MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

A Generalized GraphEM for Sparse Time-Varying Dynamical Systems

Greiff, Marcus; Di Cairano, Stefano; Mansour, Hassan; Berntorp, Karl

TR2023-086 July 11, 2023

Abstract

We consider the problem of joint parameter estimation and smoothing in structured linear systems using the expectation maximization (EM) framework. Specifically, we explore how partially known sparsity structures in the estimation model can be leveraged to improve the computation speed and performance of the considered EM approaches. We use these ideas to generalize a recently proposed GraphEM algorithm to a linear time-varying setting, where the sparsity structures may vary in time. We obtain a biconvex form of the majorizing function in the M -step, which is minimized subject to an ‘1-regularization using a Douglas-Rachford proximal splitting algorithm. Numerical results using a satellite positioning example shows significant improvements in the estimation errors and an F1-score that quantifies model sparsity.

World Congress of the International Federation of Automatic Control (IFAC) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

A Generalized GraphEM for Sparse Time-Varying Dynamical Systems

Marcus Greiff * **Stefano Di Cairano** * **Hassan Mansour** *
Karl Berntorp *

* *Mitsubishi Electric Research Labs, 02139 Cambridge, MA, USA*
(*e-mail: {greiff,mansour,berntorp}@merl.com, {dicairano}@ ieee.org*).

Abstract: We consider the problem of joint parameter estimation and smoothing in structured linear systems using the expectation maximization (EM) framework. Specifically, we explore how partially known sparsity structures in the estimation model can be leveraged to improve the computation speed and performance of the considered EM approaches. We use these ideas to generalize a recently proposed GraphEM algorithm to a linear time-varying setting, where the sparsity structures may vary in time. We obtain a biconvex form of the majorizing function in the M-step, which is minimized subject to an ℓ_1-regularization using a Douglas-Rachford proximal splitting algorithm. Numerical results using a satellite positioning example shows significant improvements in the estimation errors and an F1-score that quantifies model sparsity.

Keywords: Kalman filtering, RTS Smoothing, Parameter Estimation, Convex Optimization

1. INTRODUCTION

System identification is a cornerstone of modern control theory, in which the parameters of a system are to be determined from observed measurement data, without knowledge of the state trajectory (Ljung, 1998). Numerous approaches can be taken depending on the problem setting and prior knowledge of the system dynamics. For linear time-invariant (LTI) systems subject to additive noise, [1]

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{q}_k, \qquad \boldsymbol{q}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}), \tag{1a}$$
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{r}_k, \qquad \boldsymbol{r}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}), \tag{1b}$$

subspace methods (Katayama, 2005) such as variations of the state-space subspace system identification (4SID) algorithm (Viberg, 1995) remain popular to this day, with recent trends toward finite sample analysis (Sarkar et al., 2021; Tsiamis and Pappas, 2019) motivated by data-driven control methods. Other classical approaches such as prediction-error methods (PEM) and maximum-likelihood (ML) approaches (Ljung, 1998), come with the added benefit that prior knowledge in terms of model sparsity is relatively easy to encode in the resulting optimization problems. In the context of LTI system identification, the ML-methods are often combined with expectation maximization (EM) (Dempster et al., 1977) to generate iterative algorithms that utilize closed-form smoothing results to formulate computationally tractable algorithms, see e.g., (Neal and Hinton, 1998; Gibson and Ninness, 2005; Särkkä, 2013; Holmes, 2013; Chouzenoux and Elvira, 2020). These algorithms consist of constructing a majorizing function of the negative measurement log-likelihood (E-step), which is minimized over the unknown parameters (M-step). The later step is similar to the regression in the 4SID-algorithm (c.f., (Katayama, 2005, Chapter 6.6) and (Gibson and Ninness, 2005, Lemma 3.3)), but formulated in terms of a smoothing posterior.

In the LTI setting, EM methods tend to utilize the closed-form solutions to the resulting optimization problems in the model parameters $\boldsymbol{\Xi} \triangleq \{\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}\}$. When considering unknown sparsity in $\boldsymbol{\Xi}$, it has been suggested to add an ℓ_1-regularization term in the M-step (Chouzenoux and Elvira, 2020), referred to as the GraphEM algorithm, forgoing the closed-form solutions in favor of convex optimization that promotes sparsity. When instead considering known sparsity in $\boldsymbol{\Xi}$, [2] closed-form solutions for the M-step appear under assumptions explored in (Holmes, 2013). A more intuitive treatment of the same problem with an affine parametrization of $\boldsymbol{\Xi}$ is proposed in (Wills et al., 2018). In both works, the M-step is implemented by solving non-regularized convex optimization programs.

This paper reuses the core ideas and basis expressions in (Wills et al., 2018) and extends them to the time-varying setting, thereby encompassing a wide variety of linear-time varying (LTV) systems. In contrast to the work in (Holmes, 2013), we consider the parameters of $\{\boldsymbol{Q}, \boldsymbol{R}\}$ to be linear in the inverses of these covariance matrices (to be defined in Assumption 1). Combined, this allows us to encode partially known, but possibly time-varying, sparsity structures in $\boldsymbol{\Xi}$. As such, we extend the GraphEM algorithm to an LTV setting, where the transition model of the Markov chain $\{\boldsymbol{x}_k : k \in \mathbb{N}\}$ in (1) can be time varying.

The contributions of this paper are twofold. We present:

- A modified GraphEM with a weighted ℓ_1 regularization to promote the assumed sparsity structures in $\boldsymbol{\Xi}$, referred to as a Weighted GraphEM (or WGEM).
- A factorization of the majorizing objective function in the EM, which can be employed both in the LTI and LTV settings, where repeated minimization of the objective scales with the number of parameters representing $\boldsymbol{\Xi}$, instead of the state dimension. This is referred to as the Generalized GraphEM (or GGEM).

