right) + (\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\widetilde{\mathbf{x}}) && (68)\\
&\quad = \mathrm{tr}(\widetilde{\mathbf{P}}\boldsymbol{\Sigma}^{-1}) + \mathrm{tr}((\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\widetilde{\mathbf{x}})).
\end{aligned}$$

Finally,

$$\begin{aligned}
&\mathbb{E}\{(\mathbf{X}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})\} = \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\widetilde{\mathbf{P}}) + \mathrm{tr}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\widetilde{\mathbf{x}})(\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top)\\
&\quad = \mathrm{tr}(\boldsymbol{\Sigma}^{-1}(\widetilde{\mathbf{P}} + (\boldsymbol{\mu}-\widetilde{\mathbf{x}})(\boldsymbol{\mu}-\widetilde{\mathbf{x}})^\top)). && (69)
\end{aligned}$$

B E-STEP CALCULATIONS

Here, we explicit the end of the calculations needed for $q(\mathbf{A}; \mathbf{A}^{(i)})$. We recall that $\widetilde{\mathbf{A}} = [\mathbf{Id}_{N_x}, -\mathbf{A}]$. Let $k \in \{1, \ldots, K\}$. Then,

$$\widetilde{\mathbf{A}}^\top \mathbf{Q}^{-1} \widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{Q}^{-1} & -\mathbf{Q}^{-1}\mathbf{A} \\ \mathbf{A}^\top\mathbf{Q}^{-1} & \mathbf{A}^\top\mathbf{Q}^{-1}\mathbf{A} \end{bmatrix}. \tag{70}$$

Morever,

$$\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1})^\top = \begin{bmatrix} \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top & \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top \\ \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top & \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top \end{bmatrix}. \tag{71}$$

Thus,

$$\begin{aligned}
&\mathrm{tr}\left(\widetilde{\mathbf{A}}^\top \mathbf{Q}^{-1} \widetilde{\mathbf{A}}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1})^\top)\right) && (72)\\
&= \mathrm{tr}\left(\begin{bmatrix} \mathbf{Q}^{-1} & -\mathbf{Q}^{-1}\mathbf{A} \\ \mathbf{A}^\top\mathbf{Q}^{-1} & \mathbf{A}^\top\mathbf{Q}^{-1}\mathbf{A} \end{bmatrix} \right.\\
&\quad \times \left(\begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{s}}_{k} & \boldsymbol{\Sigma}^{\mathrm{s}}_{k}\mathbf{G}^\top_{k-1} \\ \mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_{k} & \boldsymbol{\Sigma}^{\mathrm{s}}_{k-1} \end{bmatrix}\right.\\
&\quad \left.\left. + \begin{bmatrix} \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top & \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top \\ \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top & \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top \end{bmatrix}\right)\right). && (73)
\end{aligned}$$

In order to limit computations, we can make use of the fact that

$$\mathrm{tr}\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \mathrm{tr}(A) + \mathrm{tr}(D). \tag{74}$$

Using (74) and the additivity of the trace in (73) leads to equality (75)(a). Then, using the permutation property of the trace yields the equality (75)(b):

$$\begin{aligned}
&\mathrm{tr}\left(\widetilde{\mathbf{A}}^\top \mathbf{Q}^{-1} \widetilde{\mathbf{A}}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1})^\top)\right)\\
&\overset{(a)}{=} \mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top)\right)\\
&\quad + \mathrm{tr}\left(-\mathbf{Q}^{-1}\mathbf{A}(\mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top)\right)\\
&\quad + \mathrm{tr}\left(-\mathbf{A}^\top\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k}\mathbf{G}^\top_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\right)\\
&\quad + \mathrm{tr}\left(\mathbf{A}^\top\mathbf{Q}^{-1}\mathbf{A}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\right)\\
&\overset{(b)}{=} \mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top)\right)\\
&\quad + \mathrm{tr}\left(-\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k}\mathbf{G}^\top_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top\right)\\
&\quad + \mathrm{tr}\left(-\mathbf{Q}^{-1}\mathbf{A}(\mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top)\right)\\
&\quad + \mathrm{tr}\left(\mathbf{Q}^{-1}\mathbf{A}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top\right). && (75)
\end{aligned}$$

Finally,

$$\begin{aligned}
&\mathrm{tr}\left(\widetilde{\mathbf{A}}^\top \mathbf{Q}^{-1} \widetilde{\mathbf{A}}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k:k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k:k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k:k-1})^\top)\right)\\
&= \mathrm{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top - (\boldsymbol{\Sigma}^{\mathrm{s}}_{k}\mathbf{G}^\top_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top\right.\\
&\quad \left. -\mathbf{A}(\mathbf{G}_{k-1}\boldsymbol{\Sigma}^{\mathrm{s}}_{k} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k})^\top) + \mathbf{A}(\boldsymbol{\Sigma}^{\mathrm{s}}_{k-1} + \boldsymbol{\mu}^{\mathrm{s}}_{k-1}(\boldsymbol{\mu}^{\mathrm{s}}_{k-1})^\top)\mathbf{A}^\top\right). && (76)
\end{aligned}$$

REFERENCES

[1] J. D. Hamilton, "State-space models," *Handbook of Econometrics*, vol. 4, pp. 3039–3080, 1994.

