



**HAL**  
open science

# Sparse Graphical Linear Dynamical Systems

Emilie Chouzenoux, Víctor Elvira

► **To cite this version:**

| Emilie Chouzenoux, Víctor Elvira. Sparse Graphical Linear Dynamical Systems. 2023. hal-04358490

**HAL Id: hal-04358490**

**<https://inria.hal.science/hal-04358490>**

Preprint submitted on 21 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Sparse Graphical Linear Dynamical Systems

**Emilie Chouzenoux**

*Center for Visual Computing  
Inria, University Paris Saclay  
91190 Gif-sur-Yvette, France*

EMILIE.CHOUZENOUX@INRIA.FR

**Víctor Elvira**

*School of Mathematics  
University of Edinburgh  
EH9 3FD Edinburgh, UK*

VICTOR.ELVIRA@ED.AC.UK

## Abstract

Time-series datasets are central in numerous fields of science and engineering, such as biomedicine, Earth observation, and network analysis. Extensive research exists on state-space models (SSMs), which are powerful mathematical tools that allow for probabilistic and interpretable learning on time series. Estimating the model parameters in SSMs is arguably one of the most complicated tasks, and the inclusion of prior knowledge is known to both ease the interpretation but also to complicate the inferential tasks. Very recent works have attempted to incorporate a graphical perspective on some of those model parameters, but they present notable limitations that this work addresses. More generally, existing graphical modeling tools are designed to incorporate either static information, focusing on statistical dependencies among independent random variables (e.g., graphical Lasso approach), or dynamic information, emphasizing causal relationships among time series samples (e.g., graphical Granger approaches). However, there are no joint approaches combining static and dynamic graphical modeling within the context of SSMs. This work proposes a novel approach to fill this gap by introducing a joint graphical modeling framework that bridges the static graphical Lasso model and a causal-based graphical approach for the linear-Gaussian SSM. We present DGLASSO (Dynamic Graphical Lasso), a new inference method within this framework that implements an efficient block alternating majorization-minimization algorithm. The algorithm’s convergence is established by departing from modern tools from nonlinear analysis. Experimental validation on synthetic and real weather variability data showcases the effectiveness of the proposed model and inference algorithm. This work will significantly contribute to the understanding and utilization of time-series data in diverse scientific and engineering applications where incorporating a graphical approach is essential to perform the inference.

**Keywords:** State-space models, graph inference, sparsity, graphical lasso, expectation-maximization, proximal algorithm

## 1. Introduction

Time series appear in applications of most fields of science and engineering, ranging from biomedicine, Earth observation, or network analysis, to name a few. The literature of time-series analysis in the last century is vast, mostly dominated by auto-regressive moving-averaging (ARMA) models and their extensions (Hamilton, 2020).

With the surge of the Bayesian paradigm, state-space models (SSMs) have become very popular in the last decades (Hamilton, 1994; Kim and Nelson, 1999; Sarkka, 2013) for

time series modeling. SSMs characterize complex systems through discrete-time models composed of a hidden (or latent) state that evolves in a Markovian manner. SSMs are composed of a state model, which can mimic realistically the complicated dynamics of the system, and the observation model, which links the temporal observations to the hidden state at the same time step. In SSMs, the Bayesian filtering task consists on computing the posterior probability density function (pdf) of the hidden state at a given time step, given all observations from the time series available up to that time. However, in most SSMs the posterior pdf is intractable and must be approximated, generally through particle filters (Djuric et al., 2003; Doucet et al., 2009; Naeseth et al., 2019). One relevant exception is the linear-Gaussian state-space model (LG-SSM), which allows for exact inference, when the model parameters are known, through the Kalman filter and the Rauch-Tung-Striebel (RTS) smoother (Sarkka, 2013, Chapter 8). The LG-SSM is arguably the most popular model and is still subject of intense research (Sarkka, 2013, Chapter 4). The literature of parameter estimation for LG-SSMs considers both probabilistic and point-wise approaches. The former methods can be applied for a wider class of SSMs (i.e., beyond LG-SSMs) and include for instance particle Markov chain Monte Carlo (Andrieu et al., 2010), sequential Monte Carlo squared (Chopin et al., 2013), and nested particle filters (Crisan and Míguez, 2018; Pérez-Vieites and Míguez, 2021) (see (Kantas et al., 2015) for a review).

One of the main challenges in SSMs is the estimation of the model parameters. In its canonical version, the LG-SSM is composed of four parameters. The state model is parametrized by the so-called transition matrix, which linearly links consecutive states, and the covariance matrix of the additive Gaussian state noise. The observation model is parametrized by the observation matrix, which connects (also linearly) the state with the observation at the same time-step, and the covariance matrix of the additive Gaussian observation noise. Point-wise approaches are generally based on maximum likelihood estimation of the model parameters in LG-SSMs (Sarkka, 2013, Chap. 12). A first family of methods implement recursive schemes to compute first and second derivatives of the likelihood with respect to the unknown parameters (Segal and Weinstein, 1988, 1989; Nagakura, 2021; Gupta and Mehra, 1974) (see the discussion in (Cappe et al., 2005, Sec. 10.2.4)). Iterative optimizers such as quasi-Newton (Olsson et al., 2007) or Newton-Raphson (Gupta and Mehra, 1974) are then employed to get a maximum likelihood estimate. However, due to the intricate form of the likelihood function, these optimization solvers must be used in a black-box setting, which prevents from the implementation of efficient linesearch procedures, and limits considerably the convergence guarantees. The second family of point-wise methods is based on expectation-minimization (EM) algorithms (Shumway and Stoffer, 1982)(Cappe et al., 2005, Sec. 10.4)(Sarkka, 2013, Sec. 12.2.3). EM algorithm for parameter estimation in LG-SSM exhibits a good convergence stability (Dempster et al., 1977; Wu, 1983), and a simplicity of implementation, which might explain its wide use on practical applied fields, such as finance, electrical engineering, and radar (Sharma et al., 2020, 2021; Frenkel and Feder, 1999). The benefits and drawbacks of these algorithms are discussed in (Shumway and Stoffer, 1982, Sec. 1).

Another important family of approaches for time series analysis relies on graphical modeling. Static models for representing correlation properties within multivariate times series (assuming the series element as independant samples from a random variable), have been explored for decades (Maathuis et al., 2019). Let us mention the famous Graphical Lasso

algorithm (Friedman et al., 2008), and its multiple variants (Chandrasekaran et al., 2012; Benfenati et al., 2020; Belilovsky et al., 2017; Bach and Jordan, 2004; Ying et al., 2020), for inference of sparse Gaussian graphical models. Modeling (non-independent) dynamical behaviors through (directed) graph representations, and in particular, causality, has also been explored (Eichler, 2012; Ioannidis et al., 2019; Giannakis et al., 2018).

There is actually a tight link between state-space modeling and dynamical graphical modeling, as emphasized in (Barber and Cemgil, 2010). For instance, in Sagi et al. (2023) the extended Kalman filter (EKF) is enhanced by taking a graphical perspective. (Ioannidis et al., 2018) takes an approach of jointly estimating latent processes and the topology of a related graph. (Alippi and Zambon, 2023) also adopts a graphical perspective in linear-Gaussian models (or linearized versions of them through the EKF) in order to learn model parameters through deep neural networks. This approach bears links with differentiable particle filters (DPFs) (Corenflos et al., 2021; Chen et al., 2021). In particular, the causality (in the Granger sense) within the hidden state of the state space model is directly related to the transition matrix of the model. As emphasized in (Elvira and Chouzenoux, 2022), such matrix has an interpretation as the adjacency matrix of a directed graph.

For the above reason, the approaches for graphical model inference and model parameter estimation are actually sometimes very similar, algorithmically speaking, the difference being mostly related to the interpretation of the parameters and their representation. The graphical modeling brings new insights such as the choice of specific priors (typically, sparsity) and algorithms (e.g., binary edge selection tools). In most works of graphical modeling, sparsity usually plays a key role (Brendan and Tenenbaum, 2010). Indeed, the interpretation of graphical model requires an edge selection procedure, that can be enforced by imposing a sparsity prior at the graph inference stage. A typical choice is the  $\ell_1$  (i.e., Lasso) penalty (Friedman et al., 2008; Meinshausen and Bühlmann, 2006; Chouzenoux and Elvira, 2020), which has the advantage of being convex<sup>1</sup>. Other types of priors have also been explored in (Ying et al., 2020; Benfenati et al., 2020; Chandrasekaran et al., 2012; Chouzenoux and Elvira, 2023; Elvira and Chouzenoux, 2022; Kumar et al., 2020; Hippert-Ferrer et al., 2022), with the aim of imposing specific graph structures (e.g., low rank, bounded spectrum, block sparsity). For static graph modeling, proximal algorithms (Combettes and Pesquet, 2011) (also including augmented Lagrangian primal-dual methods (Komodakis and Pesquet, 2015)) are typically the answer of choice for accounting for the various class of priors of interest in graphical inference. Discrete optimization techniques for edge selection have also been explored (see (Benfenati et al., 2018) and references therein). Dynamic graph inference has also been performed using proximal tools (Ioannidis et al., 2019). In the particular case of state-space models, as we explained earlier, several methods, for instance based on Newton or EM, are available for parameter inference. But a common drawback of most available methods is that none allow to introduce in a direct manner a graphical aware prior. Newton-based techniques are, by essence, limited to differentiable priors which prevents the use of Lasso and low-rank penalties. EM methods can be modified to incorporate a prior term on the surrogate majorant function. But this usually leads to an M-step without close form, which prevents a directly implementable algorithmic solution. In our recent works (Chouzenoux and Elvira, 2023; Elvira and Chouzenoux, 2022), we introduced more

---

1. See also (Gao et al., 2015) for state-space model inference under  $\ell_1$  prior, although not related to graphical modeling

sophisticated EM-based schemes to cope with generic convex regularizations. The goal was to estimate the transition matrix within an LG-SSM, adopting a Granger-based graphical interpretation of such matrix. A fully Bayesian approach to estimate the transition matrix was taken in (Cox and Elvira, 2022, 2023) where the space of sparse matrices was explored via reversible jump Markov chain Monte Carlo (Green and Hastie, 2009).

Despite this prolific literature, there exists a gap that we propose to fill in this work. Graphical modeling tools are either dedicated to represent static information related to statistical dependence between independent realizations of random variables (e.g., graphical Lasso approach), or to represent dynamic information related to causality behavior between time series samples. For the latter case, time series are either processed directly (Ioannidis et al., 2019), or through a state-space modeling including hidden state (Elvira and Chouzenoux, 2022). However, we are not aware of any joint approach which includes both static and dynamic graphical modeling (hence, two graphs with two distinct purposes), in the context of state-space models. Our contribution lies within this goal, as we describe in the next section.

### 1.1 Summary of our main contributions

In this work, we introduce a joint graphical modeling for representing static and dynamical behaviors in the hidden state of a linear Gaussian state-space model, bridging the gap between the static graphical Lasso model from (Friedman et al., 2008) and the dynamic model from (Elvira and Chouzenoux, 2022). We then present a novel Bayesian inference method, called DGLASSO (Dynamic Graphical Lasso), to estimate both graphs, under a sparsity prior, jointly with the construction of the filtering/smoothing distribution for the time series. The method relies on an original optimization algorithm, implementing a block alternating proximal algorithm with efficient majorization-minimization inner steps. We establish the convergence of our algorithm using recent tools from nonlinear analysis. We then perform an extensive experimental validation of the proposed model and inference algorithm by means of experiments on synthetic data and real data from weather variability analysis. For reproducibility purpose, the code for DGLASSO algorithm, is available at [https://pages.saclay.inria.fr/emilie.chouzenoux/log/DGLASSO\\_toolbox.zip](https://pages.saclay.inria.fr/emilie.chouzenoux/log/DGLASSO_toolbox.zip).

## 2. Problem setup

### 2.1 Notation

Bold symbols are used for matrix and vectors. 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  $\|\mathbf{X}\|_F$ ,  $\|\mathbf{X}\|_2$  and  $\text{tr}(\mathbf{X})$ , the Frobenius norm, the spectral norm (i.e., largest singular value), and the trace, 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  $\mathbb{R}^N$ .  $\mathcal{S}_N$  denotes symmetric matrices of  $\mathbb{R}^{N \times N}$ . Both  $\mathbb{R}^{N \times N}$  and  $\mathcal{S}_N$  are Hilbert spaces, endowed with the Frobenius norm  $\|\cdot\|_F$  and the trace scalar product  $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}(\mathbf{X}\mathbf{Y})$ .  $\mathcal{S}_N^+$  (resp.  $\mathcal{S}_N^{++}$ ) is the set of  $N \times N$  symmetric positive semidefinite (resp. definite) matrices of  $\mathbb{R}^N$ . Given a sequence of elements  $\{\mathbf{x}_k\}_{k=1}^K$  of length  $K \geq 1$  and size  $N$ , we denote each element as  $\mathbf{x}_k = (\mathbf{x}_k(n))_{1 \leq n \leq N}$  and we use the notation  $\mathbf{x}_{k_1:k_2}$  to refer to the subsequence  $\{\mathbf{x}_k\}_{k=k_1}^{k_2}$ , for  $1 \leq k_1 < k_2 \leq$

$K$ . For convex analysis concepts, we rely on the notation in the reference textbook by Bauschke and Combettes (2017). We denote  $\Gamma_0(\mathcal{H})$  the set of proper, lower semi continuous convex functions from a Hilbert space  $\mathcal{H}$  to  $(-\infty, +\infty]$  (Bauschke and Combettes, 2017, Chap. 9), and  $\partial f$  the subdifferential of  $f \in \Gamma_0(\mathcal{H})$  (Bauschke and Combettes, 2017, Chap. 16). With a slight abuse of notation, we make use of the extended form of the minus logarithm determinant function, defined as

$$(\forall \mathbf{P} \in \mathcal{S}_N) \quad -\log \det(\mathbf{P}) = \begin{cases} -\log |\mathbf{P}|, & \text{if } \mathbf{P} \in \mathcal{S}_N^{++}, \\ +\infty, & \text{otherwise,} \end{cases} \quad (1)$$

with  $|\mathbf{P}|$  the product of the eigenvalues of  $\mathbf{P}$ . According to (Bauschke and Combettes, 2017), the function in Eq. (1) belongs to  $\Gamma_0(\mathcal{S}_N)$ .

### 2.1.1 GRAPHS

We introduce here our notation for describing graphs. Most of our notation is inherited from (Bühlmann and Van De Geer, 2011).

Let us define a graph  $\mathcal{G}$  made of set of  $N$  vertices  $\mathcal{V} = \{v^{(n)} \text{ s.t. } n \in \{1, \dots, N\}\}$  and of a set of edges  $\mathcal{E} = \{e^{(n,\ell)} \text{ s.t. } (n, \ell) \in \mathbb{E}\}$ . The latter gathers ordered pairs of distinct vertices, and as such,  $\mathbb{E} \subset \{1, \dots, N\}^2$ . Undirected graphs are made of undirected edges, that is such that  $(n, \ell) \in \mathbb{E}$  and  $(\ell, n) \in \mathbb{E}$ , for every  $(n, \ell) \in \{1, \dots, N\}^2$ . In contrast, directed graphs consist of directed edges, where we say that some  $e^{(n,\ell)} \in \mathcal{E}$  is directed (from  $n$  to  $\ell$ ) if  $(\ell, n) \notin \mathcal{E}$ . We can also distinguish reflexive graphs if self-loops are allowed (i.e., one can have  $(n, n) \in \mathcal{E}$ ), and nonreflexive graphs otherwise. Given these definitions, one can simply bound the cardinality of  $\mathbb{E}$  for each category. For instance, for an undirected nonreflexive graph,  $\text{Card}(\mathbb{E}) \leq N(N-1)/2$ , while a directed nonreflexive graph has  $\text{Card}(\mathbb{E}) \leq N^2$  and a directed reflexive graph has  $\text{Card}(\mathbb{E}) \leq N(N-1)$ . Such graph definitions are binary, as it only described presence/absence of edges between vertex pairs. In this work, we require the notion of a weighted graph, where the edges  $(e^{(n,\ell)})_{(n,\ell) \in \mathbb{E}}$  are associated to real valued weights  $(\omega^{(n,\ell)})_{(n,\ell) \in \mathbb{E}}$ . The edge positions and weight values of a graph are summarized in a matrix  $\mathbf{M} \in \mathbb{R}^{N \times N}$ , where, for every  $(n, \ell) \in \mathbb{E}$ ,  $M(n, \ell) = \omega^{(n,\ell)}$  and, for every  $(n, \ell) \in \{1, \dots, N\}^2 \setminus \mathbb{E}$ ,  $M(n, \ell) = 0$ . An undirected (resp. directed) graph is thus associated to a symmetric (non symmetric) matrix  $\mathbf{M}$ . A reflexive (resp. non reflexive) graph has non zero (resp. zero) entries in the diagonal of  $\mathbf{M}$ . The number of edges is simply obtained as the number of non-zero entries in  $\mathbf{M}$ . We finally define the so-called *binary support* of the (possibly sparse) matrix  $\mathbf{M}$ . For every  $\mathbf{M} \in \mathbb{R}^{N \times N}$ ,  $\text{supp}(\mathbf{M}) = \mathbf{S} \in \{0, 1\}^{N \times N}$ , with, for every  $(n, \ell) \in \{1, \dots, N\}^2$ ,  $S(n, \ell) = 0$  if and only if  $M(n, \ell) = 0$  (i.e.,  $(n, \ell) \notin \mathbb{E}$ ).

## 2.2 Dynamical modeling

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

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k, \quad (2)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, \quad (3)$$

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 at each time  $k$  respectively,
- $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$  is the transition matrix that we aim at estimating,
- $\{\mathbf{H}_k\}_{k=1}^K \in \mathbb{R}^{N_y \times N_x}$  maps the observation model matrices, possibly varying with  $k$ , that 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 covariance matrix  $\mathbf{Q} \in \mathcal{S}_{N_x}^{++}$  that we also aim at estimating,
- $\{\mathbf{r}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{R}_k)$  is the i.i.d. observation noise process, again zero-mean Gaussian with known covariance matrices  $\mathbf{R}_k \in \mathcal{S}_{N_y}^{++}$ .

Throughout the paper, we denote  $\mathbf{P} = \mathbf{Q}^{-1}$  the precision matrix of the state noise. We furthermore 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  $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{N_x}^{++}$ . The state noises and the observation noises are mutually independent and also independent of the initial state  $\mathbf{x}_0$ .

In the next subsection, we provide some reminders about filtering/smoothing procedures for time series described by an LG-SSM.

### 2.3 Filtering and smoothing algorithms in linear dynamical systems

Both filtering and smoothing algorithms consist on the computation of a posterior probability density function (pdf) of the hidden state  $\{\mathbf{x}_k\}_{k=1}^K$ . For every  $k \in \{1, \dots, K\}$ , the filtering distribution is  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ , where we denote as  $\mathbf{y}_{1:k} = \{\mathbf{y}_j\}_{j=1}^k$  the set of observations available up to the time step  $k$ , i.e., no future observations can be used to estimate  $\mathbf{x}_k$ . The filtering problem is suitable for online processing of the observations. The smoothing distribution is  $p(\mathbf{x}_k | \mathbf{y}_{1:K})$ , where  $K$  is the final time-step for which there is an available observation, i.e., for  $k \in \{1, \dots, K-1\}$  (note that it is also possible to condition on a subset of future observations, e.g.,  $p(\mathbf{x}_k | \mathbf{y}_{1:k+\tau})$  with  $\tau \in \mathcal{N}$ ).