[1] The definitions and assumptions are made precise in Section 2.

[2] This notion of sparsity differs from (Neal and Hinton, 1998).

1.1 Notation

For $\boldsymbol{x} \in \mathbb{R}^n$, $x_i = [\boldsymbol{x}]_i$ is the i^{th} element of $\boldsymbol{x}$, and $D(\boldsymbol{x}) = n$. Matrices are indicated in bold as $\boldsymbol{X}$, and the element on row i and column j of $\boldsymbol{X}$ is $[\boldsymbol{X}]_{ij}$. Column unit vectors with the i^{th} element set to 1 are denoted by $\boldsymbol{e}_i \in \mathbb{R}^m$. The outer product of $\boldsymbol{a} \in \mathbb{R}^n$ with itself is $(\boldsymbol{a})(\boldsymbol{a})^\top = (\boldsymbol{a})(\star)^\top$. We let $\boldsymbol{M} \in \mathbb{S}^n_{++}$ ($\mathbb{S}^n_+$) denotes positive (semi) definite symmetric matrices, and $\otimes$ is the Kronecker product, where $\boldsymbol{M} = \boldsymbol{A} \otimes \boldsymbol{B}$ is block structured with entries $[\boldsymbol{A}]_{ij}\boldsymbol{B}$. Furthermore, $\circ$ denotes a Hadamard product, where $[\boldsymbol{M}]_{ij} = [\boldsymbol{A} \circ \boldsymbol{B}]_{ij} = [\boldsymbol{A}]_{ij}[\boldsymbol{B}_{ij}]$. We define a vector operation $\texttt{vec} : \mathbb{R}^{n\times m} \to \mathbb{R}^{nm\times 1}$ using lexicographical indexing, as in (Petersen et al., 2008), with an inverse $\texttt{mat} : \mathbb{R}^{nm\times 1} \to \mathbb{R}^{n\times m}$. The notation $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{P})$ indicates that $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{m}^{\boldsymbol{x}}$ and covariance $\boldsymbol{P}^{\boldsymbol{xx}}$, and the associated density function is $\mathcal{N}(\boldsymbol{x}|\boldsymbol{m}, \boldsymbol{P})$. Similarly, $\mathcal{U}(I)$ indicates a uniform distribution over an interval I. With $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state trajectory $\boldsymbol{x}_{0:k}$ from time step 0 to time step k given the measurement sequence $\boldsymbol{y}_{0:k} \coloneqq \{\boldsymbol{y}_0, \dots, \boldsymbol{y}_k\}$, where $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ is the marginal (filtering) posterior and $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:K})$ is a smoothing posterior (with $k \leq K$). For a function $y = \alpha F(x) + \beta$ with $\alpha > 0, \beta \in \mathbb{R}$ being constants independent of x, we let $y \simeq F(x)$. Similarly, if $y \leq \alpha F(x)+\beta$, then $y \lesssim F(x)$. We let the super-indices $(\cdot)^\star$ and $(\cdot)^\circ$ denote optimal and feasible solutions, respectively.

2. PRELIMINARIES

In the following, we consider LTV systems in the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k(\boldsymbol{\theta})\boldsymbol{x}_k + \boldsymbol{q}_k, \qquad \boldsymbol{q}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k(\boldsymbol{\theta})), \tag{2a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_k(\boldsymbol{\theta})\boldsymbol{x}_k + \boldsymbol{r}_k, \qquad \boldsymbol{r}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k(\boldsymbol{\theta})), \tag{2b}$$

with $\boldsymbol{x}_k \in \mathbb{R}^{D(\boldsymbol{x})}$, $\boldsymbol{y}_k \in \mathbb{R}^{D(\boldsymbol{y})}$, noise $\boldsymbol{q}_k \in \mathbb{R}^{D(\boldsymbol{x})}, \boldsymbol{r}_k \in \mathbb{R}^{D(\boldsymbol{y})}$ where the pair $(\boldsymbol{q}_k, \boldsymbol{r}_k)$ are uncorrelated, and $\boldsymbol{Q}_k(\boldsymbol{\theta}) \in \mathbb{S}^{D(\boldsymbol{x})}_{++}, \boldsymbol{R}_k(\boldsymbol{\theta}) \in \mathbb{S}^{D(\boldsymbol{y})}_{++}$ $\forall k \in [0, K] \subset \mathbb{N}$. Here, $\boldsymbol{\theta} \in \mathbb{R}^{D(\boldsymbol{\theta})}$ is a parameter vector. We assume the following:

Assumption 1. (Linear Parameter Dependence). The maps $\{\boldsymbol{A}_k(\boldsymbol{\theta}), \boldsymbol{Q}_k^{-1}(\boldsymbol{\theta}), \boldsymbol{C}_k(\boldsymbol{\theta}), \boldsymbol{R}_k^{-1}(\boldsymbol{\theta})\}$ are linear in $\boldsymbol{\theta}$, and if a parameter appears in one of the objects, it does not appear in another. If the parameters $\boldsymbol{\theta}_{\boldsymbol{M}}$ are associated with a matrix $\boldsymbol{M}_k$, then we can write $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\boldsymbol{A}}^\top, \boldsymbol{\theta}_{\boldsymbol{Q}}^\top, \boldsymbol{\theta}_{\boldsymbol{C}}^\top, \boldsymbol{\theta}_{\boldsymbol{R}}^\top)^\top$.