[2] C.-J. Kim and C. R. Nelson, *State-space models with regime switching: classical and Gibbs-sampling approaches with applications*. MIT Press Books, 1st ed., 1999.

[3] S. Sarkka, *Bayesian Filtering and Smoothing*. 3 ed., 2013.

[4] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[5] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, no. 7567, pp. 47–55, 2015.

[6] B. Choi, M. Bergés, E. Bou-Zeid, and M. Pozzi, "Short-term probabilistic forecasting of meso-scale near-surface urban temperature fields," *Environmental Modelling & Software*, vol. 145, p. 105189, 2021.

[7] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[8] A. Doucet, A. M. Johansen, *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[9] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," *IEE Proceedings-F Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[10] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[11] V. Elvira, L. Martino, M. F. Bugallo, and P. Djurić, "In search for improved auxiliary particle filters," in *Signal Processing Conference (EUSIPCO), 2018 Proceedings of the 26th European*, pp. 1–5, IEEE, 2018.

[12] N. Branchini and V. Elvira, "Optimized auxiliary particle filters: adapting mixture proposals via convex optimization," in *Uncertainty in Artificial Intelligence*, pp. 1289–1299, PMLR, 2021.

[13] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 145–152, 2019.

[14] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[15] G. C. Reinsel, *Elements of multivariate time series analysis*. Springer Science & Business Media, 1997.

[16] T. Li, J. Prieto, and J. M. Corchado, "Fitting for smoothing: A methodology for continuous-time target track estimation," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, 2016.

[17] T. Li, H. Chen, S. Sun, and J. M. Corchado, "Joint smoothing and tracking based on continuous-time target trajectory function fitting," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1476–1483, 2018.

[18] M. Fasiolo, N. Pya, and S. N. Wood, "A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology," *Statistical Science*, pp. 96–118, 2016.

[19] N. Gupta and R. Mehra, "Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 774–783, 1974.

[20] M. Segal and E. Weinstein, "A new method for evaluating the log-likelihood gradient (score) of linear dynamic systems," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 763–766, 1988.

[21] M. Segal and E. Weinstein, "A new method for evaluating the log-likelihood gradient, the hessian, and the fisher information matrix for linear dynamic systems," *IEEE Transactions on Information Theory*, vol. 35, no. 3, pp. 682–687, 1989.

[22] O. Cappe, E. Moulines, and T. Ridden, *Inference in Hidden Markov Models*. Springer Series in Statistics, Springer New York, NY, 1st ed., 2005.

[23] R. Olsson, K. Petersen, and T. Lehn-Schioler, "State-space models: from the EM algorithm to a gradient approach," *Neural Computation*, vol. 19, no. 4, p. 1097–1111, 2007.

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

[25] S. Sharma, A. Majumdar, V. Elvira, and E. Chouzenoux, "Blind Kalman filtering for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 35, pp. 4916–4919, Nov. 2020.

[26] S. Sharma, V. Elvira, E. Chouzenoux, and A. Majumdar, "Recurrent dictionary learning for state-space models with an application in stock forecasting," *Neurocomputing*, vol. 450, pp. 1–13, Aug. 2021.

[27] L. Frenkel and M. Feder, "Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 306–320, 1999.

[28] 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 (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[29] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[30] M. Eichler, "Graphical modelling of multivariate time series," *Probability Theory and Related Fields*, vol. 153, pp. 233–268, Jun. 2012.

[31] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2189–2199, Aug. 2004.

[32] D. Barber and A. T. Cemgil, "Graphical models for time-series," *IEEE Signal Processing Magazine*, vol. 27, pp. 18–28, Nov 2010.

[33] A. Pirayre, C. Couprie, L. Duval, and J.-C. Pesquet, "BRANE Clust: Cluster-assisted gene regulatory network inference refinement," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, pp. 850–860, May 2018.

[34] D. Luengo, G. Rios-Munoz, V. Elvira, C. Sanchez, and A. Artes-Rodriguez, "Hierarchical algorithms for causality retrieval in atrial fibrillation intracavitary electrograms," *IEEE Journal of Biomedical and Health Informatics*, vol. 12, pp. 143–155, Jan. 2019.