The filtering and smoothing distributions are in general intractable for most SSMs of interest. The LG-SSM is one of the exceptions that admit closed-form solutions.

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. For instance, for the LG-SSM described in (2)-(3), it is possible to obtain the filtering and smoothing distributions, for  $k = 1, \dots, K$ , 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. In particular, the Kalman filter (KF) (Kalman, 1960) allows to obtain recursively (in a forward manner) the sequence of filtering distributions. Its smoothing counterpart, the Rauch-Tung-Striebel (RTS) smoother (Briers et al., 2010), runs backwards to obtain the sequence of smoothing distributions. We note that both algorithms require the model parameters to be known, which in the case of the LG-SSM presented in the previous section are  $\mathbf{A}$ ,  $\mathbf{Q} = \mathbf{P}^{-1}$ ,  $\{\mathbf{H}_k\}_{k=1}^K$ , and  $\{\mathbf{R}_k\}_{k=1}^K$ . Algorithm 1 describes KF, which at each time step  $k \in \{1, \dots, K\}$  performs the (a) prediction/propagation step, where the mean  $\boldsymbol{\mu}_{k|k-1}$  and covariance  $\boldsymbol{\Sigma}_{k|k-1}$  of the (state) predictive distribution are obtained; and the (b) update step, where the mean  $\boldsymbol{\mu}_k$  and covariance  $\boldsymbol{\Sigma}_k$  of the filtering distribution

**Algorithm 1** *Kalman Filter*

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

**Recursive step.** For  $k = 1, \dots, K$

(a) Prediction/propagation step.

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{A}\boldsymbol{\mu}_{k-1} \quad (4)$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{A}\boldsymbol{\Sigma}_{k-1}\mathbf{A}^\top + \mathbf{P}^{-1} \quad (5)$$

(b) Update step.

$$\boldsymbol{\nu}_k = \mathbf{H}_k\boldsymbol{\mu}_{k|k-1} \quad (6)$$

$$\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\nu}_k \quad (7)$$

$$\mathbf{S}_k = \mathbf{H}_k\boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k \quad (8)$$

$$\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1} \quad (9)$$

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \quad (10)$$

$$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top \quad (11)$$

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

- 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:  $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{y}_k; \boldsymbol{\nu}_k, \mathbf{S}_k)$

are obtained. Algorithm 2 describes the RTS smoother. In this case, the iteration starts at  $k = K$  and runs backwards. It can be interpreted as a refinement from the mean and covariance matrices of the filtering distribution, given by Kalman, updating them with information present in future observations. However, note that the observations are not re-used in the RTS algorithm, i.e., all the required information in the observations is absorbed by the filtering distributions, which are used to produce the smoothing distributions.

## 2.4 Problem statement

Algorithms 1 and 2 are simple and efficient. However, they require the knowledge of the model parameters. In this paper, we assume  $\{\mathbf{H}_k\}_{k=1}^K$ , and  $\{\mathbf{R}_k\}_{k=1}^K$  to be known, and we address the problem of obtaining the filtering and smoothing distributions when matrices  $\mathbf{A}$  and  $\mathbf{P}$  are unknown, and must be estimated jointly with the filtering/smoothing step. To do so, we introduce a double graphical modeling of the state equations, where each matrix  $\mathbf{A}$  and  $\mathbf{P}$  now represents the weights of graphs with a specific, and complementary, statistical



**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  $\mathbf{P}$ .

**Initialization.** Set  $\boldsymbol{\mu}_K^s = \boldsymbol{\mu}_K$  and  $\boldsymbol{\Sigma}_K^s = \boldsymbol{\Sigma}_K$ .

**Recursive step.** For  $k = K, K - 1, \dots, 0$

$$\boldsymbol{\mu}_{k+1}^- = \mathbf{A}\boldsymbol{\mu}_k \tag{12}$$

$$\boldsymbol{\Sigma}_{k+1}^- = \mathbf{A}\boldsymbol{\Sigma}_k\mathbf{A}^\top + \mathbf{P}^{-1} \tag{13}$$

$$\mathbf{G}_k = \boldsymbol{\Sigma}_k\mathbf{A}^\top \left(\boldsymbol{\Sigma}_{k+1}^-\right)^{-1} \tag{14}$$

$$\boldsymbol{\mu}_k^s = \boldsymbol{\mu}_{k|k-1} + \mathbf{G}_k \left(\boldsymbol{\mu}_{k+1}^s - \boldsymbol{\mu}_{k+1}^-\right) \tag{15}$$

$$\boldsymbol{\Sigma}_k^s = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{G}_k \left(\boldsymbol{\Sigma}_{k+1}^s - \boldsymbol{\Sigma}_{k+1}^-\right) \mathbf{G}_k^\top \tag{16}$$

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

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

interpretation. We then propose an efficient and convergent inference approach to estimate both graphs under sparse priors given the observed sequence  $\{\mathbf{y}_k\}_{k=1}^K$ , while also obtaining its filtering and smoothing distributions at every time steps.

### 3. Proposed model and inference method

We now introduce our graphical modeling perspective of the LG-SSM. This novel view gives a graph-based interpretation of the transition matrix  $\mathbf{A}$  and precision matrix  $\mathbf{P}$ . Then, we present the optimization methodology in order to estimate such matrices by exploiting the graphical perspective.

#### 3.1 Sparse dynamical graphical model

Our novel approach interprets the transition matrix  $\mathbf{A}$  and precision matrix  $\mathbf{P}$  as a (weighted) directed and undirected graph, respectively. We now particularize this perspective for each matrix, deepening into the interpretability of such novel view, the complementarity of both graphs, and its benefits during the inference process.

### 3.1.1 STATE TRANSITION MATRIX

Matrix  $\mathbf{A}$  governs the hidden process in (2) and can be seen as the matrix parameter of an order-one vector auto-regressive (VAR) unobserved process. For every  $n \in \{1, \dots, N_x\}$  and  $\ell \in \{1, \dots, N_x\}$ , the entry  $A(n, \ell)$  contains the information of how the  $n$ -th time series  $\{\mathbf{x}_k(n)\}_{k=1}^K$  is affected by the  $\ell$ -th time series  $\{\mathbf{x}_k(\ell)\}_{k=1}^K$  in consecutive time steps. More precisely, we can express the update  $n$ -th dimension of the latent state in the generative model as

$$\mathbf{x}_k(n) = \sum_{\ell=1}^{N_x} A(n, \ell) \mathbf{x}_{k-1}(\ell) + \mathbf{q}_k(n). \quad (17)$$

Thus, if  $A(n, \ell) = 0$ , it implies that the  $\ell$ -th time series of the latent state does not provide any information to predict the  $n$ -th time series one time step ahead conditioned to observing the information in all time series for which  $A(n, m) \neq 0$ ,  $m \in \{1, \dots, N_x\} \setminus \ell$ . We express this conditional independence by denoting  $\mathbf{x}(n) \perp\!\!\!\perp \mathbf{x}(\ell) | \mathcal{X}_n$ . Here, for every  $n \in \{1, \dots, N_x\}$ , we denote the set  $\mathcal{X}_n = \{\mathbf{x}(m) | A(n, m) \neq 0, m \in \{1, \dots, N_x\}\}$ , i.e., the prediction of  $\mathbf{x}(n)$  observing the past of the time series in  $\mathcal{X}_n$  cannot be improved by observing the other time series. Note that in general, one can have  $A(n, \ell) \neq A(n, \ell)$  and  $A(n, n) \neq 0$ . Our interpretation is clearly connected to Granger causality (Granger, 1969), and more in particular to conditional Granger causality (Luengo et al., 2019). In particular,  $A(\ell, n) \neq 0$  for some  $(n, \ell) \in \{1, \dots, N_x\}^2$ , it means that  $\mathbf{x}_{k-1}(n)$  causes (in conditional Granger sense)  $\mathbf{x}_k(\ell)$ , for every  $k \in \{1, \dots, K\}$ . The conditional Granger causality directional relationships within the entries of the multivariate latent state time series  $\mathbf{x}$  can hence be represented as a graphical model made of a reflexive directed graph with  $N_x$  vertices, whose weights are those of the transpose matrix  $\mathbf{A}^\top$ . The perspective of matrix  $\mathbf{A}$  interpreted as a (weighted) directed graph bears some links with the work of graphical Granger causality by (Shojaie and Michailidis, 2010), although we here model the interactions in the latent space instead of directly on the observations. In our case the graphical modeling is simpler, since the work by Shojaie and Michailidis (2010) considers all possible interactions in the observation space across time (i.e., the interpreted graph size is  $K \cdot N_y$ ). The price to pay of our modeling is the difficulty in inferring  $\mathbf{A}$ , since it governs the latent process, hence it is never observed. We propose an advanced methodology to address the inferential task in Section 3.

**Illustrative example.** In Figure 1, we display an illustrative example of the graphical model associated to the following state equations for  $N_x = 5$ , for every  $k \in 1, \dots, K$ , with  $\mathbf{q}_k \in \mathbb{R}^{N_x}$  the latent state noise:

$$\begin{cases} \mathbf{x}_k(1) = 0.9 \mathbf{x}_{k-1}(1) + 0.7 \mathbf{x}_{k-1}(2) + \mathbf{q}_k(1), \\ \mathbf{x}_k(2) = -0.3 \mathbf{x}_{k-1}(3) + \mathbf{q}_k(2), \\ \mathbf{x}_k(3) = 0.8 \mathbf{x}_{k-1}(5) + \mathbf{q}_k(3), \\ \mathbf{x}_k(4) = -0.1 \mathbf{x}_{k-1}(2) + \mathbf{q}_k(4), \\ \mathbf{x}_k(5) = 0.5 \mathbf{x}_{k-1}(3) + \mathbf{q}_k(5). \end{cases} \quad (18)$$

We display the matrix  $\mathbf{A}$ , the associated binary matrix  $\text{supp}(\mathbf{A})$  and the resulting reflexive directed graph under this interpretation.

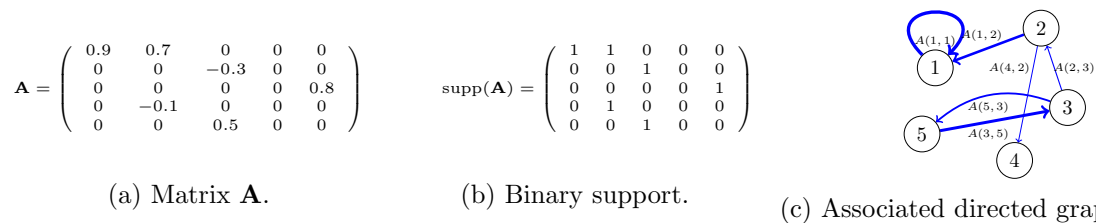


Figure 1: Graphical model associated to (18). Matrix  $\mathbf{A}$  (a), its binary support (b) and associated reflexive directed graph (c). The edges are defined as non-zero entries of  $\mathbf{A}^\top$ . Their thickness is proportional to the absolute entries of  $\mathbf{A}^\top$ .

### 3.1.2 STATE NOISE PRECISION MATRIX

Matrix  $\mathbf{Q}$  denotes the noise covariance in the state Eq. (2). Since the noise is assumed to be Gaussian, this matrix, and more precisely, the associated precision matrix  $\mathbf{P} = \mathbf{Q}^{-1}$ , also has a direct interpretation in terms of graphical modeling, using the notion of Gaussian graphical model (GGM) (Bühlmann and Van De Geer, 2011, Section 13.4)(Uhler, 2017). Since we consider  $\mathbf{Q}$  constant during the whole time series, let us denote the multivariate state noise r.v. at any time step as  $\mathbf{q} \sim \mathcal{N}(0, \mathbf{Q})$ . The GGM consists in a graphical modeling of the independence (or not) between the scalar random variables  $\mathbf{q}(1), \dots, \mathbf{q}(N_x)$ . It is easy to prove that

$$\mathbf{q}(n) \perp\!\!\!\perp \mathbf{q}(\ell) | \{\mathbf{q}(j), j \in 1, \dots, N_x \setminus \{n, \ell\}\} \iff \mathbf{P}(n, \ell) = \mathbf{P}(\ell, n) = 0, \quad (19)$$

i.e., the dimensions  $n$  and  $\ell$  of  $\mathbf{q}$  are independent given all other dimensions if and only if the entry  $\mathbf{P}(n, \ell)$  is zero (and obviously also  $\mathbf{P}(\ell, n)$  since the precision matrix is symmetric). Note that it is possible to condition in the l.h.s. of (19) only to the dimensions  $\mathbf{q}(j)$  for which  $\mathbf{P}(n, j) \neq 0$  and the equivalence would still hold. The GGM relies on an undirected graph associated to a symmetric weight matrix equals to the inverse of the covariance matrix  $\mathbf{P} = \Sigma^{-1}$ . In particular, namely  $(n, \ell) \notin \mathbb{E}$  if and only if  $P(n, \ell) = P(\ell, n) = 0$ .

This GGM construction is at the core of the famous GLASSO (Graphical Lasso) formulation (Friedman et al., 2008)(Maathuis et al., 2019, Section 9.7), whose goal is to build the maximum a posteriori estimator of  $\mathbf{P}$  given realizations of the random vector  $\mathbf{q}$  under a sparsity assumption on matrix  $\mathbf{P}$ . The sparsity is here interpreted as a way to eliminate spurious edges in the graph associated to  $\mathbf{P}$ .

**Illustrative example.** In Figure 2, we display an illustrative example on the GGM associated to a given precision matrix  $\mathbf{P}$  for  $N_x = 5$ . We show the associated binary support matrix  $\text{supp}(\mathbf{P})$  and the resulting undirected graph under this interpretation. Although self-loops (i.e., non-zero diagonal elements) occur, we removed them from the graphical representation for ease of readability.

### 3.1.3 PROPOSED UNIFYING VIEW

We now summarize the graphical perspective on both  $\mathbf{A}$  and  $\mathbf{Q}$  and describe an unifying approach, where sparsity plays a key role. Matrix  $\mathbf{A}$  is interpreted as the weight matrix of a directed graph with  $N_x$  vertices. Sparsity (i.e., absence of edge in the graph) in  $\mathbf{A}$  is

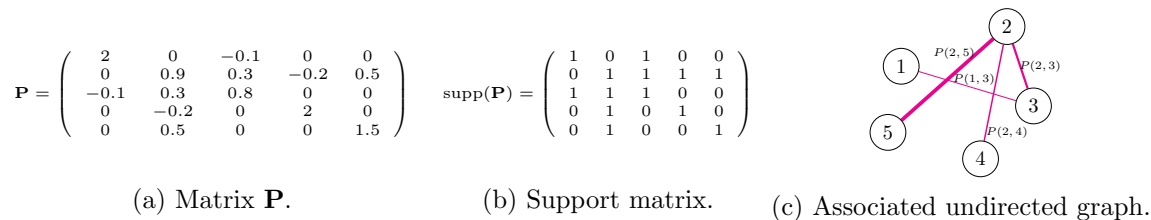


Figure 2: Matrix  $\mathbf{P}$  (a), its binary support (b), and the associated undirected graph (c) with edge thickness proportional to the absolute entries of  $\mathbf{P}$ .

interpreted as pair-wise partial/conditional independence, given a subset of the remaining time series, for a one-step ahead prediction of the hidden state. Matrix  $\mathbf{P} = \mathbf{Q}^{-1}$  is interpreted as the weight matrix of an undirected graph, related to a GGM describing the noise in the latent space. Sparsity in  $\mathbf{P}$  is interpreted as pair-wise partial/conditional independence of two dimensions of the additive state noise, given a subset of the remaining dimensions. Both graphs are reflexive, with  $N_x$  nodes and a maximum of  $N_x^2$  edges for  $\mathbf{A}$  ( $N_x(N_x - 1)$  for  $\mathbf{P}$ ) associated to  $N_x^2$  weights (i.e., the  $N_x^2$  entries of  $\mathbf{A}$  or  $\mathbf{P}$ ).

Our perspective in the state process of the LG-SSM in (2) is that  $\mathbf{A}$  encodes the way the information flows in consecutive time-steps between the nodes (state dimensions) of the network (vector state). Thus, its properties shape how the energy/information is transferred and dissipated (under the noise). In contrast,  $\mathbf{P} = \mathbf{Q}^{-1}$  encodes how information that is not in the system at time  $k - 1$  enters in the system at time  $k$ . In that respect, the interpreted graph with weight matrix  $\mathbf{P}$  encodes the dependency of the new information across the nodes of the network.

We adopt the above perspective to estimate both  $\mathbf{A}$  and  $\mathbf{Q}$  by promoting properties in both graphs. Specifically, we introduce sparsity priors on the matrices, as the sparsity property is key to reach interpretability and compactness of the whole model. In particular, it allows to understand the inner structure of the latent space. Moreover, it can be helpful to speed up computations as the sparsity level is increased, e.g., when running the Kalman filter and RTS smoother. Our proposed method DGLASSO (Dynamic Graphical Lasso) hence aims at providing the maximum a posteriori (MAP) estimator of  $\mathbf{A}$  and  $\mathbf{P}$  (i.e., the weight matrices related to the graphical modeling of the latent state correlation and causality) under Lasso sparsity regularization on both matrices, given the observed sequence  $\mathbf{y}_{1:K}$ . A visual representation of DGLASSO graphical model is given in Figure 3. The figure summarizes the relationships among the state entries of an LG-SSM using matrices ( $\mathbf{A}, \mathbf{P}$ ) from Figures 1 and 2.

**Related works:** Our approach DGLASSO generalizes important existing sparse graphical inference ones. For instance, our model with  $\mathbf{A} = 0$  (degenerate case) has no memory, and all the energy/information of the system is lost at each time step, thus the state dimensions only incorporate exogenous energy/information through the additive noises. This degenerate case is the same model than GLASSO (Friedman et al., 2008) in the case when  $\mathbf{R}_k \equiv 0$ , and same than the robust GLASSO model (Benfenati et al., 2020, Sec.5.2) when  $\mathbf{R}_k \equiv \sigma_R^2 \mathbf{Id}$ . In contrast, if the state noise covariance matrix  $\mathbf{Q}$  is known, DGLASSO coincides with our recent GraphEM framework (Elvira and Chouzenoux, 2022). Probably the

closer related work is (Ioannidis et al., 2019), which also introduces a joint graph modeling within an LG-SSM, capturing order-one causal relationships and instantaneous influence (i.e., order zero), through two sparse graphs. Their proposed inference method is an alternating optimization technique, that infers the two graphs under Lasso prior, jointly with the estimation of hidden state. In contrast with DGLASSO, in (Ioannidis et al., 2019), (i) the state model follows a structural vector autoregressive model (SVAR) where instantaneous causality and noise are distinguished, while DGLASSO assumes an order-one VAR in the hidden state; and (ii) the cost function does not result from a Bayesian modeling, and as such it is not related to a maximum a posteriori loss for the graph variables, (iii) the state estimation is point wise defined as the solution of an handcrafted optimization problem, while DGLASSO preserves a full Bayesian interpretation and hence allows the complete characterization of the filtering/smoothing state distributions. In particular, (Ioannidis et al., 2019) model does not recover GLASSO as a particular case.