In the context of the LTV state-space model in (2), the Kalman filter (KF) is the minimum mean-square error (MMSE) estimator. In this setting, given a Gaussian prior over the states, $\mathcal{N}(\boldsymbol{x}_0|\boldsymbol{m}_0, \boldsymbol{P}_0)$, the marginal filtering posterior is Gaussian $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{m}_k, \boldsymbol{P}_k)$ and computed exactly by a recursion from the prior (Särkkä, 2013). If the entire measurement sequence is known, the smoothing posterior $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:K}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{m}_k^s, \boldsymbol{P}_k^s)$ can be computed recursively from the filtering posterior using the Rauch-Tung-Striebel (RTS) smoother. This is done in a backward pass, starting from $\boldsymbol{m}_K^s = \boldsymbol{m}_K$ and $\boldsymbol{P}_K^s = \boldsymbol{P}_K$,

$$\boldsymbol{m}'_{k+1} = \boldsymbol{A}_k(\boldsymbol{\theta})\boldsymbol{m}_k, \tag{3a}$$

$$\boldsymbol{P}'_{k+1} = \boldsymbol{A}_k(\boldsymbol{\theta})\boldsymbol{P}_k\boldsymbol{A}_k(\boldsymbol{\theta})^\top + \boldsymbol{Q}_k(\boldsymbol{\theta}), \tag{3b}$$

$$\boldsymbol{G}_k^s = \boldsymbol{P}_k\boldsymbol{A}_k(\boldsymbol{\theta})^\top(\boldsymbol{P}'_{k+1})^{-1}, \tag{3c}$$

$$\boldsymbol{m}_k^s = \boldsymbol{m}_k + \boldsymbol{G}_k^s(\boldsymbol{m}_{k+1}^s - \boldsymbol{m}'_{k+1}), \tag{3d}$$

$$\boldsymbol{P}_k^s = \boldsymbol{P}_k + \boldsymbol{G}_k^s(\boldsymbol{P}_{k+1} - \boldsymbol{P}'_{k+1})(\boldsymbol{G}_k^s)^\top. \tag{3e}$$

The closed-form smoothing solution can be leveraged in parameter estimation schemes, such as the EM method.

2.1 Expectation Maximization

We briefly describe the EM, referring to (Särkkä, 2013, Section 12.2.3) for additional details. The idea of EM is to take an arbitrary density $q(\boldsymbol{x}_{0:K})$, upper bound the negative log-likelihood of the measurements $\boldsymbol{y}_{0:K}$ given $\boldsymbol{\theta}$,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\triangleq -\log p(\boldsymbol{y}_{0:K}|\boldsymbol{\theta}) \\ &\leq -\int q(\boldsymbol{x}_{0:K}) \log \frac{p(\boldsymbol{y}_{0:K}, \boldsymbol{x}_{0:K}|\boldsymbol{\theta})}{q(\boldsymbol{x}_{0:K})} \mathrm{d}\boldsymbol{x}_{0:K},\end{aligned} \tag{4}$$

and subsequently minimize the left-hand side (LHS) in (4) by minimizing the upper bound on the right-hand side (RHS) in (4). The appeal of EM is that the LHS in (4) may be a complicated function in $\boldsymbol{\theta}$, while the RHS in (4) under certain choices of q can be expressed as a convex function in $\boldsymbol{\theta}$. Specifically, if $q(\boldsymbol{x}_{0:K}) \triangleq p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{0:K}, \boldsymbol{\theta}^{(i)})$,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\lesssim \int p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{0:K}, \boldsymbol{\theta}^{(i)}) \log p(\boldsymbol{y}_{0:K}, \boldsymbol{x}_{0:K}|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x}_{0:K} \\ &\triangleq \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}),\end{aligned}$$

where

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) \simeq &\sum_{k=0}^{K-1} \log\det(\boldsymbol{Q}_k(\boldsymbol{\theta})) + \sum_{k=0}^{K} \log\det(\boldsymbol{R}_k(\boldsymbol{\theta})) + \\ &\sum_{k=0}^{K} \mathrm{Tr}\left\{\boldsymbol{R}_k(\boldsymbol{\theta})^{-1}\mathbb{E}[(\boldsymbol{y}_k - \boldsymbol{C}_k(\boldsymbol{\theta})\boldsymbol{x}_k)(\star)^\top|\boldsymbol{y}_{0:K}]\right\} + \\ &\sum_{k=0}^{K-1} \mathrm{Tr}\left\{\boldsymbol{Q}_k(\boldsymbol{\theta})^{-1}\mathbb{E}[(\boldsymbol{x}_{k+1} - \boldsymbol{A}_k(\boldsymbol{\theta})\boldsymbol{x}_k)(\star)^\top|\boldsymbol{y}_{0:K}]\right\}.\end{aligned} \tag{5}$$

Due to the similarities in the functional relationships involving $\{\boldsymbol{C}(\boldsymbol{\theta}), \boldsymbol{R}(\boldsymbol{\theta})\}$ and $\{\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{Q}(\boldsymbol{\theta})\}$, we focus on the latter with the understanding that all subsequent derivations apply to both. In the LTI setting, we obtain