[35] C. Ravazzi, R. Tempo, and F. Dabbene, "Learning influence structure in sparse social networks," *IEEE Transactions on Control of Network Systems*, vol. PP, pp. 1–1, 12 2017.

[36] J. Richiardi, S. Achard, B. Horst, , and D. V. D. Ville, "Machine learning with brain graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 58–70, 2013.

[37] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, vol. 9, pp. 432–441, jul 2008.

[38] J. T. Chiu, Y. Deng, and A. M. Rush, "Low-rank constraints for fast inference in structured models," in *Advances in Neural Information Processing Systems* (A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), 2021.

[39] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[40] M. Eichler, "Causal inference with multiple time series: principles and problems," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1997, p. 20110613, 2013.

[41] E. Chouzenoux and V. Elvira, "GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints," in *In Proceedings of the 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020)*, pp. 5840–5844, 4–8 May 2020.

[42] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

[43] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[44] M. Briers, A. Doucet, and S. Maskell, "Smooting algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 61, 2010.

[45] B. Thiesson, D. M. Chickering, D. Heckerman, and C. Meek, "ARMA time-series modeling with graphical models," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pp. 552–560, 2004.

[46] N. Kantas, A. Doucet, S. Singh, and J. Maciejowski, "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models," in *Proceedings of the IFAC Symposium on System Identification (SYSID 2009)*, (Saint-Malo, France), 6-8 July 2009.

[47] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of monte carlo methods for parameter estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 1, pp. 1–62, 2020.

[48] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, pp. 47–60, Nov. 1996.

[49] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[50] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, 2015.

[51] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, Springer, 2011.

[52] H. Raguet, J. Fadili, and G. Peyré, "A generalized forward-backward splitting," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1199–1226, 2013.

[53] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[54] P. L. Combettes and J.-C. Pesquet, "Fixed point strategies in data science," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3878–3905, 2021.

[55] P. L. Combettes and L. E. Glaudin, "Proximal activation of smooth functions in splitting algorithms for convex image recovery," *SIAM Journal on Imaging Sciences*, vol. 12, no. 4, pp. 1905–1935, 2019.

[56] L. Briceño-Arias and N. Pustelnik, "Proximal or gradient steps for cocoercive operators," tech. rep., 2021. https://arxiv.org/pdf/2101.06152.pdf.

[57] L. M. Briceño-Arias and P. L. Combettes, "A monotone+skew splitting model for composite monotone inclusions in duality," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1230–1250, 2011.

[58] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *Journal of Nonlinear and Convex Analysis*, vol. 16, no. 12, 2015.

[59] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[60] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[61] A. Bird and C. K. Williams, "Customizing sequence generation with multi-task dynamical systems," tech. rep., 2019. https://arxiv.org/abs/1910.05026.

[62] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and its Applications*. Springer Texts Statistics, Springer, Cham, 4th ed., 2007.

[63] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, p. 1–106, Jan. 2012.

[64] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[65] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs, "A variational formulation for frame-based inverse problems," *Inverse Problems*, vol. 23, pp. 1495–1518, June 2007.
[66] A. Benfenati, E. Chouzenoux, and J.-C. Pesquet, "Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation," *Signal Processing*, vol. 169, p. 107417, Apr. 2020.
[67] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.
[68] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[69] S. L. Bressler and A. K. Seth, "Wiener–Granger causality: a well established methodology," *NeuroImage*, vol. 58, no. 2, pp. 323–329, 2011.
[70] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006.
[71] V. Elvira and I. Santamaria, "Multiple importance sampling for symbol error rate estimation of maximum-likelihood detectors in mimo channels," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1200–1212, 2021.
[72] L. El Gueddari, C. G R, E. Chouzenoux, and P. Ciuciu, "Calibration-less multi-coil compressed sensing magnetic resonance image reconstruction based on OSCAR regularization," *MDPI Journal of Imaging, Special Issue on Inverse Problems and Imaging*, vol. 7, no. 58, pp. X–X+20, 2021.
[73] A. Florescu, E. Chouzenoux, J.-C. Pesquet, P. Ciuciu, and S. Ciochina, "A majorize-minimize memory gradient method for complex-valued inverse problems," *Signal Processing*, vol. 103, pp. 285–295, 2014.
[74] Y. Jiang, M. K. Varanasi, and J. Li, "Performance analysis of ZF and MMSE equalizers for MIMO systems: An in-depth study of the high snr regime," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2008–2026, 2011.
[75] V. Elvira and I. Santamaria, "Multiple importance sampling for efficient symbol error rate estimation," *IEEE Signal Processing Letters*, vol. 26, no. 3, pp. 420–424, 2019.