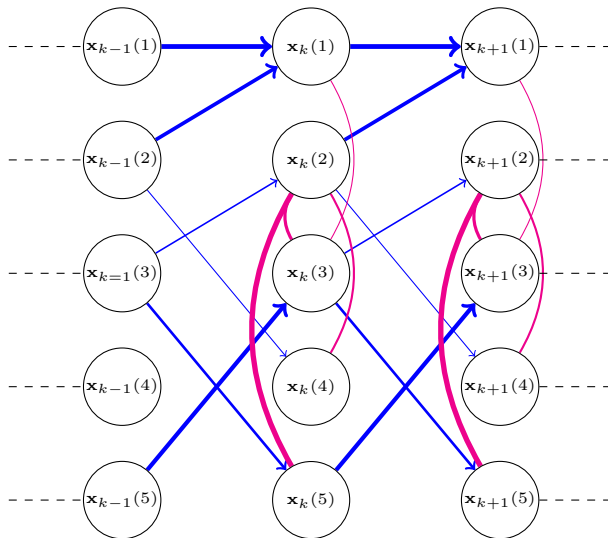


Figure 3: Summary representation of the DGLASSO graphical model, for the example graphs presented in Figs. 1 and 2. Blue (oriented) edges represent Granger causality between state entries among consecutive time steps, encoded in matrix  $\mathbf{A}$  (Fig. 1). Magenta edges represent static (i.e., instantaneous) relationships between the state entries, at every time step, due to correlated state noise described by matrix  $\mathbf{P}$  (Fig. 2).

### 3.2 Optimization problem

The considered MAP inference problem reads as an optimization problem that we formulate hereafter. More specifically, let us denote the posterior of the unknown parameter,  $p(\mathbf{A}, \mathbf{P} | \mathbf{y}_{1:K})$ , where the hidden states have been marginalized. It is direct to show, using Bayes rule and composition with the (strictly increasing) logarithmic function, that the maximum of  $p(\mathbf{A}, \mathbf{P} | \mathbf{y}_{1:K}) \propto p(\mathbf{A}, \mathbf{P})p(\mathbf{y}_{1:K} | \mathbf{A}, \mathbf{P})$ , with  $p(\mathbf{A}, \mathbf{P})$  some prior on the parameters  $\mathbf{A}$  and  $\mathbf{P}$ , coincides with the minimum of the following loss function:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) \triangleq \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) + \mathcal{L}_0(\mathbf{A}, \mathbf{P}). \quad (20)$$

with  $\mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) \triangleq -\log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P})$  and  $\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = -\log p(\mathbf{A}, \mathbf{P})$ . According to (Sarkka, 2013, Chap. 12),

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = \sum_{k=1}^K \frac{1}{2} \log \det(2\pi \mathbf{S}_k) + \frac{1}{2} \mathbf{z}_k^\top \mathbf{S}_k^{-1} \mathbf{z}_k, \quad (21)$$

where  $\mathbf{z}_k = \mathbf{y}_k - \mathbf{H}_k \mathbf{A} \boldsymbol{\mu}_{k-1}$  and  $\mathbf{S}_k$  is the covariance matrix of the predictive distribution  $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{A}, \mathbf{P}) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}_k \mathbf{A} \boldsymbol{\mu}_{k-1}, \mathbf{S}_k)$ , both being obtained by the KF in Alg. 1 run for given  $(\mathbf{A}, \mathbf{P})$  (see (Sarkka, 2013, Section 4.3)).

As already mentioned, the introduction of priors that induce sparsity is advantageous due to several reasons. First, it generally reduces over-fitting, particularly when  $K$  is low compared to the number of parameters to be estimated. Also, it enhances the interpretability. The zero elements in  $\mathbf{A}$  and  $\mathbf{P}$  have a clear interpretation of conditional independence between the time series. Ideally, we would use the  $\ell_0$  norm of  $\mathbf{A}$  and  $\mathbf{P}$ , i.e., penalizing the number of non-zero entries of the matrix. However, this norm is known to have undesirable properties such as being non-convex, non continuous, and associated to a improper law  $p(\mathbf{A})$ . We thus propose instead the regularization term

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1. \quad (22)$$

The  $\ell_1$  norm in (22), defined as the sum of absolute values of the matrix entries, is a proper convex function, that leads to the so-called Lasso regularization (Bach et al., 2012). Note that this penalty, used in numerous works of signal processing and machine learning (Tibshirani, 1996; Chaux et al., 2007), including graph signal processing (Friedman et al., 2008; Benfenati et al., 2020), is associated with a joint Laplace prior distribution on  $\mathbf{A}$  and  $\mathbf{P}$ . Such joint distribution factorizes (i.e., the prior assumes independence on both parameters), the means are zero, and the scale parameters are proportional to, respectively,  $\lambda_A$  and  $\lambda_P$ . The larger the regularization parameter  $\lambda_A$  (or  $\lambda_P$ ), the higher sparsity of  $\mathbf{A}$  (or  $\mathbf{P}$ ), with the degenerate case of a null  $\mathbf{A}$  (and  $\mathbf{P}$ ) when the regularization parameter grows.

### 3.3 General minimization procedure

The expressions (21)-(22) provide an effective way to evaluate (20). However, due to the recursive form in (21), it is challenging to derive direct quantities (e.g., gradient) for  $\mathcal{L}_{1:K}$ . Moreover, despite its simple expression, the regularization term (22) is non differentiable. For both reasons, the minimization of (20) is a challenging question.

We propose a block alternating majorization-minimization (MM) technique to jointly infer the probabilistic filtering/smoothing distributions of the SSM, along with MAP estimates of  $(\mathbf{A}, \mathbf{P})$ . Our method presents the advantage of sound convergence guarantees and the ability to incorporate sparsity priors on both  $\mathbf{A}$  and  $\mathbf{P}$ . The general idea of MM is to replace a complicated optimization problem by a sequence of more tractable ones (Sun et al., 2016; Hunter and Lange, 2004). Surrogate approximations of the cost function are built iteratively, following a majorization principle. For any estimates  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$  (i.e., in the interior domain of definition of  $\mathcal{L}$ ) of  $(\mathbf{A}, \mathbf{P})$ , a majorizing approximation is constructed for the likelihood term. It is required to satisfy both

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \geq \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}), \quad (23)$$

and also the so-called tangency condition

$$\mathcal{Q}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}). \quad (24)$$

The MM algorithm then alternates between **M**ajorization step to build function  $\mathcal{Q} + \mathcal{L}_0$  satisfying conditions (23) and (24), and **M**inimization step to minimize this majorizing approximation. In our proposed approach, we adopt a block alternating implementation of the MM method, where only one variable (i.e.,  $\mathbf{A}$  or  $\mathbf{P}$ ) is updated at each iteration, following a cyclic rule (Jacobson and Fessler, 2007; Hong et al., 2016). This alternating strategy considerably simplifies the majorizing function construction as well as its minimization. We furthermore add so-called proximal terms in both updates. Let us recall that, for a function  $\phi : \mathcal{H} \mapsto (-\infty, +\infty] \in \Gamma_0(\mathcal{H})$ , with  $\mathcal{H}$  a Hilbert space with endowed norm  $\|\cdot\|$ , the proximity operator<sup>2</sup> of function  $f$  at  $\tilde{\mathbf{V}} \in \mathcal{H}$  is defined as (Combettes and Pesquet, 2011)

$$\text{prox}_\phi(\tilde{\mathbf{V}}) = \underset{\mathbf{V} \in \mathcal{H}}{\text{argmin}} \left( \phi(\mathbf{V}) + \frac{1}{2} \|\mathbf{V} - \tilde{\mathbf{V}}\|^2 \right). \quad (25)$$

In our context, the considered Hilbert space is either  $\mathbb{R}^{N_x \times N_x}$  for the update of the transition matrix or  $\mathcal{S}_{N_x}$  for the update of the precision matrix, and the endowed norm is in both cases the Frobenius norm  $\|\cdot\|_F$ . Introducing proximity terms thus amounts to adding to each majorizing function a quadratic distance to the previous iterate, weighted by a positive factor. This modification preserves the MM interpretation of the method, while ensuring improved stability and convergence guarantees. As we show below, the iterates belong to the interior of domain of the loss function by construction. Namely for every  $i \in \mathbb{N}$ ,  $(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \in \mathbb{R}^{N_x \times N_x} \times \mathcal{S}_{N_x}^{++}$ , so the precision matrix remains invertible along the iterations and the algorithm is well defined. The resulting DGLASSO approach is summarized in Algorithm 3. DGLASSO aims at providing ultimately the MAP estimates for the matrix parameters  $(\mathbf{A}, \mathbf{P})$  of the considered LG-SSM, through the minimization of (20). The covariance state noise MAP estimate is straightforwardly obtained by inversion of the precision state noise matrix provided as an output of DGLASSO. The state filtering/smoothing pdf associated to each estimates are obtained by running KF/RTS loops when setting  $(\mathbf{A}, \mathbf{P})$  equal to DGLASSO outputs. Next sections are dedicated to the (i) construction of the majorizing function, (ii) discussion about the resolution of each inner step, and (iii) convergence analysis.

### 3.4 Building the majorizing function

In this section, we derive a theorem regarding the expression of the loss function  $\mathcal{L}_{1:K}$  and a valid majorant function for it.

**Theorem 1** *The loss function can be expressed as*

$$\begin{aligned} (\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) &= \frac{1}{2} \sum_{k=1}^K \left( (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \right) \\ &\quad - \frac{K}{2} \log \det(2\pi\mathbf{P}) + \log p(\mathbf{x}_{0:K} | \mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}) + \log p(\mathbf{x}_0) - \sum_{k=1}^K p(\mathbf{y}_k | \mathbf{x}_k). \end{aligned} \quad (28)$$

---

2. See also <http://proximity-operator.net/>

**Algorithm 3** DGLASSO algorithm

**Inputs.** Prior parameters  $\boldsymbol{\mu}_0$  and  $\boldsymbol{\Sigma}_0$ ; model parameters  $\{\mathbf{H}_k\}_{k=1}^K$  and  $\{\mathbf{R}_k\}_{k=1}^K$ ; set of observations  $\{\mathbf{y}_k\}_{k=1}^K$ ; hyper-parameters  $(\lambda_A, \lambda_P) > 0$ ; stepsizes  $(\theta_A, \theta_P) > 0$ ; precisions  $(\varepsilon, \xi) > 0$ .

**Initialization.** Set  $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)}) \in \mathbb{R}^{N_x \times N_x} \times \mathcal{S}_{N_x}^{++}$ .

**Recursive step.** For  $i = 0, 1, \dots$ :

(a) //Update transition matrix

(i) Build surrogate function satisfying (23) and (24) at  $\tilde{\mathbf{A}} = \mathbf{A}^{(i)}$  and  $\tilde{\mathbf{P}} = \mathbf{P}^{(i)}$ .

(ii) Run Algorithm 4 with precision  $\xi$  to solve

$$\begin{aligned} \mathbf{A}^{(i+1)} &= \text{prox}_{\mathbf{A} \rightarrow \theta_A \mathcal{Q}(\mathbf{A}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \theta_A \lambda_A \|\mathbf{A}\|_1} (\mathbf{A}^{(i)}), \\ &= \underset{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}}{\text{argmin}} \mathcal{Q}(\mathbf{A}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \mathbf{A}^{(i)}\|_F^2. \end{aligned} \quad (26)$$

(b) //Update noise precision matrix

(i) Build surrogate function satisfying (23) and (24) at  $\tilde{\mathbf{A}} = \mathbf{A}^{(i+1)}$  and  $\tilde{\mathbf{P}} = \mathbf{P}^{(i)}$ .

(ii) Run Algorithm 5 with precision  $\xi$  to solve

$$\begin{aligned} \mathbf{P}^{(i+1)} &= \text{prox}_{\mathbf{P} \rightarrow \theta_P \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \theta_P \lambda_P \|\mathbf{P}\|_1} (\mathbf{P}^{(i)}), \\ &= \underset{\mathbf{P} \in \mathcal{S}_{N_x}}{\text{argmin}} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_P} \|\mathbf{P} - \mathbf{P}^{(i)}\|_F^2. \end{aligned} \quad (27)$$

If  $\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F \leq \varepsilon \|\mathbf{A}^{(i)}\|_F$  and  $\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F \leq \varepsilon \|\mathbf{P}^{(i)}\|_F$ , stop the recursion by returning  $(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)})$ .

**Output.** MAP estimates of the transition and state noise precision matrices.

Moreover, consider the outputs of Algs. 1 and 2 for a given  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ . Denote

$$\begin{cases} \boldsymbol{\Psi} &= \frac{1}{K} \sum_{k=1}^K (\boldsymbol{\Sigma}_k^s + \boldsymbol{\mu}_k^s (\boldsymbol{\mu}_k^s)^\top), \\ \boldsymbol{\Delta} &= \frac{1}{K} \sum_{k=1}^K (\boldsymbol{\Sigma}_k^s \mathbf{G}_{k-1}^\top + \boldsymbol{\mu}_k^s (\boldsymbol{\mu}_{k-1}^s)^\top), \\ \boldsymbol{\Phi} &= \frac{1}{K} \sum_{k=1}^K (\boldsymbol{\Sigma}_{k-1}^s + \boldsymbol{\mu}_{k-1}^s (\boldsymbol{\mu}_{k-1}^s)^\top), \end{cases} \quad (29)$$



where, for every  $k \in \{1, \dots, K\}$ ,  $\mathbf{G}_k = \Sigma_k \tilde{\mathbf{A}}^\top (\tilde{\mathbf{A}} \Sigma_k \tilde{\mathbf{A}}^\top + \tilde{\mathbf{P}}^{-1})^{-1}$  (see Alg. 2). Then, conditions (23) and (24) hold with

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \frac{K}{2} \text{tr} \left( \mathbf{P}(\Psi - \Delta \mathbf{A}^\top - \mathbf{A} \Delta^\top + \mathbf{A} \Phi \mathbf{A}^\top) \right) - \frac{K}{2} \log \det(2\pi \mathbf{P}). \quad (30)$$

As a consequence, for every  $(\theta_A, \theta_P) > 0$ ,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) \leq \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - \mathcal{Q}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P} - \tilde{\mathbf{P}}\|_F^2, \quad (31)$$

with equality holding for  $\mathbf{A} = \tilde{\mathbf{A}}$  and  $\mathbf{P} = \tilde{\mathbf{P}}$ .

**Proof** See Appendix A. ■

Theorem 1 allows to build, for any tangent point  $(\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ , a majorizing approximation (31) for  $\mathcal{L}$ . DGLASSO method leverages this property designing a block alternating MM scheme. At each iteration  $i \in \mathbb{N}$ , two steps are conducted, namely the update of (a) the transition matrix, (b) the noise precision matrix. Each steps follows an MM structure, that is it first builds a majorizing approximation for  $\mathcal{L}$  at the current estimates, using Theorem 1, and then minimizes it with respect to the active variable ( $\mathbf{A}$  in step (a), or  $\mathbf{P}$  in step (b)). Processing the variables in two distinct steps allows to build upon the desirable convex structure of (31) with respect to one of the variable, the other being fixed. The good performance of MM approaches combined with block alternating steps have been illustrated in (Hong et al., 2016; Chouzenoux et al., 2016; Hien et al., 2020). In particular, convergence guarantees are at reach, as we will show in Section 4.

**Remark 2** *From Theorem 1, the construction of the majorizing function of  $\mathcal{L}$  at each tangent point  $(\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$  requires to run the KF Algorithm 1 and the RTS Algorithm 2. This feature was already present in previous EM-based approaches for LG-SSM parameter estimations (see for instance (Sarkka, 2013, Chap. 12) and (Elvira and Chouzenoux, 2022)). Due to its block alternating form, the proposed DGLASSO algorithm requires to build twice per iteration a majorizing approximation, which means that the data must be processed by KF/RTS twice per iteration. In contrast, gradient-based methods relying on sensitivity equations (Gupta and Mehra, 1974) only require the KF recursions for building their updates, but not the RTS recursions, being thus more computationally efficient. However, such approaches cannot easily encompass non-smooth sparsity priors. Moreover, if no prior is assumed (i.e.,  $\lambda_A = \lambda_P = 0$ ), the aforementioned methods exhibit very slow convergence in this problem as we will show in our numerical experiments. This might be because the gradient of the neg-log-likelihood is extremely ill-conditioned in this context.*

### 3.5 Resolution of the inner problems

We now discuss the structure and resolution of the inner problems (26) and (27) arising in Algorithm 3.

**Minimization with respect to  $\mathbf{A}$ :** Let  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ . By definition of the proximity operator (25) and the majorant expression in (30), Eq. (26) amounts to minimizing the following function

$$\begin{aligned} \text{minimize}_{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}} \left( \mathcal{C}_1(\mathbf{A}) \triangleq \frac{\theta_A K}{2} \text{tr} \left( \tilde{\mathbf{P}}(\Psi - \Delta \mathbf{A}^\top - \mathbf{A} \Delta^\top + \mathbf{A} \Phi \mathbf{A}^\top) \right) + \theta_A \lambda_A \|\mathbf{A}\|_1 \right. \\ \left. + \frac{1}{2} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 \right). \end{aligned} \quad (32)$$

Remarkably, the problem above is a special instance of a multivariate Lasso regression problem (Tibshirani, 1996), for which many efficient iterative solvers are available. The specificity here is that the problem is strongly convex thanks to the proximal term. We thus suggest the use of the Dykstra-like algorithm by Bauschke and Combettes (2008), whose iterations are recalled in the Appendix B. This method presents the advantage of fast convergence rate, ease of implementation and no parameter tuning.

**Minimization with respect to  $\mathbf{P}$ :** Let  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ . The update of Eq. (27) solves a minimization problem with generic form

$$\text{minimize}_{\mathbf{P} \in \mathcal{S}_{N_x}} \left( \mathcal{C}_2(\mathbf{P}) \triangleq \frac{\theta_P K}{2} \text{tr}(\mathbf{P} \mathbf{\Pi}) - \frac{\theta_P K}{2} \log \det \mathbf{P} + \theta_P \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2} \|\mathbf{P} - \tilde{\mathbf{P}}\|_F^2 \right), \quad (33)$$

where we denote

$$\mathbf{\Pi} = \Psi - \Delta \tilde{\mathbf{A}}^\top - \tilde{\mathbf{A}} \Delta^\top + \tilde{\mathbf{A}} \Phi \tilde{\mathbf{A}}^\top. \quad (34)$$

Here we have used the definition of the proximity operator (25) and the majorant expression in (30) (ignoring the constant multiplicative term in the logarithm). Matrix  $\mathbf{\Pi} \in \mathcal{S}_{N_x}$  in (34) is constant with respect to variable  $\mathbf{P}$ . Remarkably, (33) reads as a regularized form of the famous GLASSO problem (Friedman et al., 2008), and gets actually equivalent to it when  $\theta_P \rightarrow \infty$ . Matrix  $\mathbf{\Pi}$  plays the role of the empirical covariance matrix in GLASSO, and  $\frac{2\lambda_P}{K}$  acts as the weight on the  $\ell_1$  term. The proximal term works as a Tikhonov-like regularizer, ensuring the strong convexity of the problem and thus the unicity of its minimizer. Moreover, by the definition of the log-determinant in Eq. (1), the solution of (33) belongs to  $\mathbf{P} \in \mathcal{S}_{N_x}^{++}$ , i.e., the precision matrix is invertible and thus a covariance matrix can be deduced from it by inversion. Standard GLASSO solvers can be easily modified to solve (33). We present in the Appendix B the complete derivations, when applying the Dykstra-like algorithm from (Bauschke and Combettes, 2008), which presents the advantage of fast convergence and ease of implementation.