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)}) \simeq &-\log(\det(\boldsymbol{Q}^{-1})) \\ &+ \mathrm{Tr}\left\{\boldsymbol{Q}^{-1}(\boldsymbol{\Sigma} - \boldsymbol{\Gamma}\boldsymbol{A}^\top - \boldsymbol{A}\boldsymbol{\Gamma}^\top + \boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{A}^\top)\right\},\end{aligned} \tag{6}$$

where

$$\boldsymbol{\Sigma} = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{P}_{k+1}^s + \boldsymbol{m}_{k+1}^s(\boldsymbol{m}_{k+1}^s)^\top, \tag{7a}$$

$$\boldsymbol{\Gamma} = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{P}_{k+1}^s(\boldsymbol{G}_k^s)^\top + \boldsymbol{m}_{k+1}^s(\boldsymbol{m}_k^s)^\top, \tag{7b}$$

$$\boldsymbol{\Phi} = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{P}_k^s + \boldsymbol{m}_k^s(\boldsymbol{m}_k^s)^\top. \tag{7c}$$

with the local minimizers of (6) given by

$$\boldsymbol{\theta}_{\boldsymbol{A}}^\star = \texttt{vec}(\boldsymbol{\Gamma}\boldsymbol{\Phi}^{-1}), \tag{8a}$$

$$\boldsymbol{\theta}_{\boldsymbol{Q}}^\star = \texttt{vec}(\boldsymbol{\Sigma} - \boldsymbol{\Gamma}\boldsymbol{A}^\top - \boldsymbol{A}\boldsymbol{\Gamma}^\top + \boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{A}^\top). \tag{8b}$$

The EM algorithm consists of first computing a smoothing posterior given $\boldsymbol{\theta}^{(i)}$ (the E-step), and subsequently updating $\boldsymbol{\theta}^{(i)} \to \boldsymbol{\theta}^{(i+1)}$ by minimizing (6) (the M-step), as summarized in Algorithm 1. In this basic formulation, the EM: (i) does not straightforwardly generalize to the LTV setting; (ii) does not easily incorporate prior knowledge of sparsity; and (iii) does not enforce sparsity in $\boldsymbol{\theta}$. The last point was addressed in the recently proposed GraphEM.

Algorithm 1 EM With Unknown Sparsity

1: **Receive:** $\boldsymbol{y}_{0:K}, \boldsymbol{m}_0, \boldsymbol{P}_0, \boldsymbol{\theta}^{(0)}$
2: **for** $i = 0, 1, \ldots, N-1$ **or until converged do**
// E-step
3: Evaluate $\{\boldsymbol{m}^s_{0:K}, \boldsymbol{P}^s_{0:K}, \boldsymbol{G}^s_{0:K} | \boldsymbol{\theta}^{(i)}\}$ by (3)
// M-step
4: Solve $\boldsymbol{\theta}^{(i+1)} = \operatorname{argmin}_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)})$ using (8)
5: **end for**
6: **Output:** $\boldsymbol{m}^s_{0:K}, \boldsymbol{P}^s_{0:K}, \boldsymbol{\theta}^{(N)}$

3. LINEAR TIME-INVARIANT SETTING

The GraphEM in (Chouzenoux and Elvira, 2020) is also developed in an LTI setting, computing the maximum a posteriori (MAP) estimate of $\boldsymbol{A}$ subject to an ℓ_1 regularization. As in the original algorithm, we here use the linear parameterization $\boldsymbol{A} = \texttt{mat}(\boldsymbol{\theta}_{\boldsymbol{A}}) \in \mathbb{R}^{D(\boldsymbol{x}) \times D(\boldsymbol{x})}$, let $\bar{\boldsymbol{A}} = \texttt{mat}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) \in \mathbb{R}^{D(\boldsymbol{x}) \times D(\boldsymbol{x})}$ for convenience, and consider

$$\min_{\boldsymbol{\theta}_{\boldsymbol{A}}} f(\boldsymbol{\theta}_A) + g(\boldsymbol{\theta}_A), \tag{9}$$

where $f(\boldsymbol{\theta}_A) = \mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)})|_{\boldsymbol{\theta}_{\boldsymbol{Q}} = \boldsymbol{\theta}^{\circ}_{\boldsymbol{Q}}}$, and $g(\boldsymbol{\theta}_A) = \lambda \|\boldsymbol{\theta}_{\boldsymbol{A}}\|_1$. A proximal operator is defined in the Frobenius norm, as

$$\texttt{prox}_{\alpha f}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) = \operatorname*{argmin}_{\boldsymbol{\theta}_{\boldsymbol{A}}} \alpha f(\boldsymbol{\theta}_A) + \frac{1}{2}\|\boldsymbol{A} - \bar{\boldsymbol{A}}\|_F^2 \tag{10a}$$

$$= (\boldsymbol{I} \otimes \boldsymbol{Q} + \alpha \boldsymbol{\Phi} \otimes \boldsymbol{I})^{-1} \texttt{vec}(\alpha \boldsymbol{\Gamma} + \boldsymbol{Q}\bar{\boldsymbol{A}}). \tag{10b}$$

The second equality in (10) follows from first-order optimality conditions, resulting in a Lyapunov equation whose solution is given in (10b), see, e.g., (Petersen et al., 2008). This form differs slightly to the original GraphEM, but its numerical conditioning may differ depending on the application. As in (Chouzenoux and Elvira, 2020), the proximal operator for the ℓ_1-regularizer is defined as

$$\texttt{prox}_{\alpha g}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) = \operatorname*{argmin}_{\boldsymbol{\theta}_{\boldsymbol{A}}} \alpha\lambda \|\boldsymbol{\theta}_{\boldsymbol{A}}\|_1 + \frac{1}{2}\|\boldsymbol{\theta}_{\boldsymbol{A}} - \bar{\boldsymbol{\theta}}_{\boldsymbol{A}}\|_2^2 \tag{11a}$$

$$= \operatorname{sign}([\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}]_i) \max(0, |[\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}]_i| - \alpha\lambda). \tag{11b}$$

In GraphEM, (9) is solved using a Douglas-Ratchford (DR) algorithm (Combettes and Pesquet, 2007) in lieu of iterative thresholding methods, such as the iterative shrinkage-thresholding algorithm (ISTA) in (Daubechies et al., 2004) or its accelerated equivalent in (Beck and Teboulle, 2009). The DR subroutine is given in Algorithm 2. For details on proximal splitting and convergence proofs, refer to (Parikh et al., 2014) and (Giselsson, 2015).

In contrast to the EM, the computational complexity in the GraphEM is dominated by solving the Lyapunov equation in (10) at $O(D(\boldsymbol{x})^6)$. However, if the process noise is isotropic, i.e., $\boldsymbol{Q} = \sigma^2 \boldsymbol{I}$, the inversion in (10) simplifies and has complexity $O(D(\boldsymbol{x})^3)$. In our notation, the resulting proximal operator can be written

$$\texttt{prox}_{\alpha f}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) = \texttt{vec}\Big\{ \Big(\frac{\alpha}{\sigma^2}\boldsymbol{\Gamma} + \bar{\boldsymbol{A}} \Big) \Big(\boldsymbol{I} + \frac{\alpha}{\sigma^2}\boldsymbol{\Phi} \Big)^{-1} \Big\}. \tag{12}$$

In both (10) and (12), consecutive evaluations of this proximal operator in the DR can be done by forward/backward substitution since $(\boldsymbol{I} \otimes \boldsymbol{Q} + \alpha \boldsymbol{\Phi} \otimes \boldsymbol{I})$ admits a Cholesky factorization. The iteration complexity of the DR-routine in Algorithm 2 is $O(D(\boldsymbol{x})^4)$, or $O(D(\boldsymbol{x})^2)$ if the process noise is isotropic. The GraphEM in (Chouzenoux and Elvira, 2020) is implemented as in Algorithm 1, iterating over $\boldsymbol{A}$ and replacing the minimization on line 4 with the

Algorithm 2 DR-subroutine used in the GraphEM.

1: **Receive:** $\boldsymbol{z}^{(0)}$
2: Factorize $\boldsymbol{L}^\top \boldsymbol{L} = \boldsymbol{I} \otimes \boldsymbol{Q} + \alpha \boldsymbol{\Phi} \otimes \boldsymbol{I}$ if possible
3: **for** $i = 0, 1, 2, \ldots, N-1$ **or until converged do**
4: $\boldsymbol{y}^{(i)} = \texttt{prox}_{\alpha f}(\boldsymbol{z}^{(i)})$ with substitution using $\boldsymbol{L}$
5: $\boldsymbol{x}^{(i)} = \texttt{prox}_{\alpha g}(2\boldsymbol{y}^{(i)} - \boldsymbol{z}^{(i)})$
6: $\boldsymbol{z}^{(i+1)} = \boldsymbol{z}^{(i)} + \boldsymbol{y}^{(i)} - \boldsymbol{x}^{(i)}$
7: **end for**
8: **Output:** $\boldsymbol{\theta}_{\boldsymbol{A}} = \boldsymbol{y}^{(i)}$

DR in Algorithm 2, where nominally $\alpha = 1$. In the implementation for this paper, the Cholesky factors associated with $\texttt{prox}_{\alpha f}$ are precomputed at the start of each M-step.

Remark 1. The reason for not identifying the noise covariance matrices to an ℓ_1 constraint with a proximal operator defined in the Frobenius norm is that a Riccati-like matrix equation emerges. Solutions exist and can be computed as in (Higham and Kim, 2000) but are generally not unique. As such, the GraphEM requires $\boldsymbol{Q}$ to be dense and estimated as in (8) with a large iteration complexity of the DR. Alternatively, the model class can be restricted to isotropic process noise $\boldsymbol{Q} = \sigma^2 \boldsymbol{I}$, resulting in an M-step

$$\sigma^2 = \frac{1}{D(\boldsymbol{x})} \operatorname{Tr}(\boldsymbol{\Sigma} - \boldsymbol{\Gamma}\boldsymbol{A}^\top - \boldsymbol{A}\boldsymbol{\Gamma}^\top + \boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{A}^\top). \tag{13}$$

In the original GraphEM, the system matrix is assumed to be sparse, but the general sparsity patterns are completely unknown. However, we can make this prior more precise by a weighted ℓ_1-regularization. Consider a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{D(\boldsymbol{x}) \times D(\boldsymbol{x})}$ with positive elements where a large value indicates that the corresponding element in $\boldsymbol{A}$ is more likely to be zero in the solution, and consider a regularizer

$$\bar{g}(\boldsymbol{\theta}_{\boldsymbol{A}}) = \|\boldsymbol{\theta}_{\boldsymbol{A}}\|_{1, \texttt{vec}(\boldsymbol{\Lambda})} = \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{A}})} [\texttt{vec}(\boldsymbol{\Lambda})]_i |[\boldsymbol{\theta}_{\boldsymbol{A}}]_i|, \tag{14}$$

with an associated proximal operator

$$\texttt{prox}_{\alpha \bar{g}}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) = \operatorname*{argmin}_{\boldsymbol{\theta}_{\boldsymbol{A}}} \alpha \bar{g}(\boldsymbol{\theta}_{\boldsymbol{A}}) + \frac{1}{2}\|\boldsymbol{\theta}_{\boldsymbol{A}} - \bar{\boldsymbol{\theta}}_{\boldsymbol{A}}\|_2^2 \tag{15a}$$

$$= \operatorname{sign}([\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}]_i) \max(0, |[\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}]_i| - \alpha[\texttt{vec}(\boldsymbol{\Lambda})]_i). \tag{15b}$$

This facilitates more informative priors over $\boldsymbol{A}$ in the GraphEM, referred to as the Weighted GraphEM (WGEM).