#### 4. Convergence analysis

We now present our convergence proof for the proposed DGLASSO algorithm presented in Algorithm 3. Our analysis is made assuming that the inner steps (26) and (27) are solved in an exact manner. The extension of the result to the case of an inexact resolution of the subproblems is discussed at the end of the section.

#### 4.1 Descent property

**Lemma 1** *Assuming exact resolution of (26) and (27), the sequence  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$  produced by DGLASSO algorithm satisfies*

$$(\forall i \in \mathbb{N}) \quad \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}). \quad (35)$$

**Proof** Let  $i \in \mathbb{N}$ . First, let us Theorem 1 at  $(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = (\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)})$  (Inequality (a)), and the definition of  $\mathbf{P}^{(i+1)}$  in (27) (Inequality (b)),

$$\begin{aligned} \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) &\stackrel{(a)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i+1)}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F^2 \end{aligned} \quad (36)$$

$$\begin{aligned} &\stackrel{(b)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2. \end{aligned} \quad (37)$$

Inequality (37) simplifies into

$$\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 = \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}). \quad (38)$$

Applying Theorem 1 now at  $(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = (\mathbf{A}^{(i)}, \mathbf{P}^{(i)})$  (Inequality (a)), and the definition of  $\mathbf{A}^{(i+1)}$  in (26) (Inequality (b)), leads to

$$\begin{aligned} \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) &\stackrel{(a)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2 \end{aligned} \quad (39)$$

$$\begin{aligned} &\stackrel{(b)}{\leq} \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A}^{(i)} - \mathbf{A}^{(i)}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2, \end{aligned} \quad (40)$$

which simplifies into

$$\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \leq \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}^{(i)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 = \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}), \quad (41)$$

and concludes the proof.  $\blacksquare$

If the cost function  $\mathcal{L}$  is lower bounded (e.g., if it is coercive), Lemma 1 implies the convergence of sequence  $\{\mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)})\}_{i \in \mathbb{N}}$  to a finite value and, as such, the existence of cluster points in  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$ . This is however a rather weak convergence result and we propose hereafter a thorough analysis relying on recent tools of nonlinear analysis (Attouch et al., 2010; Bolte et al., 2014) combined with the works (Tien et al., 2020; Hien et al., 2020) on the convergence of block alternating MM schemes.

## 4.2 Convergence guarantees

**Theorem 3** Consider the sequence  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$  generated by DGLASSO, assuming exact resolution of both inner steps (26) and (27). If the sequence  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$  is bounded, then  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$  converges to a critical point of  $\mathcal{L}$ .

**Proof** The convergence analysis relies in proving that the exact form of DGLASSO algorithm is a special instance of TITAN algorithm from (Tien et al., 2020), and as such, inherits (Tien et al., 2020, Theorem 7) and (Tien et al., 2020, Theorem 8), under our assumptions.

- Let us introduce the following notations.

$$\begin{aligned}
 (\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad f(\mathbf{A}, \mathbf{P}) &= \frac{1}{2} \sum_{k=1}^K \left( (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \right) \\
 &+ \log p(\mathbf{x}_{0:K} | \mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}) - \log p(\mathbf{x}_0) - \sum_{k=1}^K \log p(\mathbf{y}_k | \mathbf{x}_k), \quad (42)
 \end{aligned}$$

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad g_1(\mathbf{A}) = \lambda_A \|\mathbf{A}\|_1, \quad (43)$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad g_2(\mathbf{P}) = -\frac{K}{2} \log \det(2\pi\mathbf{P}) + \lambda_P \|\mathbf{P}\|_1, \quad (44)$$

so that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) = f(\mathbf{A}, \mathbf{P}) + g_1(\mathbf{A}) + g_2(\mathbf{P}), \quad (45)$$

with  $f$  lower semi-continuous function,  $g_1 \in \Gamma_0(\mathbb{R}^{N_x \times N_x})$  and  $g_2 \in \Gamma_0(\mathcal{S}_{N_x})$ . Moreover, let us denote

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{J}(\mathbf{A}, \mathbf{P}) = \frac{K}{2} \text{tr} \left( \mathbf{P}(\Psi - \Delta\mathbf{A}^\top - \mathbf{A}\Delta^\top + \mathbf{A}\Phi\mathbf{A}^\top) \right), \quad (46)$$

and, for every  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ ,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad u_1(\mathbf{A}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \mathcal{J}(\mathbf{A}, \tilde{\mathbf{P}}) + \frac{1}{2\theta_A} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 + f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - \mathcal{J}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}), \quad (47)$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad u_2(\mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \mathcal{J}(\tilde{\mathbf{A}}, \mathbf{P}) + \frac{1}{2\theta_P} \|\mathbf{P} - \tilde{\mathbf{P}}\|_F^2 + f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - \mathcal{J}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}). \quad (48)$$

By Theorem 1, the following majorization properties hold for every  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ ,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad u_1(\mathbf{A}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \geq f(\mathbf{A}, \tilde{\mathbf{P}}), \quad (49)$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad u_2(\mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \geq f(\tilde{\mathbf{A}}, \mathbf{P}). \quad (50)$$

and we have the tangency condition, for every  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ ,

$$u_1(\tilde{\mathbf{A}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}), \quad (51)$$

$$u_2(\tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}). \quad (52)$$

Then, straightforward computations allow to rewrite the iterates of Algorithm 3 as follows:

$$(\forall i \in \mathbb{N}) \quad \begin{cases} \mathbf{A}^{(i+1)} = \operatorname{argmin}_{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}} u_1(\mathbf{A}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + g_1(\mathbf{A}), \\ \mathbf{P}^{(i+1)} = \operatorname{argmin}_{\mathbf{P} \in \mathcal{S}_{N_x}} u_2(\mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + g_2(\mathbf{P}), \end{cases} \quad (53)$$

which identifies with the iterative scheme TITAN from (Tien et al., 2020), in the case of two blocks and setting the extrapolation step to zero. The rest of the proof amounts to check the fulfillment of the assumptions required for (Tien et al., 2020, Theorem 7) and (Tien et al., 2020, Theorem 8).

- Let us denote, for every  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ ,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \tilde{u}_1(\mathbf{A}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = u_1(\mathbf{A}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + g_1(\mathbf{A}), \quad (54)$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \tilde{u}_2(\mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = u_2(\mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + g_2(\mathbf{P}). \quad (55)$$

Functions  $u_1$  and  $u_2$  are quadratic and strongly convex with respective strong convexity constants  $\theta_A^{-1}$  and  $\theta_P^{-1}$ . Since both  $g_1$  and  $g_2$  are convex, functions  $\tilde{u}_1$  and  $\tilde{u}_2$  are also strongly convex, with respective strong convexity constants  $\theta_A^{-1}$  and  $\theta_P^{-1}$ . Let  $i \in \mathbb{N}$ . According to the optimality conditions of both equations in (53), there exists  $\mathbf{T}_1^{(i+1)} \in \partial \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \in \mathbb{R}^{N_x \times N_x}$  and  $\mathbf{T}_2^{(i+1)} \in \partial \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \in \mathcal{S}_{N_x}$  such that

$$\begin{cases} \operatorname{tr} \left( \mathbf{T}_1^{(i+1)} (\mathbf{A}^{(i)} - \mathbf{A}^{(i+1)}) \right) \geq 0, \\ \operatorname{tr} \left( \mathbf{T}_2^{(i+1)} (\mathbf{P}^{(i)} - \mathbf{P}^{(i+1)}) \right) \geq 0. \end{cases} \quad (56)$$

Moreover, by strong convexity of both  $\tilde{u}_1$  and  $\tilde{u}_2$ ,

$$\begin{cases} \tilde{u}_1(\mathbf{A}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \operatorname{tr} \left( \mathbf{T}_1^{(i+1)} (\mathbf{A}^{(i)} - \mathbf{A}^{(i+1)}) \right) + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F^2, \\ \tilde{u}_2(\mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \operatorname{tr} \left( \mathbf{T}_2^{(i+1)} (\mathbf{P}^{(i)} - \mathbf{P}^{(i+1)}) \right) + \frac{1}{2\theta_P} \|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F^2. \end{cases} \quad (57)$$

Hence, using (56),

$$\begin{cases} \tilde{u}_1(\mathbf{A}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F^2, \\ \tilde{u}_2(\mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_P} \|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F^2, \end{cases} \quad (58)$$

and, using (49)-(50)-(51)-(52),

$$\begin{cases} \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_A} \|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F^2, \\ \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) + \frac{1}{2\theta_P} \|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F^2. \end{cases} \quad (59)$$

It means that the so-called NDSP (nearly sufficiently decreasing property) condition from (Tien et al., 2020) holds with, for every  $i \in \mathbb{N}$ ,  $(\gamma_1^{(i)}, \gamma_2^{(i)}, \eta_1^{(i)}, \eta_2^{(i)}) \equiv (0, 0, \theta_A^{-1}, \theta_P^{-1})$ . Remark that our proof for (59) constitutes an alternative proof for Lemma 1.

• According to (Gupta and Mehra, 1974), function  $f$  is continuously differentiable. As  $g_1 \in \Gamma_0(\mathbb{R}^{N_x \times N_x})$  and  $g_2 \in \Gamma_0(\mathcal{S}_{N_x})$ , we have for every  $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$  and  $\mathbf{P} \in \mathcal{S}_{N_x}$ ,

$$\begin{cases} \partial_A (f(\mathbf{A}, \mathbf{P}) + g_1(\mathbf{A})) = \nabla_A f(\mathbf{A}, \mathbf{P}) + \partial g_1(\mathbf{A}), \\ \partial_P (f(\mathbf{A}, \mathbf{P}) + g_2(\mathbf{P})) = \nabla_P f(\mathbf{A}, \mathbf{P}) + \partial g_2(\mathbf{P}), \end{cases} \quad (60)$$

so that (Tien et al., 2020, Assumption 3(i)) holds. Moreover, by construction of the majorizing function  $\mathcal{J}$ , it is also continuously differentiable and we have the coincidence of the gradient at the majorization point, namely for every  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ ,

$$\nabla_A f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \nabla \tilde{u}_1(\tilde{\mathbf{A}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}), \quad (61)$$

$$\nabla_P f(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \nabla \tilde{u}_2(\tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}). \quad (62)$$

Thus, according to (Tien et al., 2020, Lem.2), (Tien et al., 2020, Assumption 2) is verified. Moreover, as the restriction of  $\mathcal{J}$  to both of its variables is quadratic, the partial gradients  $\nabla u_1$  and  $\nabla u_2$  are linear and thus Lipschitz continuous on any bounded subset of  $\mathbb{R}^{N_x \times N_x}$  and  $\mathcal{S}_{N_x}$ , respectively, which yields the fulfillment of (Tien et al., 2020, Assumption 3).

• Since NSDP and (Tien et al., 2020, Assumption 2) hold, the result (Tien et al., 2020, Theorem 6) allows to show that, if the sequence  $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i \in \mathbb{N}}$  is bounded, then every limit point  $(\mathbf{A}^*, \mathbf{P}^*)$  of it is a critical point of  $\mathcal{L}$ . Moreover, (Tien et al., 2020, Assumption 3) is satisfied and (Tien et al., 2020, Cond.1) and (Tien et al., 2020, Cond.4) are trivially met in our case. We can also show that the loss function  $\mathcal{L}$  satisfies the Kurdyka-Lojasiewicz property (Bolte et al., 2014) (using a similar proof than (Chouzenoux et al., 2019, Lemma 2)). Thus, (Tien et al., 2020, Theorem 7) holds which concludes our proof. ■

### 4.3 Discussion

The proof of Theorem 3 is grounded on the recent works by Tien et al. (2020) and Hien et al. (2020), generalizing the works by Bolte et al. (2014) and Chouzenoux et al. (2016), for establishing the convergence of block alternating MM schemes under the Kurdyka-Lojasiewicz (KL) inequality assumption (Lojasiewicz, 1963). The latter assumption is a powerful tool of nonlinear functional analysis that has been popularized in the seminal paper by Attouch et al. (2010). In their paper, the authors show how to prove the convergence of the iterates generated by a large family of minimization schemes, under the sole requirement that the function to minimize satisfies the KL inequality. The latter requirement is very mild, as it holds for a large class of functions, non necessarily convex, as soon as they can be embedded within an o-minimal structure. Semi-algebraic and analytical functions are example of such functions. In our analysis, for the sake of conciseness, we skipped the proof showing that  $\mathcal{L}$  satisfies KL as it is identical to the one of (Chouzenoux et al., 2019, Lem.2).

Theorem 3 assumes that both inner steps (26) and (27) are solved in an exact manner. Due to strong convexity, each problem has a unique solution, which ensures the well posedness of the study. However, in both case, the solution does not take a closed form (except

when  $\lambda_A$  or  $\lambda_P$  equals zero, in such case we retrieve the MLEM updates from (Sarkka, 2013, Chap. 12)). Iterative inner solvers are thus necessary, and we proposed some efficient ones in Section 3.5. The extension of our convergence study to the case of inexact resolution of (26) and (27) is not straightforward, up to our knowledge. Convergence of inexact proximal schemes for KL losses has been studied in various works, such as (Attouch et al., 2013; Cherni et al., 2020; Chouzenoux et al., 2014) to name a few. But we are not aware of any study covering the block alternating MM scheme considered here, and thus left the convergence study of the inexact implementation of DGLASSO as future work. In practice, we impose a rather demanding condition on the stopping rule for the inner solvers of (26) and (27) and did not observe any numerical instabilities of the proposed algorithm.

## 5. Experiments

We perform a thorough evaluation study on various controlled scenarios where the ground truth matrices denoted  $(\mathbf{A}^*, \mathbf{P}^*)$  (as well as  $\mathbf{Q}^* = (\mathbf{P}^*)^{-1}$ ) are predefined, and the time series  $\{\mathbf{y}_k, \mathbf{x}_k\}_{k=1}^K$  are build directly from the LG-SSM model (2)-(3) using such matrices. The goal is then to provide estimates  $(\hat{\mathbf{A}}, \hat{\mathbf{P}}, \hat{\mathbf{Q}})$  of  $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$ , given the observation of  $\{\mathbf{y}_k\}_{k=1}^K$ . Except otherwise stated, all the compared methods have moreover access to a perfect knowledge of  $(\mathbf{R}_k, \mathbf{H}_k, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ , in top of accessing the time series  $\{\mathbf{y}_k\}_{k=1}^K$ . The hidden state  $\{\mathbf{x}_k\}_{k=1}^K$  is, by definition, not assumed to be known, and is only used to compute quality metrics on test sets. We first work on synthetic data in Section 5.1, where the structure, the sparsity level, the conditioning, of the sought matrices  $(\mathbf{A}^*, \mathbf{P}^*)$  is controlled. This allows us to evaluate our method on multiple cases, to discuss its parameter tuning, and to compare it to benchmarks in terms of inference quality and complexity. We then address in Section 5.2, a set of realistic problems of graph inference arising in weather variability analysis, using four datasets built upon the Neurips data challenge (Runge et al., 2020). This second set of experiments aims at evaluating the ability of DGLASSO to model and estimate a large class of graph structures (here, 200 different graphs per dataset), in comparison with other state-of-the-art graph detection methods.

All codes are run on a Desktop Dell Latitude computer, with 11th Gen Intel(R) Core(TM) i7-1185G7 at 3.00GHz, equipped with 32Go Ram, using Matlab R2021a software. The code is publicly available at [https://pages.saclay.inria.fr/emilie.chouzenoux/log/DGLASSO\\_toolbox.zip](https://pages.saclay.inria.fr/emilie.chouzenoux/log/DGLASSO_toolbox.zip), for reproducibility purpose.

### 5.1 Synthetic data

#### 5.1.1 DATASETS

We set  $K = 10^3$ ,  $\mathbf{R}_k = \sigma_{\mathbf{R}}^2 \mathbf{Id}_{N_y}$  for every  $k \in \{1, \dots, K\}$ ,  $\boldsymbol{\mu}_0 \in \mathbb{R}^{N_x}$  is a vector of ones,  $\boldsymbol{\Sigma}_0 = \sigma_0^2 \mathbf{Id}_{N_x}$  with  $(\sigma_{\mathbf{R}}, \sigma_0) = (10^{-1}, 10^{-4})$ . Matrix  $\mathbf{H}_k$  is set to identity matrix for every  $k \in \{1, \dots, K\}$ , so that  $N_x = N_y$ . This setting models a one-to-one correspondence between states and observations. Identifiability issues are hence avoided.

We set  $N_x = N_y = 9$ , and we rely on block-diagonal matrices  $(\mathbf{A}^*, \mathbf{P}^*)$ , made of 3 blocks with dimensions  $3 \times 3$ . The diagonal blocks of  $\mathbf{A}^*$  are randomly set as matrices of auto-regressive processes of order one, AR(1), rescaled to have a spectral norm equals 0.9 to ensure the process stability. The diagonal blocks of  $\mathbf{P}^*$  are randomly set following

the procedure from (Moré and Toraldo, 1989), with predefined conditioning number  $c$ , with  $\log_{10}(c) \in \{0.1, 0.2, 0.5, 1\}$ , leading to datasets A, B, C and D, respectively.

DGLASSO provides estimates  $(\hat{\mathbf{A}}, \hat{\mathbf{P}})$  as a direct output. The estimate  $\hat{\mathbf{Q}}$  is simply deduced as  $\hat{\mathbf{Q}} = (\hat{\mathbf{P}})^{-1}$ . Precision parameters in DGLASSO are set to  $(\varepsilon, \xi) = (10^{-3}, 10^{-4})$ . We initialize DGLASSO with  $\mathbf{P}^{(0)} = 10^{-1}\mathbf{Id}_{N_x}$ , and  $\mathbf{A}^{(0)}$  equals to a stable auto-regressive order one matrix with entries  $A^{(0)}(n, m) = (10^{-1})^{|n-m|}$ , rescaled to have spectral norm equals to 0.99. The setting of regularization parameters  $(\lambda_A, \lambda_P)$  is discussed in the dedicated Section 5.1.4. Performance of the method for varying initializations is discussed in Appendix C.2.

### 5.1.2 COMPARISONS TO OTHER METHODS

In our experiments, we also compare DGLASSO with other model inference techniques for LG-SSM.

We consider the EM method from (Shumway and Stoffer, 1982) (denoted after by MLEM) to compute  $(\hat{\mathbf{A}}, \hat{\mathbf{Q}})$  as maximum likelihood estimates (i.e., no regularization is employed) of matrices  $(\mathbf{A}^*, \mathbf{Q}^*)$ , the estimation  $\hat{\mathbf{P}}$  being defined here as the inverse of  $\hat{\mathbf{Q}}$ . Comparison with another state-of-the-art approach for maximum likelihood computation, relying on sensitivity function minimization from (Gupta and Mehra, 1974), is presented in Appendix C.1.