4. LINEAR TIME-VARYING SETTING

The core ideas translate to a more general LTV setting. Instead of having the (weighted) ℓ_1-norm regularize $\boldsymbol{A}$, we span $\boldsymbol{A}_k$ at a specific time step k using a basis defined in Assumption 2, and have the ℓ_1-norm regularize $\boldsymbol{A}_k$ such that redundant basis matrices are removed. To restrict the model class, we make the following assumptions:

Assumption 2. The system matrix $\boldsymbol{A}_k$ is spanned by a time-varying basis $\{\boldsymbol{A}_k^{(i)} \in \mathbb{R}^{D(\boldsymbol{x}) \times D(\boldsymbol{x})} \backslash \{\boldsymbol{0}\}\}_{i=0}^{D(\boldsymbol{\theta}_{\boldsymbol{A}})}$,

$$\boldsymbol{A}_k(\boldsymbol{\theta}) = \boldsymbol{A}_k^{(0)} + \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{A}})} \boldsymbol{A}_k^{(i)} [\boldsymbol{\theta}_{\boldsymbol{A}}]_i. \tag{16}$$

Assumption 3. The noise covariance $\boldsymbol{Q}_k$ is spanned by time-varying basis $\{\boldsymbol{Q}_k^{-1,(i)} \in \mathbb{R}^{D(\boldsymbol{x}) \times D(\boldsymbol{x})}\}_{i=0}^{D\boldsymbol{\theta}_{\boldsymbol{Q}}}$,

$$\boldsymbol{Q}_k^{-1}(\boldsymbol{\theta}) = \boldsymbol{Q}_k^{-1,(0)} + \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{Q}})} \boldsymbol{Q}_k^{-1,(i)} [\boldsymbol{\theta}_{\boldsymbol{Q}}]_i. \tag{17}$$

Assumptions 2 and 3 enable a time-varying $\boldsymbol{\theta}$, as it can be encoded through the time-varying matrix basis. In Section 4.1 we show that the solution to the M-step can be computed analytically if an additional assumption is made on the inverse noise basis in (17), see Assumption 4.

Assumption 4. $\boldsymbol{Q}_k^{-1}(\boldsymbol{\theta})$ is block diagonal, where:

(i) each element $\boldsymbol{Q}_k^{-1,(i)}$ of the basis is constituted by $m_{ik}^{\boldsymbol{Q}} > 0$ non-zero blocks $\boldsymbol{Q}_{jk}^{-1,(i)} \in \mathbb{S}_{++}^{n_{ijk}^{\boldsymbol{Q}} \times n_{ijk}^{\boldsymbol{Q}}}$;
(ii) on each time step, the elements of the basis are non-overlapping, with $\boldsymbol{Q}_k^{-1,(i)} \circ \boldsymbol{Q}_k^{-1,(j)} = \boldsymbol{0}\ \forall i \neq j$;
(iii) $[\boldsymbol{\theta}_{\boldsymbol{Q}}]_i > 0 \forall i \Leftrightarrow \boldsymbol{Q}_k^{-1}(\boldsymbol{\theta}) \succ 0$.

4.1 Factoring the $\mathcal{Q}$-function

We start by considering the component of the $\mathcal{Q}$-function in (6) related to $\{\boldsymbol{A}_k(\boldsymbol{\theta}), \boldsymbol{Q}_k(\boldsymbol{\theta})\}$, restated for convenience:

$$\mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)}) \simeq -\sum_{k=0}^{K-1} \log(\det(\boldsymbol{Q}_k^{-1}(\boldsymbol{\theta}))) + \sum_{k=0}^{K-1} \operatorname{Tr}\left\{\boldsymbol{Q}_k(\boldsymbol{\theta})^{-1} \mathbb{E}[(\boldsymbol{x}_{k+1} - \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}_k)(\star)^\top | \boldsymbol{y}_{0:K}]\right\}. \tag{18}$$

Proposition 1. By Assumptions 1–4, (18) can be written

$$\mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)}) \simeq \bar{\mathcal{Q}}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)}) \triangleq \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{Q}})} \Big\{ -M^{(i)} \log([\boldsymbol{\theta}_{\boldsymbol{Q}}]_i) \\ [\boldsymbol{\theta}_{\boldsymbol{Q}}]_i \Big((\boldsymbol{\theta}_{\boldsymbol{A}})^\top \boldsymbol{H}^{(i)} \boldsymbol{\theta}_{\boldsymbol{A}} + (\boldsymbol{f}^{(i)})^\top \boldsymbol{\theta}_{\boldsymbol{A}} + c^{(i)} \Big) \Big\}, \tag{19}$$

with objects $\boldsymbol{H}^{(i)} \in \mathbb{S}_+^{D(\boldsymbol{\theta}_{\boldsymbol{A}}) \times D(\boldsymbol{\theta}_{\boldsymbol{A}})}$, $\boldsymbol{f}^{(i)} \in \mathbb{R}^{D(\boldsymbol{\theta}_{\boldsymbol{A}})}$, $c^{(i)} \in \mathbb{R}$, $M^{(i)} \in \mathbb{Z}_{\geq 0}$ defined for all $i \in 0, ..., D(\boldsymbol{\theta}_{\boldsymbol{Q}})$.

This follows algebraically, and the optimization problems

$$\boldsymbol{\theta}_{\boldsymbol{A}}^\star = \operatorname{argmin}_{\boldsymbol{\theta}_{\boldsymbol{A}}} \mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)})|_{\boldsymbol{\theta}_{\boldsymbol{Q}} = \boldsymbol{\theta}_{\boldsymbol{Q}}^\circ}, \tag{20a}$$

$$\boldsymbol{\theta}_{\boldsymbol{Q}}^\star = \operatorname{argmin}_{\boldsymbol{\theta}_{\boldsymbol{Q}}} \mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)})|_{\boldsymbol{\theta}_{\boldsymbol{A}} = \boldsymbol{\theta}_{\boldsymbol{A}}^\circ}, \tag{20b}$$

are (individually) convex, with minimizers given by

$$\boldsymbol{\theta}_{\boldsymbol{A}}^\star = -(2\boldsymbol{H})^{-1}\boldsymbol{f}, \tag{21a}$$

$$[\boldsymbol{\theta}_{\boldsymbol{Q}}^\star]_i = \frac{M^{(i)}}{(\boldsymbol{\theta}_{\boldsymbol{A}}^\circ)^\top \boldsymbol{H}^{(i)} \boldsymbol{\theta}_{\boldsymbol{A}}^\circ + (\boldsymbol{f}^{(i)})^\top \boldsymbol{\theta}_{\boldsymbol{A}}^\circ + c^{(i)}}, \tag{21b}$$

$$\boldsymbol{H} = \boldsymbol{H}^{(0)} + \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{Q}})} [\boldsymbol{\theta}_{\boldsymbol{Q}}^\circ]_i \boldsymbol{H}^{(i)}, \tag{21c}$$

$$\boldsymbol{f} = \boldsymbol{f}^{(0)} + \sum_{i=1}^{D(\boldsymbol{\theta}_{\boldsymbol{Q}})} [\boldsymbol{\theta}_{\boldsymbol{Q}}^\circ]_i \boldsymbol{f}^{(i)}. \tag{21d}$$

Here, the biconvex form in (19) is easily evaluated in the case where the basis spanning $\{\boldsymbol{A}_k(\boldsymbol{\theta}_{\boldsymbol{A}}), \boldsymbol{Q}_k(\boldsymbol{\theta}_{\boldsymbol{Q}})\}$ are time-varying, while the objects $\{\boldsymbol{H}^{(i)}, \boldsymbol{f}^{(i)}, c^{(i)}\}$ are independent of time (due to assumptions 1–4) thereby facilitating extensions of the GraphEM to an LTV setting.

Instead of evaluating (21b) directly, we can once again add an ℓ_1-regularization over $\boldsymbol{\theta}_{\boldsymbol{A}}$, and consider the problem

$$\min_{\boldsymbol{\theta}_{\boldsymbol{A}}} \bar{f}(\boldsymbol{\theta}_{\boldsymbol{A}}) + g(\boldsymbol{\theta}_{\boldsymbol{A}}), \tag{22}$$

with $\bar{f}(\boldsymbol{\theta}_{\boldsymbol{A}}) = \bar{\mathcal{Q}}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)})|_{\boldsymbol{\theta}_{\boldsymbol{Q}} = \boldsymbol{\theta}_{\boldsymbol{Q}}^\circ}$ in (19) and $g(\boldsymbol{\theta}_{\boldsymbol{A}}) = \lambda \|\boldsymbol{\theta}_{\boldsymbol{A}}\|_1$. We define a proximal operator in the two-norm,

$$\texttt{prox}_{\alpha \bar{f}}(\bar{\boldsymbol{\theta}}_{\boldsymbol{A}}) = \operatorname*{argmin}_{\boldsymbol{\theta}_{\boldsymbol{A}}} \left\{ \alpha \bar{f}(\boldsymbol{\theta}_A) + \frac{1}{2} \|\boldsymbol{\theta}_{\boldsymbol{A}} - \bar{\boldsymbol{\theta}}_{\boldsymbol{A}}\|_2^2 \right\} \tag{23a}$$

$$= -(\alpha \boldsymbol{H} + \boldsymbol{I})^{-1} (\bar{\boldsymbol{\theta}}_{\boldsymbol{A}} - (\alpha/2)\boldsymbol{f}), \tag{23b}$$

with $\{\boldsymbol{H}, \boldsymbol{f}\}$ given in (21c) and (21d), respectively. We use $\texttt{prox}_{\alpha g}$ from (11b) and solve (22) using the DR in Algorithm 2, but now run with respect to $\{\bar{f}, g\}$ using the proximal operator in (23). This replaces the M-step in the conventional EM defined in Algorithm 1 with an M-step utilizing the DR-subroutine and the decomposition in Proposition 1, referred to as the Generalized GraphEM (GGEM). To make the implementation efficient, we compute the Cholesky factors of $(\alpha \boldsymbol{H} + \boldsymbol{I})$ at the start of each M-step, and evaluate (23) in the DR algorithm by substitution. Any update of $\boldsymbol{\theta}_{\boldsymbol{Q}}$ is done using (21b), and updates in both $\boldsymbol{\theta}_{\boldsymbol{A}}$ and $\boldsymbol{\theta}_{\boldsymbol{Q}}$ are done in the fashion of a single-step block coordinate descent.