Second, we consider three model inference techniques that explicitly incorporate a sparse graphical prior knowledge on the sought matrices. Namely, we compare to GLASSO (Friedman et al., 2008), that considers a simple static and noiseless version of the LG-SSM (i.e.,  $\hat{\mathbf{A}}$  is empirically set to zero and observation noise is neglected). Matrix  $\hat{\mathbf{P}}$  is then obtained through a maximum a posteriori formulation under an  $\ell_1$  prior. The convex GLASSO loss is minimized using the proximal splitting method described in (Benfenati et al., 2020, Alg.1). Matrix  $\hat{\mathbf{Q}}$  is deduced by inversion of the resulting  $\hat{\mathbf{P}}$ . We also compare with the robust GLASSO (rGLASSO) approach introduced in (Benfenati et al., 2020), that explicitly accounts for the expression of  $\mathbf{R}_k$  in the maximum a posteriori loss expression. For the sake of fair comparison, we use hereagain an  $\ell_1$  penalty to obtain  $\hat{\mathbf{P}}$  although more sophisticated priors could be encompassed by rGLASSO. For both aforementioned methods, we rely on the `Matlab` toolbox provided by the authors<sup>3</sup>. Finally, we provide the results obtained with the GRAPHEM method we recently introduced in (Elvira and Chouzenoux, 2022). GRAPHEM provides a maximum a posteriori estimate  $\hat{\mathbf{A}}$  under an  $\ell_1$  prior, while  $\hat{\mathbf{Q}}$  is empirically set to  $\sigma_{\mathbf{Q}}^2\mathbf{Id}_{N_y}$ , with a finetuned  $\sigma_{\mathbf{Q}} > 0$ . The `Matlab` toolbox provided by the authors<sup>4</sup> is used to produce the results for this method.

All the comparative methods are programmed in the same language, they are initialized using a similar strategy as our DGLASSO method, and similar stopping criteria and hyperparameter tuning approach are employed, for fair comparisons.

### 5.1.3 EVALUATION METRICS

We first evaluate the results of our method, as well as the comparative benchmarks, through quality assessment metrics. Namely, we use the relative mean square error (RMSE) between

3. <http://www-syscom.univ-mlv.fr/benfenat/Software.html>

4. <https://pages.saclay.inria.fr/emilie.chouzenoux/LogicielEN.html>



the ground truth matrices  $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$  and the estimated  $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$  (when available). For instance,

$$\text{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = \frac{\|\mathbf{A}^* - \widehat{\mathbf{A}}\|_{\text{F}}^2}{\|\mathbf{A}^*\|_{\text{F}}^2}. \quad (63)$$

$\text{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}})$  and  $\text{RMSE}(\mathbf{Q}^*, \widehat{\mathbf{Q}})$  are defined in a similar fashion. We also compute area-under-curve (AUC) and F1 score comparing the non-zero entries (that is, the graph edges positions) of the sparse matrices  $(\mathbf{A}^*, \mathbf{P}^*)$  and their estimates  $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$ . A threshold value of  $10^{-10}$  (in absolute value) is used for the detection hypothesis. We furthermore evaluate the ability of the estimated model parameters to actually describe and infer the time series (both observed and hidden states). To that end, we build test time series  $(\mathbf{x}^{\text{test}}, \mathbf{y}^{\text{test}})$ , not seen by the algorithms, constructed by running the ground truth LG-SSM (i.e., with ground truth matrix parameters  $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$ ). We then run KF and RTS algorithms 1 and 2, respectively, using either the ground truth matrices  $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$  or their estimations  $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$ , to build, for every  $k \in \{1, \dots, K\}$ , the predictive distribution means  $(\boldsymbol{\mu}_k^*, \boldsymbol{\nu}_k^*, \boldsymbol{\mu}_k^{s*})$  and  $(\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\nu}}_k, \widehat{\boldsymbol{\mu}}_k^s)$ , respectively.

This allows in particular to compute the cumulative normalized mean squared error (cNMSE) between the predictive distribution means using either the ground truth model matrices or the estimated ones. Namely, we calculate

$$\text{cNMSE}(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}}) = \frac{\sum_{k=1}^K \|\boldsymbol{\nu}_k^* - \widehat{\boldsymbol{\nu}}\|^2}{\sum_{k=1}^K \|\boldsymbol{\nu}_k^*\|^2}, \quad (64)$$

as well as  $\text{cNMSE}(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ , and  $\text{cNMSE}(\boldsymbol{\mu}^{s*}, \widehat{\boldsymbol{\mu}}^s)$ .

Finally, we evaluate the negative logarithm of the marginal likelihood  $\mathcal{L}_{1:K}(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$  as defined in (21), on the test time series.

#### 5.1.4 TUNING THE REGULARIZATION PARAMETERS

Let us first discuss the setting of the DGLASSO hyper-parameters  $(\lambda_A, \lambda_P)$ , accounting for the sparsity priors on  $\mathbf{A}$  and  $\mathbf{P}$ , respectively. To that aim, we ran DGLASSO for 100 values of hyperparameters  $(\lambda_A, \lambda_P)$ , regularly spaced on a log-scale grid between 1 and  $10^2$ , and repeated the experience on 50 randomly generated time series, for dataset A. We display in Figure 4 the averaged values, over the random runs, of several quantitative metrics, as a function of hyperparameters  $(\lambda_A, \lambda_P)$ . We also report in the caption the averaged RMSE scores obtained when running DGLASSO with  $(\lambda_A, \lambda_P) = (0, 0)$  (i.e., MLEM result). As it can be observed, both F1/RMSE for the estimation of the transition matrix  $\mathbf{A}$  are mostly governed by the value of  $\lambda_A$ , while the quality scores for the state noise precision matrix  $\mathbf{P}$  are mostly influenced by  $\lambda_P$ . Note that the RMSE scores on  $\mathbf{Q}$ , not shown here, follow similar evolution than RMSE on  $\mathbf{P}$ . Just inspecting F1/RMSE appears not informative enough to set parameters  $(\lambda_A, \lambda_P)$ , as each metric and each parameter seems to push towards a different goal. The maps of  $\text{cNMSE}(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}})$  and of the marginal likelihood log-loss  $\mathcal{L}_{1:K}$  show very similar behavior. Note that the later is a practically interesting metric, because it does not require the knowledge of the ground truth matrices. On this example, it however appears not discriminative enough. Typically, it stays almost constant for a (too) wide value range of  $(\lambda_A, \lambda_P)$ , which confirms again the ill-posedness of the minimization

problem. The maps for  $\text{cNMSE}(\boldsymbol{\mu}^*, \hat{\boldsymbol{\mu}})$  and  $\text{cNMSE}(\boldsymbol{\mu}^{s*}, \hat{\boldsymbol{\mu}}^s)$  are very similar. Interestingly, the minimization of both these quantities, related to the state mean distributions, seems to provide a meaningful value range for the regularization parameters, narrowing down around values that achieve an optimal compromise between (i) good estimation of the sought matrices, (ii) good estimation of the sparse matrices support, and (iii) good predictive behavior for time series inference by KF/RTS techniques. Same conclusions were reached for the other three datasets. Note additionally that the DGLASSO results outperform for a wide range of  $(\lambda_A, \lambda_P)$  those obtained with MLEM, confirming the validity of the proposed sparsity prior. This will be more deeply examined in an upcoming section. In our forthcoming experiments, except otherwise stated, we fixed  $(\lambda_A, \lambda_P)$  through a rough grid search to minimize  $\text{cNMSE}(\boldsymbol{\mu}^*, \hat{\boldsymbol{\mu}})$  averaged on few (typically 5) runs. On real datasets, we advocate the use of the marginal likelihood on test trials, as a control variate.

### 5.1.5 INFLUENCE OF CONDITIONING NUMBER

We report in Table 1 the results, averaged over 50 realizations of the time series generation. We do not provide the scores for the estimation of  $\mathbf{A}$  (resp.  $(\mathbf{P}, \mathbf{Q})$ ) for GLASSO/rGLASSO (resp. GRAPHEM), as those methods are not designed for such task. The general trend in the results is a slight decrease of the quality of estimation for all methods, when  $c_Q$  increases (i.e., from dataset A to D). This is expected, as an ill-conditioned matrix  $\mathbf{Q}$  complicates the mathematical structure of the likelihood term.

Regarding the estimation of  $\mathbf{A}$ , GRAPHEM method presents the best results for the three considered metrics. DGLASSO is second best, while MLEM follows. As already stated, GLASSO/rGLASSO do not estimate  $\mathbf{A}$  (i.e., they assume this matrix to be zero, that is the process is i.i.d. without any Markov structure). Regarding the estimation of  $(\mathbf{P}, \mathbf{Q})$ , DGLASSO is outperforming the benchmarks in terms of RMSE score. The second best, in terms of RMSE is MLEM. GLASSO and rGLASSO gets very bad RMSE scores, probably because of the model mismatch induced by assuming  $\mathbf{A}$  to be zero. The edge detection performance of DGLASSO are excellent, except in few cases where rGLASSO gets slightly better results. MLEM gets poorer results than DGLASSO in all metrics. In particular it exhibits a bad F1 score, as it does not include sparse prior, and thus does not succeed to eliminate any spurious edges.

Regarding the distribution mean calculation, it is remarkable that DGLASSO outperforms in all examples the benchmarks, which shows that the proposed formulation allows to provide model parameters that are best suited for inference using KF/RTS at test phase. The marginal likelihood log-loss is also minimized with the quantities provided by our method. This could appear as counter-intuitive, as the method is not specifically designed to minimize this loss (while MLEM is). The advantage of DGLASSO is that it accounts for prior knowledge, making the inferred model more robust, and less prone to overfitting, which is translated into better behavior on an independent test time series.

We display in Figure 5 box plots assessing the variability to the RMSE and F1 scores, for both MLEM and DGLASSO methods, on 50 runs on dataset A and dataset D. The RMSE values are in most runs lower (i.e., better) for the proposed method. Both methods are quite stable with respect to the time series generation, as the plots show few outliers. For dataset D, corresponding to a more challenging case with an ill-conditioned matrix  $\mathbf{P}^*$ ,

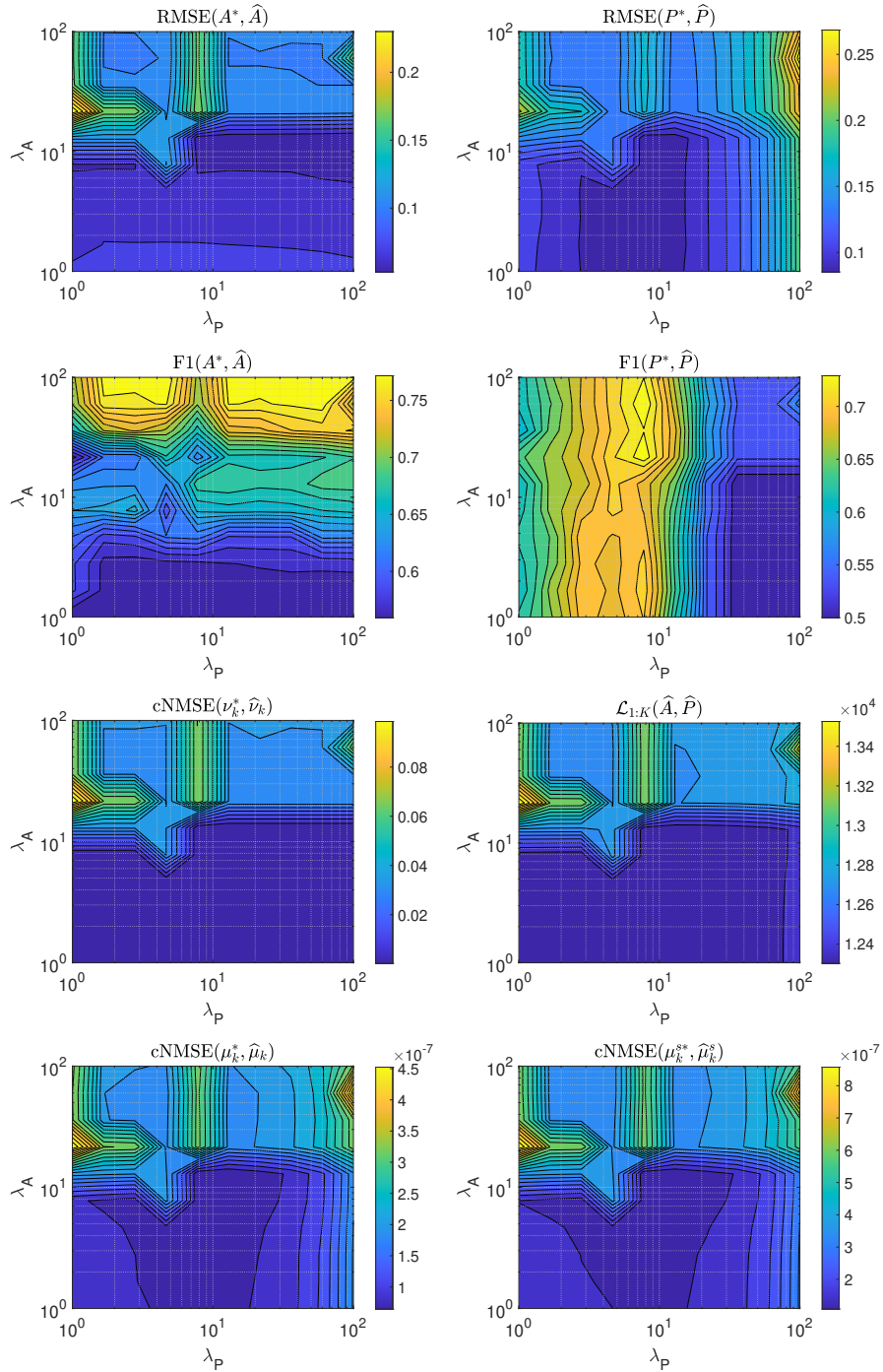


Figure 4: Evolution of RMSE, F1, cNMSE and loss scores on the estimation of  $\mathbf{A}$  (left) and  $\mathbf{P}$  (right) by DGLASSO, as a function of hyperparameters  $(\lambda_A, \lambda_P)$ , for dataset A (averaged on 10 runs). As a comparison, the averaged RMSE scores for  $(\lambda_A, \lambda_P) = (0, 0)$  (i.e., MLEM) on this example were  $(0.077, 0.106)$  for  $(\mathbf{A}, \mathbf{P})$ , respectively.

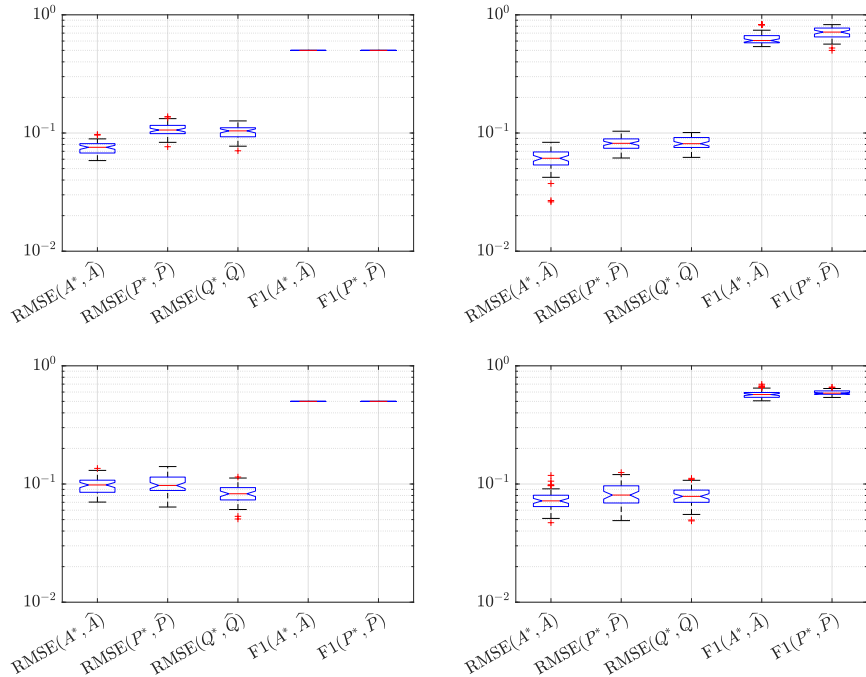


Figure 5: Box plots for the RMSE and F1 scores when running MLEM (left) and DGLASSO (right) on 50 randomly generated LG-SSM time series, for dataset A (top) and dataset D (bottom). DGLASSO outperforms MLEM on all runs, in terms of accuracy metrics (the higher, the better), while its RMSE scores are better (i.e., lower) for DGLASSO for most runs. Dataset D is more challenging, in terms of inference, thus yielding to more spread results for both methods.

the results are slightly more spread, especially for the metrics related to the recovery quality of this matrix. Remark that the F1 scores of MLEM are constant, equal to 0.5. As already pointing out, this is expected as MLEM is not designed to perform an edge detection task.

We refer the reader to Appendix C.3 for extra experiments on the synthetic datasets, assessing the performance of DGLASSO when the sparsity level of  $\mathbf{A}^*$  increases.

### 5.1.6 COMPLEXITY ANALYSIS

We now examine the time complexity of the method, as well as of MLEM, when processing dataset A, for various values of the time series length  $K$ , namely  $K \in \{100, 200, 500, 1000, 2000, 5000\}$  (we recall that our previous experiments were all done with  $K = 1000$ ). We display in Figure 6(left) the computing time in seconds for computing  $(\hat{\mathbf{A}}, \hat{\mathbf{P}})$ , for DGLASSO and MLEM, averaged over 50 realizations. We also display, in Figure 6(middle) for the same experiment, the RMSE between  $\mathbf{A}^*$  and  $\hat{\mathbf{A}}$ , and in Figure 6(right) the metric  $\text{cNMSE}(\boldsymbol{\mu}^*, \hat{\boldsymbol{\mu}})$ . One can notice that our method is slightly more demanding in terms of computational time, but it scales similarly than its non regularized version MLEM. Moreover, as already observed in our previous experiments, DGLASSO outperforms MLEM on both metrics shown here, in all tested values of  $K$ . As expected, the results are better for higher values of  $K$ , at

Table 1: Results for the four considered datasets A to D, with an increasing conditioning number of  $\mathbf{P}^*$  equals to  $\log_{10}(c) \in \{0.1, 0.2, 0.1, 1\}$ , respectively. We evaluate the methods in terms of estimation quality for  $(\mathbf{A}, \mathbf{P}, \mathbf{Q})$ , using either RMSE as defined in (63), and edge detection scores (AUC, F1), as well as in terms of inference quality using cNMSE metrics defined in (64).

	Method	Estimation of $\mathbf{A}$			Estimation of $\mathbf{P}$			Estim. $\mathbf{Q}$	State distrib.		Predictive distrib.	
		RMSE	AUC	F1	RMSE	AUC	F1	RMSE	cNMSE( $\mu^*, \hat{\mu}$ )	cNMSE( $\mu^{S*}, \hat{\mu}^S$ )	cNMSE( $\nu^*, \hat{\nu}$ )	$\mathcal{L}_{1,K}(\mathbf{A}, \hat{\mathbf{P}})$
Dataset A	DGLASSO	0.061	0.843	<b>0.641</b>	<b>0.082</b>	0.778	0.698	<b>0.083</b>	<b><math>6.394 \times 10^{-8}</math></b>	<b><math>1.050 \times 10^{-7}</math></b>	<b><math>2.984 \times 10^{-4}</math></b>	<b>12 307.169</b>
	MLEM	0.076	0.817	0.500	0.105	0.857	0.500	0.102	$1.095 \times 10^{-7}$	$1.803 \times 10^{-7}$	$4.843 \times 10^{-4}$	12 341.205
	GLASSO	NA	NA	NA	0.818	0.804	0.496	1073.510	$4.485 \times 10^{-6}$	$7.180 \times 10^{-6}$	1.000	28 459.294
	rGLASSO	NA	NA	NA	0.764	<b>0.924</b>	0.598	31.689	$2.826 \times 10^{-6}$	$5.492 \times 10^{-6}$	1.000	22 957.693
	GRAPHEM	<b>0.045</b>	<b>0.895</b>	<b>0.847</b>	NA	NA	NA	NA	$4.364 \times 10^{-6}$	$6.944 \times 10^{-6}$	$2.980 \times 10^{-4}$	29 035.030
Dataset B	DGLASSO	0.068	0.833	0.603	<b>0.070</b>	0.893	<b>0.835</b>	<b>0.071</b>	<b><math>7.490 \times 10^{-8}</math></b>	<b><math>1.236 \times 10^{-7}</math></b>	<b><math>3.281 \times 10^{-4}</math></b>	<b>11 806.744</b>
	MLEM	0.080	0.815	0.500	0.106	0.898	0.500	0.100	$1.299 \times 10^{-7}$	$2.133 \times 10^{-7}$	$4.619 \times 10^{-4}$	11 833.448
	GLASSO	NA	NA	NA	0.827	0.826	0.505	341.873	$5.069 \times 10^{-6}$	$8.072 \times 10^{-6}$	1.000	27 744.964
	rGLASSO	NA	NA	NA	0.734	<b>0.930</b>	0.608	33.896	$3.215 \times 10^{-6}$	$6.187 \times 10^{-6}$	1.000	22 530.036
	GRAPHEM	<b>0.047</b>	<b>0.893</b>	<b>0.848</b>	NA	NA	NA	NA	$5.158 \times 10^{-6}$	$8.036 \times 10^{-6}$	$2.912 \times 10^{-4}$	29 031.412
Dataset C	DGLASSO	0.070	0.829	0.581	<b>0.090</b>	0.954	<b>0.830</b>	<b>0.078</b>	<b><math>1.896 \times 10^{-7}</math></b>	<b><math>2.994 \times 10^{-7}</math></b>	<b><math>3.956 \times 10^{-4}</math></b>	<b>10 311.104</b>
	MLEM	0.081	0.810	0.500	0.097	<b>0.974</b>	0.500	0.094	$2.583 \times 10^{-7}$	$4.180 \times 10^{-7}$	$5.053 \times 10^{-4}$	10 326.410
	GLASSO	NA	NA	NA	0.901	0.805	0.489	$3.926 \times 10^{17}$	0.012	0.012	1.000	26 634.892
	rGLASSO	NA	NA	NA	0.805	0.928	0.614	29.530	$7.195 \times 10^{-6}$	$1.320 \times 10^{-5}$	1.000	21 322.247
	GRAPHEM	<b>0.049</b>	<b>0.892</b>	<b>0.857</b>	NA	NA	NA	NA	$1.055 \times 10^{-5}$	$1.641 \times 10^{-5}$	$3.912 \times 10^{-4}$	29 023.369
Dataset D	DGLASSO	0.073	0.835	0.575	<b>0.083</b>	<b>1.000</b>	0.598	<b>0.080</b>	<b><math>5.127 \times 10^{-7}</math></b>	<b><math>8.243 \times 10^{-7}</math></b>	<b><math>3.373 \times 10^{-4}</math></b>	<b>7 911.943</b>
	MLEM	0.098	0.808	0.500	0.095	<b>1.000</b>	0.500	0.084	$6.296 \times 10^{-7}$	$1.027 \times 10^{-6}$	$4.219 \times 10^{-4}$	7 923.850
	GLASSO	NA	NA	NA	0.964	0.941	0.550	187.823	$2.348 \times 10^{-5}$	$3.701 \times 10^{-5}$	1.000	23 684.178
	rGLASSO	NA	NA	NA	0.882	0.956	<b>0.645</b>	28.703	$1.886 \times 10^{-5}$	$3.239 \times 10^{-5}$	1.000	20 100.491
	GRAPHEM	<b>0.061</b>	<b>0.892</b>	<b>0.864</b>	NA	NA	NA	NA	$2.503 \times 10^{-5}$	$3.839 \times 10^{-5}$	$3.743 \times 10^{-4}$	29 016.321

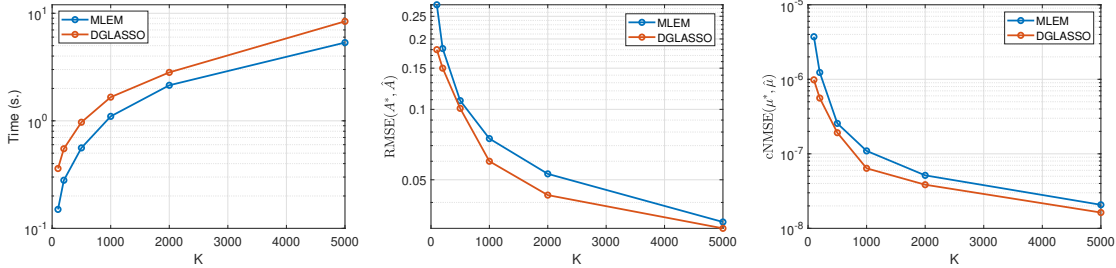


Figure 6: Evolution of the complexity time (left),  $\text{RMSE}(\mathbf{A}^*, \hat{\mathbf{A}})$  (middle) and  $\text{cNMSE}(\mu^*, \hat{\mu})$  (right) metrics, as a function of the time series length  $K$ , for experiments on dataset A averaged over 50 runs.

the price of an increased computational time. Interestingly, the regularization still yields improved results for very large  $K$ .

## 5.2 Weather data

### 5.2.1 EXPERIMENTAL SETTINGS

We now evaluate our method on realistic graph datasets arising from causal discovery studies in the field of weather variability tracking. Specifically, we consider two sets of 200 sparse matrices  $\mathbf{A}^* \in \mathbb{R}^{N_x}$ , with  $N_x = 5$  or 10 respectively, representing the ground truth causal graphs used to produce WEATH datasets in the Neurips 2019 data challenge (Runge et al.,

2020)<sup>5</sup>. For each  $\mathbf{A}^*$ , we create times series  $(\mathbf{x}_k, \mathbf{y}_k)_{k=1}^K$  using (2)-(3), with  $K = 10^3$ ,  $\mathbf{H}^* = \mathbf{Id}$  (i.e.,  $N_y = N_x$ ), and  $(\sigma_{\mathbf{R}}, \sigma_0) = (10^{-1}, 10^{-4})$ . We set  $\mathbf{Q}^*$  as a block diagonal matrix of  $J$  blocks with dimensions  $B_{j_{1 \leq j \leq J}}$ , with  $\sum_{j=1}^J B_j = N_x$ . Here, we consider two settings for the conditioning of  $\mathbf{Q}^*$ , namely one with a condition number close to one (i.e.,  $\mathbf{Q}^*$  is close to identity matrix) and another with high condition number. The main characteristics of the datasets and their names are summarized in Table 2.

We evaluate our results in terms on quality assessment metrics of the estimated  $\hat{\mathbf{A}}$  when compared to its ground truth  $\mathbf{A}^*$ , as this is the quantity of interest for these datasets. We compute  $\text{RMSE}(\hat{\mathbf{A}}, \mathbf{A}^*)$ , as well as the precision, recall, specificity, accuracy, and F1 score for detecting the non-zero entries of the transition matrix (that is, the graph edges positions). A threshold value of  $10^{-10}$  on the absolute entries is used for the detection hypothesis.

As for comparison, we also provide the results obtained with MLEM and GRAPHEM approaches, as in our previous experiments. The same stopping criterion than in our previous set of experiments is used. The hyperparameters are finetuned using the strategy depicted on the previous section. We used rough grids for the finetuning and keep the parameters fixed for all the dataset, to avoid any overfitting behavior. In addition to these EM-based methods, we provide comparisons with two Granger-causality approaches for graphical modeling, namely pairwise Granger Causality (PGC) and conditional Granger Causality (CGC). Those approaches are based on sparse vector autoregressive (VAR) models. We allow the order of the VAR process to be up to  $p = 11$ , CGC is run with a maximum distance of 5 causal links, and in each experiment we display the best results (in F1 score) for the statistical tests performed with the significance level  $\alpha \in \{0.01, 0.05, 0.1\}$  (see more details in (Luengo et al., 2019)). As PGC and CGC do not provide a weighted graph estimation, no RMSE score is computed in those cases.

### 5.2.2 RESULTS

We summarize our results in Table 3. We also report the averaged inference time for all methods. Remarkably, the proposed method outperforms all benchmarks in all the considered metrics, related to the inference of the graph weights (RMSE metric) and the edge detection task (binary classification metrics). As expected, the result quality tends to slightly degrade when a more complicated structure is assumed for  $\mathbf{Q}^*$  (see, for instance, WeathN5a versus WeathN5b), and when the dataset size increases (see, for instance, WeathN5a versus WeathN10a). MLEM has very poor edge detection metrics, since it does not exploit any sparsity prior on the sought matrix. GRAPHEM has better behavior, but, in these datasets, it stays way behind the quality of DGLASSO, by all metrics (which was not the case for the synthetic data). This shows that our method really makes the change when dealing with complex graphs and correlated noise in the observations. In terms of computational time, the proposed method has similar complexity than the two other EM-based approaches. The results of CGC and PGC are satisfactory in the first two examples although the binary metrics are far from the performance of DGLASSO. PGC still gives meaningful results in the last two examples, but CGC has a low performance due to a high number of false negatives (i.e., many existing links are not discovered). We also note that while PGC and

5. <https://causeme.uv.es/static/datasets/TestWEATH/>

Dataset	$N_x = N_y$	$(B_j)_{1 \leq j \leq J}$	$\log_{10}(\text{cond}(\mathbf{Q}^*))$
WeathN5a	5	(2, 3)	0.1
WeathN5b	5	(2, 3)	1
WeathN10a	10	(5, 5)	0.1
WeathN10b	10	(5, 5)	1

Table 2: Details about the datasets. Each dataset is associated with 200 examples for matrix  $\mathbf{A}^*$ .

Table 3: Results for climate datasets along with computing times. All the metrics are averaged over the 200 examples of the dataset.

	method	RMSE	accur.	prec.	recall	spec.	F1	Time (s.)
WeathN5a	DGLASSO	<b>0.108</b>	<b>0.937</b>	0.894	0.998	0.894	<b>0.937</b>	0.608
	MLEM	0.140	0.413	0.413	<b>1.000</b>	0.000	0.584	0.596
	GRAPHEM	0.127	0.703	0.595	<b>1.000</b>	0.496	0.742	0.606
	PGC	-	0.772	<b>0.902</b>	0.515	<b>0.953</b>	0.652	0.019
	CGC	-	0.672	0.828	0.285	0.945	0.415	0.026
WeathN5b	DGLASSO	<b>0.166</b>	<b>0.773</b>	0.668	0.992	0.619	<b>0.788</b>	0.630
	MLEM	0.197	0.413	0.413	<b>1.000</b>	0.000	0.584	0.376
	GRAPHEM	0.186	0.629	0.536	<b>1.000</b>	0.368	0.694	0.470
	PGC	-	0.675	<b>0.677</b>	0.469	0.819	0.544	0.017
	CGC	-	0.634	0.659	0.263	<b>0.895</b>	0.369	0.023
WeathN10a	DGLASSO	<b>0.202</b>	<b>0.948</b>	<b>0.898</b>	0.925	0.954	<b>0.890</b>	1.363
	MLEM	0.264	0.219	0.219	<b>1.000</b>	0.000	0.359	0.834
	GRAPHEM	0.224	0.511	0.311	1.000	0.374	0.473	1.445
	PGC	-	0.879	0.904	0.504	<b>0.983</b>	0.644	0.232
	CGC	-	0.773	0.539	0.211	0.932	0.278	0.358
WeathN10b	DGLASSO	<b>0.192</b>	<b>0.866</b>	<b>0.633</b>	0.994	0.829	<b>0.769</b>	0.557
	MLEM	0.342	0.219	0.219	<b>1.000</b>	0.000	0.359	0.989
	GRAPHEM	0.219	0.855	0.620	0.994	0.816	0.757	0.655
	PGC	-	0.799	0.558	0.473	0.890	0.506	0.154
	CGC	-	0.750	0.407	0.218	<b>0.900</b>	0.265	0.178

CGC have significantly lower running times for the case with  $N_y = 5$ , the computational cost for  $N_y = 10$  is closer to DGLASSO while the performance gap is higher. Thus, in this examples we show that as the dimension grows, the computational cost of all methods is comparable while the proposed method has clearly a better performance. We also display some examples of inferred graphs in Figures 7 and 8. We can observe that DGLASSO is able to capture in a very accurate manner the structure of the graphs, despite the wide variability of the dataset. MLEM and GRAPHEM capture the main edges, but their graphs are perturbed by several spurious edges, which shows how important it is to adopt a joint graph modeling, with sparsity priors on each.

## 6. Conclusion

This paper proposes a joint graphical modeling approach that incorporates a graphical perspective on the dynamics of the hidden state of a linear Gaussian state-space model.

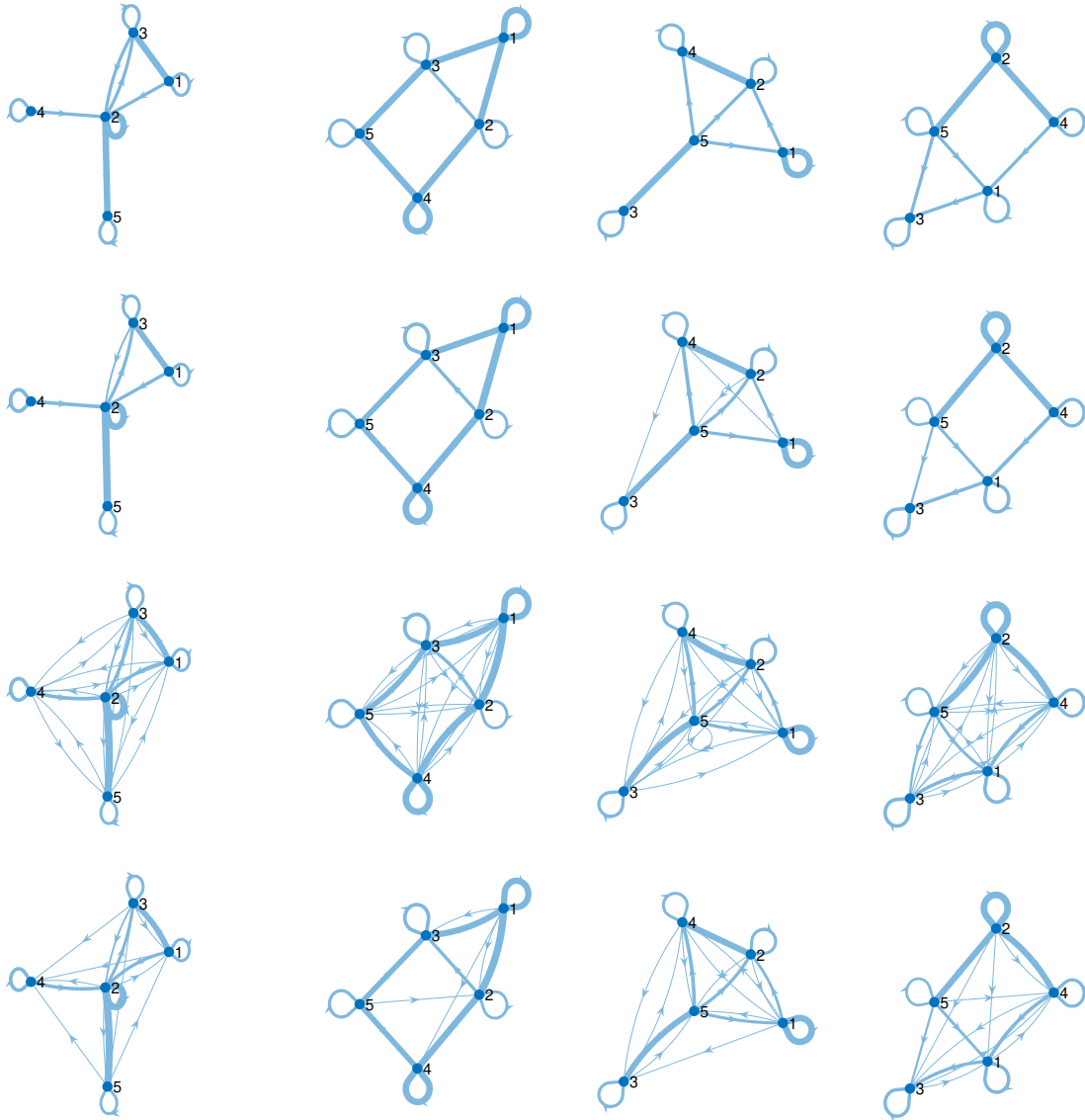


Figure 7: Graph inference results on 4 examples extracted from the dataset *WeathN5a*. From top to bottom: Original graph representation of  $\mathbf{A}^*$ , and of its estimation  $\hat{\mathbf{A}}$ , using DGLASSO, MLEM, GRAPHM, respectively.

In particular, we propose a joint approach that considers a sparse undirected graph as a prior on the precision matrix of the hidden state noise and a sparse directed graph for the transition matrix that models the state dynamics. By bridging the gap between the static graphical Lasso model and the dynamic state-space model, we provide a novel comprehensive framework for interpretable inference with time-series data. The presented inference method, based on an efficient block alternating majorization-minimization algorithm, enables simultaneous estimation of both graphs and the construction of the filtering/smoothing



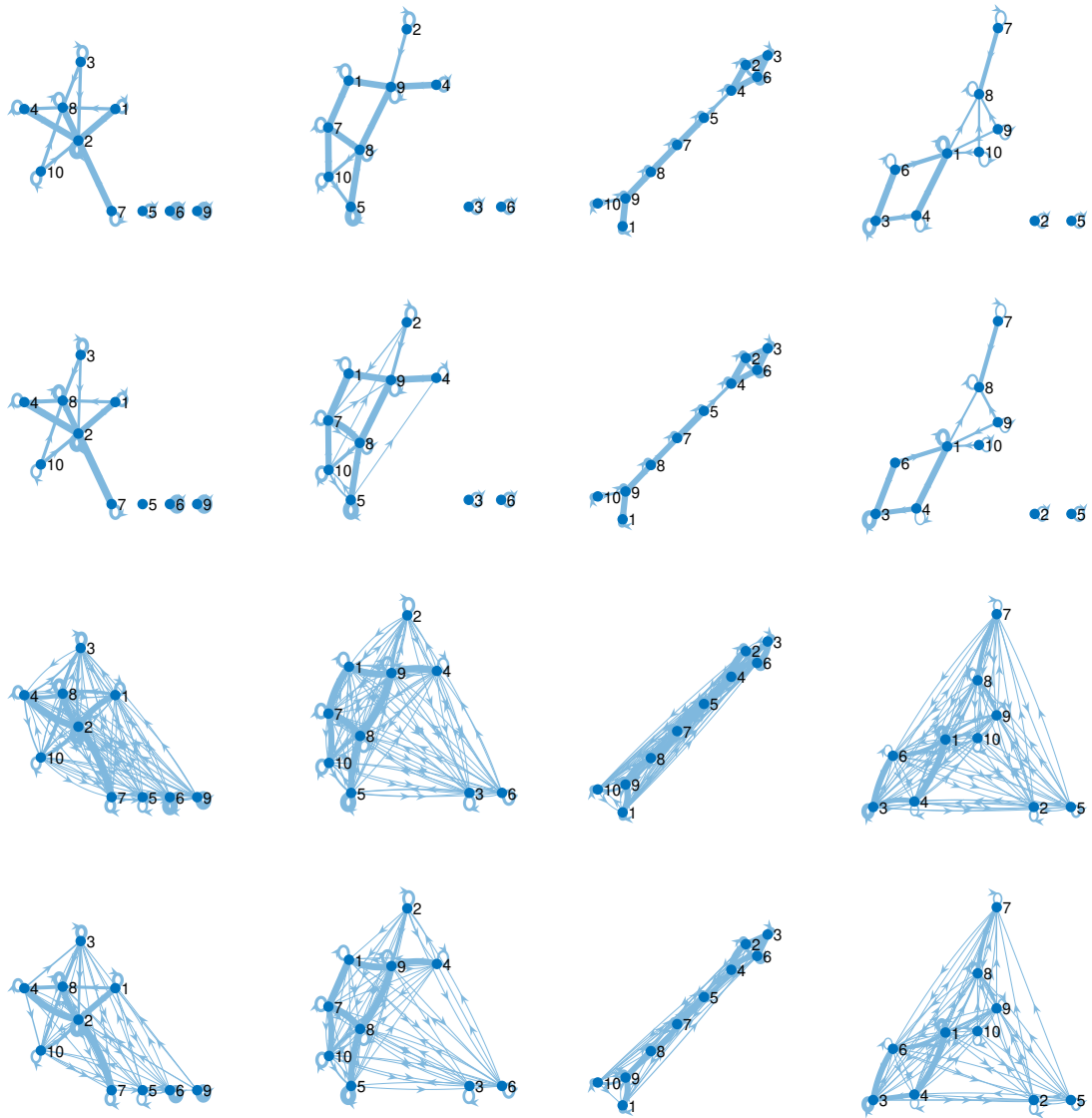


Figure 8: Graph inference results on 4 examples extracted from the dataset *WeathN10a*. From top to bottom: Original graph representation of  $\mathbf{A}^*$ , and of its estimation  $\hat{\mathbf{A}}$ , using DGLASSO, MLEM, GRAPHM, respectively.

distribution for the time series. The established convergence of our algorithm, departing from recent nonlinear analysis tools, enhances the reliability and practicality of our approach. Through extensive experimental validation on synthetic and real weather variability data, we have demonstrated the effectiveness and potential of our proposed model and inference algorithm. The results showcase its ability to uncover meaningful insights from time-series data, contributing not only to better forecasting performance but also to a better understanding of complex phenomena in various scientific and engineering domains. Fu-

ture research can further explore and extend the presented framework to tackle even more challenging state-space models, and more complex graphical structures.

### **Acknowledgments**

É.C. acknowledges support from the European Research Council Starting Grant MAJORIS ERC-2019-STG-850925. The work of V. E. is supported by the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01), the Leverhulme Research Fellowship (RF-2021-593), and by ARL/ARO under grant W911NF-22-1-0235.

The authors thank Gustau Camps-Valls and his team, for providing the ground truth data for (Runge et al., 2020).

## A. Proof of Theorem 1

For any initial state  $\mathbf{x}_{0:K}$ , with non zero probability, the neg-log-likelihood  $(\mathbf{A}, \mathbf{P}) \rightarrow (\mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = -\log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P}))$  reads, according to Bayes' rule,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \\ \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = -\log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P}) + \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}). \quad (65)$$

According to Eqs. (2)-(3)

$$\log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P}) = \log p(\mathbf{x}_0) + \sum_{k=1}^K \log p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{A}, \mathbf{P}) + \sum_{k=1}^K \log p(\mathbf{y}_k|\mathbf{x}_k). \quad (66)$$

Moreover, using again Eq. (2) and the statistical model of the state noise,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad (67)$$

$$\sum_{k=1}^K \log p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{A}, \mathbf{P}) = -\frac{1}{2} \sum_{k=1}^K \left( (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) - \log \det(2\pi\mathbf{P}) \right), \quad (68)$$

$$= -\frac{1}{2} \sum_{k=1}^K \left( (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \right) + \frac{K}{2} \log \det(2\pi\mathbf{P}), \quad (69)$$

which concludes the first part of the proof.

Let us now consider some  $\tilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$  and  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ . We start by recalling some known results arising from the EM methodology (Dempster et al., 1977; Wu, 1983), that we specify here for our context for the sake of clarity. First, we notice that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \\ \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = - \int \log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P}) p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) d\mathbf{x}_{0:K}, \quad (70)$$

since  $\log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P})$  is constant with respect to the integration variable, and the distribution  $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}})$  integrates to one. Then, according to (65) and (70), the expectation of the neg-log-likelihood multiplied by  $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}})$  over all possible values of the unknown state reads:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = - \overbrace{\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P}) d\mathbf{x}_{0:K}}^{q(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}})} \\ + \overbrace{\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}) d\mathbf{x}_{0:K}}^{h(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}})}. \quad (71)$$

In particular, for  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ ,  $\mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = q(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + h(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}})$  so that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) - \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = q(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - q(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \\ + h(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - h(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}). \quad (72)$$

Using Gibbs's inequality,  $h(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \leq h(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ , with equality at  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ . Thus, using (72), that is

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) \leq q(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - q(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}), \quad (73)$$

where equality holds at  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ . Notice that, for any function reading

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = q(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \text{ct}_{/\mathbf{A}, \mathbf{P}}, \quad (74)$$

we obviously still have

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) \leq \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - \mathcal{Q}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}), \quad (75)$$

where equality hereagain holds at  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ . Using (75), (20), (22) and noticing that, for every  $(\theta_A, \theta_P) > 0$ ,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \frac{1}{2\theta_A} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 \geq 0, \quad \frac{1}{2\theta_P} \|\mathbf{P} - \tilde{\mathbf{P}}\|_F^2 \geq 0, \quad (76)$$

with equality holding at  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ , we deduce the desired majorizing property

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) \leq \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}) - \mathcal{Q}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \\ + \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P} - \tilde{\mathbf{P}}\|_F^2, \quad (77)$$

with equality holding at  $(\mathbf{A}, \mathbf{P}) = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ . The remaining of the proof amounts to expliciting the expression for  $(\mathbf{A}, \mathbf{P}) \rightarrow \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}})$  satisfying (74) with function  $q$  defined as in (71):

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \\ q(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = - \int p(\mathbf{x}_{0:K} | \mathbf{y}_{1:K}, \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K} | \mathbf{A}, \mathbf{P}) d\mathbf{x}_{0:K}. \quad (78)$$

Following (Sarkka, 2013, Theorem 12.4) (see also an alternative proof in (Elvira and Chouzenoux, 2022, Sec.III-B)), (71)-(74) hold for

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \tilde{\mathbf{A}}, \tilde{\mathbf{P}}) = \frac{K}{2} \text{tr}(\mathbf{P}(\Psi - \Delta \mathbf{A}^\top - \mathbf{A} \Delta^\top + \mathbf{A} \Phi \mathbf{A}^\top)) - \frac{K}{2} \log \det(2\pi \mathbf{P}), \quad (79)$$

and

$$\Psi = \frac{1}{K} \sum_{k=1}^K \left( \Sigma_k^s + \mu_k^s (\mu_k^s)^\top \right), \quad (80)$$

$$\Delta = \frac{1}{K} \sum_{k=1}^K \left( \Sigma_k^s \mathbf{G}_{k-1}^\top + \mu_k^s (\mu_{k-1}^s)^\top \right), \quad (81)$$

$$\Phi = \frac{1}{K} \sum_{k=1}^K \left( \Sigma_{k-1}^s + \mu_{k-1}^s (\mu_{k-1}^s)^\top \right). \quad (82)$$

Hereabove,  $(\mu_k^s, \Sigma_k^s)_{0 \leq k \leq K-1}$  denotes the mean and covariance of the smoothing distribution obtained when running Algs. 1-2 using  $(\tilde{\mathbf{A}}, \tilde{\mathbf{P}})$ . Moreover, the matrix

$$\mathbf{G}_k = \Sigma_k \tilde{\mathbf{A}}^\top \left( \tilde{\mathbf{A}} \Sigma_k \tilde{\mathbf{A}}^\top + \tilde{\mathbf{P}}^{-1} \right) \quad (83)$$

is defined in Alg. 2. This concludes the proof.

## B. Proximal algorithms to solve the inner steps

We present Algorithms 4 and 5, that are proximal splitting algorithms to solve, respectively, the inner problems (32) and (33). Specifically, both algorithms are special instances of the Dykstra-like splitting algorithm from (Bauschke and Combettes, 2008) (see also (Combettes and Pesquet, 2011, Sec.5)), for the minimization of the sum of two convex but non-differentiable functions. Sequence  $(\mathbf{A}_n)_{n \in \mathbb{N}}$  (resp.  $(\mathbf{P}_n)_{n \in \mathbb{N}}$ ) is guaranteed to converge to the solution of problem (32) (resp. (33)). The proximity steps involved in Algorithms 4 and 5 have closed form expressions that can be found for instance in the reference book (Bauschke and Combettes, 2017). We explicit them hereafter for the sake of completeness.

**Proximity of  $\ell_1$ .** Let  $\gamma > 0$  and  $\tilde{\mathbf{V}} \in \mathbb{R}^{N_x \times N_x}$ . Then,

$$\text{prox}_{\gamma \ell_1}(\tilde{\mathbf{V}}) \quad (84)$$

$$= \left( \text{sign}(\tilde{V}(n, \ell)) \max(0, \tilde{V}(n, \ell) - \gamma) \right)_{1 \leq n, \ell \leq N_x}, \quad (85)$$

which amounts to applying the soft thresholding operator with weight  $\gamma$  on every entry of the matrix input  $\tilde{\mathbf{V}}$ .

**Proximity of quadratic term.** Let  $\gamma > 0$  and  $\tilde{\mathbf{W}} \in \mathbb{R}^{N_x \times N_x}$ . Then, by definition,

$$\hat{\mathbf{Z}} = \text{prox}_{\mathbf{W} \rightarrow \gamma \text{tr}(-\tilde{\mathbf{P}}\Delta\mathbf{W} - \tilde{\mathbf{P}}\mathbf{W}\Delta^\top + \tilde{\mathbf{P}}\mathbf{W}\Psi\mathbf{W}^\top)}(\tilde{\mathbf{W}}) \quad (86)$$

$$= \underset{\mathbf{W} \in \mathbb{R}^{N_x \times N_x}}{\text{argmin}} \gamma \text{tr} \left( -\tilde{\mathbf{P}}\Delta\mathbf{W}^\top - \tilde{\mathbf{P}}\mathbf{W}\Delta^\top + \tilde{\mathbf{P}}\mathbf{W}\Psi\mathbf{W}^\top \right) + \frac{1}{2} \|\mathbf{W} - \tilde{\mathbf{W}}\|_{\mathbb{F}}^2. \quad (87)$$

The optimality condition for (87) gives

$$-\gamma \tilde{\mathbf{P}}\Delta - \gamma \Delta^\top \tilde{\mathbf{P}} + \gamma \tilde{\mathbf{P}}\hat{\mathbf{Z}}\Psi + \gamma \tilde{\mathbf{P}}^\top \hat{\mathbf{Z}}\Psi^\top + \hat{\mathbf{Z}} - \tilde{\mathbf{W}} = \mathbf{0}. \quad (88)$$

Since  $\tilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ , and  $\Psi \in \mathcal{S}_{N_x}$  (by construction), we have equivalently,

$$-\gamma\Delta - \gamma\tilde{\mathbf{P}}^{-1}\Delta^\top\tilde{\mathbf{P}} + 2\gamma\hat{\mathbf{Z}}\Psi + \tilde{\mathbf{P}}^{-1}\hat{\mathbf{Z}} - \tilde{\mathbf{P}}^{-1}\tilde{\mathbf{W}} = \mathbf{0}. \quad (89)$$

Thus,

$$\hat{\mathbf{Z}} = \text{lyapunov}\left(\tilde{\mathbf{P}}^{-1}, 2\gamma\Psi, \gamma(\Delta + \tilde{\mathbf{P}}^{-1}\Delta^\top\tilde{\mathbf{P}}) + \tilde{\mathbf{P}}^{-1}\tilde{\mathbf{W}}\right), \quad (90)$$

where  $\mathbf{A} = \text{lyapunov}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$  provides the solution to the Lyapunov equation  $\mathbf{X}\mathbf{A} + \mathbf{A}\mathbf{Y} = \mathbf{Z}$ .

**Proximity of log-determinant term.** Let  $\gamma > 0$  and  $\tilde{\mathbf{W}} \in \mathcal{S}_{N_x}^{++}$ . Then, by definition,

$$\hat{\mathbf{Z}} = \text{prox}_{\mathbf{W} \rightarrow \gamma(-\log \det(\mathbf{W}) + \text{tr}(\mathbf{W}\mathbf{\Pi}))}\left(\tilde{\mathbf{W}}\right) \quad (91)$$

$$= \underset{\mathbf{W} \in \mathcal{S}_{N_x}}{\text{argmin}} \quad -\gamma \log \det(\mathbf{W}) + \gamma \text{tr}(\mathbf{W}\mathbf{\Pi}) + \frac{1}{2}\|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2. \quad (92)$$

Using (Bauschke and Combettes, 2017, Chap. 24) (see also (Benfenati et al., 2020)), for every  $\alpha > 0$ ,

$$\hat{\mathbf{Z}} = \mathbf{U}\text{Diag}\left(\left(\frac{1}{2}(\omega(n) + \sqrt{\omega(n)^2 + 4\gamma})\right)_{1 \leq n \leq N_x}\right)\mathbf{U}^\top \quad (93)$$

where  $\boldsymbol{\omega} = (\omega(n))_{1 \leq n \leq N_x}$  gathers the eigenvalues of  $\tilde{\mathbf{W}} - \gamma\mathbf{\Pi} \in \mathcal{S}_{N_x}$  and  $\mathbf{U} \in \mathbb{R}^{N_x \times N_x}$  is an orthogonal matrix such that

$$\tilde{\mathbf{W}} - \gamma\mathbf{\Pi} = \mathbf{U}\text{Diag}(\boldsymbol{\omega})\mathbf{U}^\top. \quad (94)$$

## C. Additional experiments

We present here additional experimental results completing Section 5.1.

### C.1 Maximum likelihood calculation

We compare MLEM approach from (Shumway and Stoffer, 1982) with the sensitivity equation approach from (Segal and Weinstein, 1989), for computing maximum likelihood estimates of the sought matrices. For the latter, we use the gradient expressions from (Nagakura, 2021), and performed the minimization using a Quasi-Newton routine from the OPTIMIZATION TOOLBOX from MATLAB, and used the notation MLQN. We ran both MLEM and MLQN methods on the four synthetic datasets A to D. Figure 9 displays the evolution of the loss function along iterations for both MLEM and MLQN algorithms. One can notice that both reach a similar value at convergence. Moreover, MLEM requires very few iterations (typically, less than 10) to stabilize, while MLQN displays a slower convergence profile despite the quasi-Newton (i.e., second-order) acceleration strategy. This might be due to the ill conditioning of the minimization problem. In terms of time complexity, both methods are comparable, requiring a full pass on the time series, through a Kalman update, at each iteration. Given these results, we decided to keep only MLEM as a comparative benchmark for our experiments.

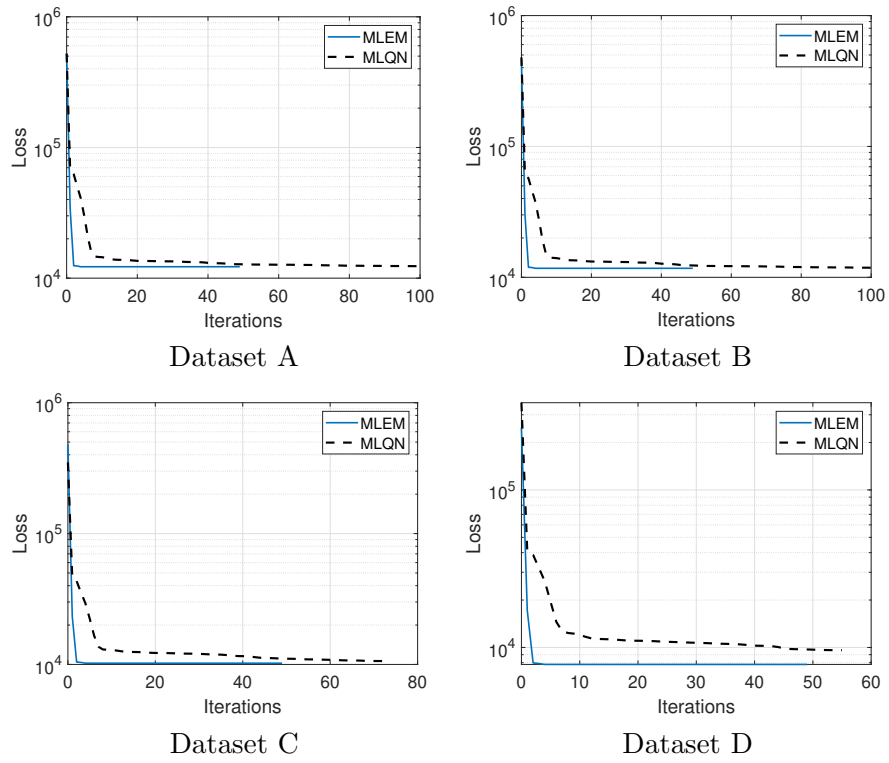


Figure 9: Evolution of  $\mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{Q}^{(i)})$  along iterations using either MLEM or MLQN approach.

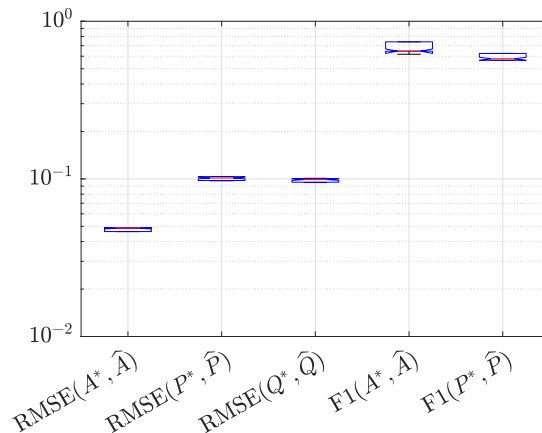


Figure 10: Box plots for the RMSE and F1 scores for retrieving  $(\mathbf{A}, \mathbf{P}, \mathbf{Q})$  matrices, when running DGLASSO on one single LG-SSM time series using dataset A, and 50 random initializations  $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)})$ . Noticeably, low variability is observed for all metrics.

### C.2 Robustness to initialization

DGLASSO algorithm amounts to minimizing a non-convex loss function. As such, its results might be sensitive to the initialization of the algorithm. To evaluate this aspect, we consider the computation of  $(\hat{\mathbf{A}}, \hat{\mathbf{P}})$  given the observation of a single time series generated by the ground truth LG-SSM, when using 50 different initializations of DGLASSO algorithm. To do so, we use the same initialization strategy as discussed above, now with  $(a, p)$  randomly selected as  $p \sim \mathcal{U}([0, 1])$  and  $a \sim \mathcal{U}([0, 1])$ . Figure 10 displays the box plots for RMSE and F1 scores obtained for dataset A. One can notice that the box plots are very concentrated, showing a good robustness of the method to its initialization, with the wider spreading observed for the F1 score on  $\mathbf{A}$ . Similar behavior was observed for the other three datasets.

### C.3 Influence of sparsity level

We evaluate here the performance of DGLASSO, as well as the benchmarks, when varying the sparsity level of the ground truth matrices. To do so, we perform slight changes in the dataset, to vary the sparsity pattern of the matrices (i.e., the edge structure of the graphs). First, we modify the ground truth matrix  $\mathbf{A}^*$  by keeping  $s_A \in \{27, 15, 10, 5\}$  entries of it, within the 27 block diagonal ones, to be non-zero, the others being set to zero. We then rescaled the matrix to keep a spectral norm equals 0.99. Matrix  $\mathbf{Q}^*$  is taken from dataset A. The results are reported in Table 4.

As we can observe, the performance of MLEM, in terms of RMSE and F1 score, drop dramatically when the sparsity level on  $\mathbf{A}$  increases. This is expected as this approach does not promote any sparsity prior on matrix  $\mathbf{A}$ . The best AUC are either obtained by MLEM, DGLASSO or rGLASSO (for  $\mathbf{P}$ ), depending on the test cases. GLASSO/rGLASSO metrics slightly improve when  $\mathbf{A}^*$  gets sparser, which is expected, as their assumption of a zero transition matrix gets more realistic. Hereagain, DGLASSO outperforms the benchmarks in most cases.



## References

- C. Alippi and D. Zambon. Graph Kalman filters. Technical report, 2023. <https://arxiv.org/pdf/2303.12021.pdf>.
- C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-lojasiewicz inequality. *Mathematics of Operation Research*, 35(2):438–457, 2010.
- H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting. *Mathematical Programming*, 137(1):91–129, Feb. 2013.
- F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, Jan. 2012.
- F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *IEEE Transactions on Signal Processing*, 52(8):2189–2199, Aug. 2004. doi: 10.1109/TSP.2004.831032.
- D. Barber and A. T. Cemgil. Graphical models for time-series. *IEEE Signal Processing Magazine*, 27(6):18–28, Nov 2010. doi: 10.1109/MSP.2010.938028.
- H. H. Bauschke and P. L. Combettes. A Dykstra-like algorithm for two monotone operators. *Pacific Journal of Optimization*, 4:383–391, 2008.
- H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017. doi: 10.1007/978-3-319-48311-5.
- E. Belilovsky, K. Kastner, G. Varoquaux, and M. B. Blaschko. Learning to discover sparse graphical models. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70, pages 440–448, 2017.
- A. Benfenati, E. Chouzenoux, L. Duval, J.-C. Pesquet, and A. Pirayre. A review on graph optimization and algorithmic frameworks. Technical report, 2018. <https://hal.science/hal-01901499/document>.
- A. Benfenati, E. Chouzenoux, and J.-C. Pesquet. Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation. *Signal Processing*, 169:107417, Apr. 2020.
- J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, Aug. 2014.

- L. Brendan and J. Tenenbaum. Discovering structure by learning sparse graphs. In *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society (CogSci 2010)*, pages 778–784, Portland, Oregon, USA, 11-14 Aug. 2010.
- M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(61), 2010.
- P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.
- O. Cappe, E. Moulines, and T. Ridden. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer New York, NY, 1st edition, 2005.
- V. Chandrasekaran, P. Parrilo, and A. Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.
- C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23(4):1495–1518, June 2007.
- X. Chen, H. Wen, and Y. Li. Differentiable particle filters through conditional normalizing flow. In *Proceedings of the IEEE 24th International Conference on Information Fusion (FUSION 2021)*, pages 1–6, 2021.
- A. Cherni, E. Chouzenoux, L. Duval, and J.-C. Pesquet. SPOQ lp-over-lq regularization for sparse signal recovery applied to mass spectrometry. *IEEE Transactions on Signal Processing*, 68:6070–6084, 2020.
- N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. Smc2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(3):397–426, 2013.
- 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)*, pages 5840–5844, 4–8 May 2020.
- E. Chouzenoux and V. Elvira. Graphit: Iterative reweighted l1 algorithm for sparse graph inference in state-space models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, Rhodes Island, Greece, 4-10 June 2023.
- E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014.
- E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward-backward algorithm. *Journal of Global Optimization*, 66(3):457–485, 2016.
- E. Chouzenoux, T. T.-K. Lau, and J.-C. Pesquet. Optimal multivariate Gaussian fitting with applications to PSF modeling in two-photon microscopy imaging. *Journal of Mathematical Imaging and Vision*, 61(7):1037–1050, 2019.

- P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.
- A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, page 2100–2111, Jul 2021.
- B. Cox and V. Elvira. Parameter estimation in sparse linear-Gaussian state-space models via reversible jump Markov Chain Monte Carlo. In *Proceedings of the 30th European Signal Processing Conference (EUSIPCO 2022)*, pages 797–801, Belgrade, Serbia, 29 Aug.-2 Sep. 2022.
- B. Cox and V. Elvira. Sparse Bayesian estimation of parameters in linear-Gaussian state-space models. *IEEE Transactions on Signal Processing (to appear in)*, 2023.
- D. Crisan and J. Miguez. Nested particle filters for online parameter estimation in discrete-time state-space markov models. 2018.
- 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)*, 39(1):1–38, 1977.
- P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.
- A. Doucet, A. M. Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.
- M. Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1):233–268, Jun. 2012. ISSN 1432-2064. doi: 10.1007/s00440-011-0345-8. URL <https://doi.org/10.1007/s00440-011-0345-8>.
- V. Elvira and E. Chouzenoux. Graphical inference in linear-Gaussian state-space models. *IEEE Transactions on Signal Processing*, 70:4757–4771, 2022.
- 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*, 47(2):306–320, 1999.
- J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, jul 2008.
- R. Gao, F. Tronarp, and S. Sarkka. Iterated extended Kalman smoother-based variable splitting for l1-regularized state estimation. *IEEE Transactions on Signal Processing*, 67(19):5078–5092, 2015.
- G. Giannakis, Y. Shen, and G. Karanikolas. Topology identification and learning over graphs: Accounting for nonlinearities and dynamics. *Proceedings of the IEEE*, 106(5):787–807, May 2018.

- C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.
- P. J. Green and D. I. Hastie. Reversible jump MCMC. *Genetics*, 155(3):1391–1403, 2009.
- N. Gupta and R. Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, 19(6):774–783, 1974.
- J. D. Hamilton. State-space models. *Handbook of Econometrics*, 4:3039–3080, 1994.
- J. D. Hamilton. *Time series analysis*. Princeton university press, 2020.
- T. Hien, N. Gillis, and P. Patrinos. Inertial block proximal method for non-convex non-smooth optimization. In *Proceedings of the Thirty-seventh International Conference on Machine Learning (ICML 2020)*, 2020.
- A. Hippert-Ferrer, F. Bouchard, A. Mian, T. Vayer, and A. Breloy. Learning graphical factor models with Riemannian optimization. Technical report, 2022. <https://arxiv.org/abs/2210.11950>.
- M. Hong, M. Razaviyayn, Z.-Q. Luo, and J. S. Pang. A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Process. Mag.*, 33(1):57–77, Jan. 2016.
- D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Amer. Stat.*, 58(1):30–37, Feb. 2004.
- V. Ioannidis, D. Romero, and G. Giannakis. Inference of spatio-temporal functions over graphs via multikernel kriged Kalman filtering. *IEEE Transactions on Signal Processing*, 66(12):3228–3239, 2018.
- V. Ioannidis, Y. Shen, and G. Giannakis. Semi-blind inference of topologies and dynamical processes over dynamic graphs. *IEEE Transactions on Signal Processing*, 67(9):2263–2274, May 2019.
- M. Jacobson and J. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Transactions on Image Processing*, 16(10):2411–2422, 2007.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.
- N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. Technical report, 2015. <https://arxiv.org/abs/1412.8695>.
- C.-J. Kim and C. R. Nelson. *State-space models with regime switching: classical and Gibbs-sampling approaches with applications*. MIT Press Books, 1st edition, 1999.

- 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*, 32(6):31–54, 2015.
- S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. P. Palomar. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research*, 21(22):1–60, 2020.
- S. Lojasiewicz. Unpropriété topologique des sous-ensembles analytiques réels. *Editions du Centre National de la Recherche Scientifique*, pages 87–89, 1963.
- 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*, 12(1):143–155, Jan. 2019.
- M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright. *Handbook of Graphical Models*. CRC Press, Boca Raton, FL, 2019.
- N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.
- J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerical Mathematics*, 55:377–400, 1989.
- C. A. Naesseth, F. Lindsten, T. B. Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.
- D. Nagakura. Computing exact score vectors for linear Gaussian state space models. *Communications in Statistics - Simulation and Computation*, 50(8):2313–2326, 2021.
- R. Olsson, K. Petersen, and T. Lehn-Schioler. State-space models: from the EM algorithm to a gradient approach. *Neural Computation*, 19(4):1097–1111, 2007.
- S. Pérez-Vieites and J. Míguez. Nested gaussian filters for recursive bayesian inference and nonlinear tracking in state space models. *Signal Processing*, 189:108295, 2021.
- J. Runge, X.-A. Tibau, M. Bruhns, J. Muñoz-Marí, and G. Camps-Valls. The causality for climate competition. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, volume 123, pages 110–120, 2020.
- G. Sagi, N. Shlezinger, and T. Rautenberg. Extended Kalman filter for graph signals in nonlinear dynamic systems. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, Rhodes Island, Greece, 4-10 June 2023.
- S. Sarkka. *Bayesian Filtering and Smoothing*. 3 edition, 2013.
- M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient (score) of linear dynamic systems. *IEEE Transactions on Automatic Control*, 33(8):763–766, 1988. doi: 10.1109/9.1295.

- 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*, 35(3):682–687, 1989. doi: 10.1109/18.30995.
- S. Sharma, A. Majumdar, V. Elvira, and E. Chouzenoux. Blind Kalman filtering for short-term load forecasting. *IEEE Transactions on Power Systems*, 35(6):4916–4919, Nov. 2020.
- S. Sharma, V. Elvira, E. Chouzenoux, and A. Majumdar. Recurrent dictionary learning for state-space models with an application in stock forecasting. *Neurocomputing*, 450:1–13, Aug. 2021.
- A. Shojaie and G. Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.
- R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.
- Y. Sun, P. Babu, and D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794 – 816, 2016.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- L. T. K. Tien, D. N. Phan, and N. Gillis. An inertial block majorization minimization framework for nonsmooth nonconvex optimization. Technical report, 2020. <https://arxiv.org/abs/2010.12133>.
- C. Uhler. Gaussian graphical models: An algebraic and geometric perspective. Technical report, 2017. <https://arxiv.org/abs/1707.04345>.
- C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
- J. Ying, J. V. d. M. Cardoso, and D. P. Palomar. Does the  $\ell_1$ -norm learn a sparse graph under Laplacian constrained graphical models? Technical report, 2020. <https://arxiv.org/abs/2006.14925>.

**Algorithm 4** Proximal splitting method to solve (32)

**Inputs.**  $\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \Psi, \Delta, \Phi$ . Precision  $\xi > 0$ .

1. **Setting.** Set stepsize  $\vartheta \in (0, 2)$ .
2. **Initialization.** Set  $\mathbf{V}_0 = \tilde{\mathbf{A}}$ .
3. **Recursive step.** For  $n = 1, 2, \dots$ :

$$\begin{aligned}
 \mathbf{A}_n &= \text{prox}_{\theta_A \lambda_A \ell_1} \left( \tilde{\mathbf{A}} - \mathbf{V}_n \right) \\
 \mathbf{W}_n &= \mathbf{V}_n + \vartheta \mathbf{A}_n \\
 \mathbf{Z}_n &= \text{prox}_{\mathbf{W} \rightarrow \frac{\vartheta \theta_A K}{2} \text{tr}(-\tilde{\mathbf{P}} \Delta \mathbf{W} - \tilde{\mathbf{P}} \mathbf{W} \Delta^\top + \tilde{\mathbf{P}} \mathbf{W} \Psi \mathbf{W}^\top)} \left( \vartheta^{-1} \mathbf{W}_n \right) \\
 \mathbf{V}_{n+1} &= \mathbf{W}_n - \vartheta \mathbf{Z}_n.
 \end{aligned} \tag{95}$$

If  $|\mathcal{C}_1(\mathbf{A}_n) - \mathcal{C}_1(\mathbf{A}_{n-1})| \leq \xi$ , **stop the recursion.**

**Output.** Transition matrix  $\hat{\mathbf{A}} = \mathbf{A}_n$ .

**Algorithm 5** Proximal splitting method to solve (33)

**Inputs.**  $\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \Psi, \Delta, \Phi$ . Precision  $\xi > 0$ .

1. **Setting.** Set stepsize  $\vartheta \in (0, 2)$  and  $\Pi$  as in (34).
2. **Initialization.** Set  $\mathbf{V}_0 = \tilde{\mathbf{P}}$ .
3. **Recursive step.** For  $n = 1, 2, \dots$ :

$$\begin{aligned}
 \mathbf{P}_n &= \text{prox}_{\theta_P \lambda_P \ell_1} \left( \tilde{\mathbf{P}} - \mathbf{V}_n \right) \\
 \mathbf{W}_n &= \mathbf{V}_n + \vartheta \mathbf{P}_n \\
 \mathbf{Z}_n &= \text{prox}_{\mathbf{W} \rightarrow \frac{\vartheta \theta_P K}{2} (-\log \det(\mathbf{W}) + \text{tr}(\Pi \mathbf{W}))} \left( \vartheta^{-1} \mathbf{W}_n \right) \\
 \mathbf{V}_{n+1} &= \mathbf{W}_n - \vartheta \mathbf{Z}_n.
 \end{aligned} \tag{96}$$

If  $|\mathcal{C}_2(\mathbf{A}_n) - \mathcal{C}_2(\mathbf{A}_{n-1})| \leq \xi$ , **stop the recursion.**

**Output.** Precision matrix  $\hat{\mathbf{P}} = \mathbf{P}_n$ .

Table 4: Results for sparsity levels  $s_A \in \{27, 15, 10, 5\}$  of  $\mathbf{A}^*$ , using  $(\mathbf{P}^*, \mathbf{Q}^*)$  from dataset A. The first set of results, with  $s_A = 27$  identifies with the first row block of Tab. 1.

	Method	Estimation of $\mathbf{A}$			Estimation of $\mathbf{P}$			Estim. $\mathbf{Q}$	State distrib.		Predictive distrib.	
		RMSE	AUC	F1	RMSE	AUC	F1	RMSE	cNMSE( $\mu^*, \hat{\mu}$ )	cNMSE( $\mu^{s*}, \hat{\mu}^s$ )	cNMSE( $\nu^*, \hat{\nu}$ )	$\mathcal{L}_{1,K}(\mathbf{A}, \hat{\mathbf{P}})$
$s_A = 27$	DGLASSO	0.061	0.843	0.641	<b>0.082</b>	0.778	<b>0.698</b>	<b>0.083</b>	<b><math>6.394 \times 10^{-8}</math></b>	<b><math>1.050 \times 10^{-7}</math></b>	<b><math>2.984 \times 10^{-4}</math></b>	<b>12 307.169</b>
	MLEM	0.076	0.817	0.500	0.105	0.857	0.500	0.102	$1.095 \times 10^{-7}$	$1.803 \times 10^{-7}$	$4.843 \times 10^{-4}$	12 341.205
	GLASSO	NA	NA	NA	0.818	0.804	0.496	1073.510	$4.485 \times 10^{-6}$	$7.180 \times 10^{-6}$	1.000	28 459.294
	rGLASSO	NA	NA	NA	0.764	<b>0.924</b>	0.598	31.689	$2.826 \times 10^{-6}$	$5.492 \times 10^{-6}$	1.000	22 957.693
	GRAPHEM	<b>0.045</b>	<b>0.895</b>	<b>0.847</b>	NA	NA	NA	NA	$4.364 \times 10^{-6}$	$6.944 \times 10^{-6}$	$2.980 \times 10^{-4}$	29 035.030
$s_A = 15$	DGLASSO	<b>0.108</b>	0.916	0.781	<b>0.077</b>	0.883	<b>0.855</b>	<b>0.075</b>	<b><math>8.268 \times 10^{-7}</math></b>	<b><math>1.157 \times 10^{-6}</math></b>	<b><math>5.183 \times 10^{-3}</math></b>	<b>11 801.220</b>
	MLEM	0.159	<b>0.933</b>	0.313	0.105	<b>0.899</b>	0.500	0.101	$1.298 \times 10^{-6}$	$2.110 \times 10^{-6}$	$6.052 \times 10^{-3}$	11 806.483
	GLASSO	NA	NA	NA	0.463	0.702	0.576	7.088	$2.064 \times 10^{-5}$	$3.723 \times 10^{-5}$	1.000	14 258.010
	rGLASSO	NA	NA	NA	0.425	0.723	0.598	4.562	$1.379 \times 10^{-5}$	$2.768 \times 10^{-5}$	1.000	14 432.765
	GRAPHEM	0.127	0.928	<b>0.721</b>	NA	NA	NA	NA	$5.077 \times 10^{-5}$	$6.455 \times 10^{-5}$	$3.802 \times 10^{-3}$	29 034.715
$s_A = 10$	DGLASSO	0.089	0.981	<b>0.938</b>	<b>0.084</b>	0.768	<b>0.686</b>	<b>0.084</b>	<b><math>9.386 \times 10^{-7}</math></b>	<b><math>1.415 \times 10^{-6}</math></b>	<b><math>9.748 \times 10^{-3}</math></b>	12 290.789
	MLEM	0.167	<b>1.000</b>	0.220	0.105	<b>0.900</b>	0.500	0.101	$2.383 \times 10^{-6}$	$3.918 \times 10^{-6}$	$2.592 \times 10^{-2}$	<b>11 819.739</b>
	GLASSO	NA	NA	NA	0.365	0.667	0.500	0.469	$3.026 \times 10^{-5}$	$5.958 \times 10^{-5}$	1.000	13 173.462
	rGLASSO	NA	NA	NA	0.339	0.749	<b>0.699</b>	0.632	$1.970 \times 10^{-5}$	$4.375 \times 10^{-5}$	1.000	13 497.061
	GRAPHEM	<b>0.117</b>	<b>1.000</b>	0.745	NA	NA	NA	NA	$8.956 \times 10^{-5}$	$1.124 \times 10^{-4}$	$1.226 \times 10^{-2}$	29 034.727
$s_A = 5$	DGLASSO	<b>0.100</b>	0.823	<b>0.774</b>	<b>0.084</b>	0.764	<b>0.682</b>	<b>0.085</b>	<b><math>9.217 \times 10^{-7}</math></b>	<b><math>1.206 \times 10^{-6}</math></b>	<b><math>9.981 \times 10^{-3}</math></b>	<b>12 268.667</b>
	MLEM	0.234	0.893	0.116	0.106	<b>0.856</b>	0.500	0.102	$2.232 \times 10^{-6}$	$3.538 \times 10^{-6}$	$5.610 \times 10^{-2}$	12 312.749
	GLASSO	NA	NA	NA	0.242	0.667	0.500	0.306	$1.406 \times 10^{-5}$	$3.095 \times 10^{-5}$	1.000	12 892.144
	rGLASSO	NA	NA	NA	0.262	0.704	0.566	0.451	$1.195 \times 10^{-5}$	$2.884 \times 10^{-5}$	1.000	12 834.737
	GRAPHEM	0.127	<b>0.931</b>	0.573	NA	NA	NA	NA	$1.060 \times 10^{-4}$	$1.225 \times 10^{-4}$	$1.477 \times 10^{-2}$	29 034.456