5. NUMERICAL EXAMPLES

Many systems can be modeled with partially known sparsity in the prediction and measurement models. One such example comes from satellite positioning, where a receiver's kinematic states, $\boldsymbol{z}^\top = (\boldsymbol{p}^\top, \boldsymbol{v}^\top) \in \mathbb{R}^6$, are to be inferred from on a set of distance measurements corrupted by various biases (Berntorp et al., 2020). Here, $\boldsymbol{p}$ is the receiver's position, $\boldsymbol{v} = \dot{\boldsymbol{p}}$ is the receiver's velocity, and the three dimensions are assumed to be differentially independent. To show how the proposed methods can be used, we consider a model similar to (Greiff et al., 2021) and introduce parameter adaptation in the prediction model. To this end, we start by finding a suitable basis for the prediction model, satisfying Assumptions 2–4. We let

$$\boldsymbol{A}_k^{\text{CV}} \triangleq \begin{bmatrix} 1 & h_k \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2},\ \boldsymbol{M}_k^{\text{CV}} \triangleq \frac{12}{h_k^3} \begin{bmatrix} 1 & -\frac{h_k}{2} \\ -\frac{h_k}{2} & \frac{h_k^2}{3} \end{bmatrix} \in \mathbb{S}_{++}^{2\times 2}, \tag{24}$$

where $h_k > 0$ [s] is a sampling period, such that $p(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k) \sim \mathcal{N}((\boldsymbol{I}_3 \otimes \boldsymbol{A}_k^{\text{CV}})\boldsymbol{z}_k, (\boldsymbol{I}_3 \otimes (\boldsymbol{M}_k^{\text{CV}})^{-1}))$ is a constant velocity (CV) model with unit variance. As in Greiff et al. (2021), we take the biases associated with $N_s = 10$ satellites to evolve by a constant position model. A suitable basis for the inverse process-noise is then $\boldsymbol{Q}_k^{-1,(0)} = \boldsymbol{0}$ and

$$\boldsymbol{Q}_k^{-1,(1)} = \operatorname{diag}(\boldsymbol{M}_k^{\text{CV}}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}), \tag{25a}$$

$$\boldsymbol{Q}_k^{-1,(2)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{M}_k^{\text{CV}}, \boldsymbol{0}, \boldsymbol{0}), \tag{25b}$$

$$\boldsymbol{Q}_k^{-1,(3)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{M}_k^{\text{CV}}, \boldsymbol{0}), \tag{25c}$$

$$\boldsymbol{Q}_k^{-1,(4)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, h_k^{-1}\boldsymbol{I}_{3N_s}), \tag{25d}$$

resulting in $D(\boldsymbol{\theta}_{\boldsymbol{Q}}) = 4$, corresponding to the inverse variance of the random walks driving the constant velocity and position models. For illustration purposes, assume that $\boldsymbol{A}_k^{\text{CV}}$ is unknown to the estimator, but that the positions and biases are differentially independent. We get $D(\boldsymbol{\theta}_{\boldsymbol{A}}) = 12$ parameters and 13 basis matrices for $\boldsymbol{A}$ which can be characterized by $\boldsymbol{e}_1 = (1, 0)^\top$ and $\boldsymbol{e}_2 = (0, 1)^\top$, as

$$\boldsymbol{A}_k^{(0)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{I}),\quad \boldsymbol{A}_k^{(i+4)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{N}_i, \boldsymbol{0}, \boldsymbol{0}), \tag{26a}$$

$$\boldsymbol{A}_k^{(i)} = \operatorname{diag}(\boldsymbol{N}_i, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}), \boldsymbol{A}_k^{(i+8)} = \operatorname{diag}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{N}_i, \boldsymbol{0}), \tag{26b}$$

for $i \in \{1, 2, 3, 4\}$ with

$$\boldsymbol{N}_1 = \boldsymbol{e}_1\boldsymbol{e}_1^\top,\ \boldsymbol{N}_2 = \boldsymbol{e}_1\boldsymbol{e}_2^\top,\ \boldsymbol{N}_3 = \boldsymbol{e}_2\boldsymbol{e}_1^\top,\ \boldsymbol{N}_4 = \boldsymbol{e}_2\boldsymbol{e}_2^\top.$$

The measurements are sampled with the true parameters

$$\boldsymbol{\theta}^{\text{true}} = (\underbrace{(\mathbf{1}_3 \otimes \texttt{vec}(\boldsymbol{A}_k^{\text{CV}}))^\top}_{\boldsymbol{\theta}_{\boldsymbol{A}}^{\text{true}}}, \underbrace{\sigma_X^{-2}, \sigma_Y^{-2}, \sigma_Z^{-2}, \sigma_B^{-2}}_{\boldsymbol{\theta}_{\boldsymbol{Q}}^{\text{true}}})^\top, \quad (27)$$

but the smoother is initialized according to

$$\boldsymbol{\theta}_{\boldsymbol{A}}^{(0)} - \boldsymbol{\theta}_{\boldsymbol{A}}^{\text{true}} \sim \mathcal{N}(\mathbf{0}, 0.1^2 \boldsymbol{I}), \quad (28a)$$

$$[\boldsymbol{\theta}_{\boldsymbol{Q}}^{(0)}]_i / [\boldsymbol{\theta}_{\boldsymbol{Q}}^{\text{true}}]_i \sim \mathcal{U}([0.1, 9]),\ \forall i = 1, \ldots, D(\boldsymbol{\theta}_{\boldsymbol{Q}}). \quad (28b)$$

The measurement model is defined as in (Greiff et al., 2021), but in an undifferenced setting with inflated noise,

$$\boldsymbol{y}_k = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{p}_k) \\ \boldsymbol{h}(\boldsymbol{p}_k) \\ \mathbf{0}_{10\times 1} \\ \mathbf{0}_{10\times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{10\times 6} & \mathbf{0}_{10} & -\boldsymbol{I}_{10} & \boldsymbol{I}_{10} \\ \mathbf{0}_{10\times 6} & 0.2\boldsymbol{I}_{10} & \boldsymbol{I}_{10} & \boldsymbol{I}_{10} \\ \mathbf{0}_{10\times 6} & \mathbf{0}_{10} & \boldsymbol{I}_{10} & \mathbf{0}_{10} \\ \mathbf{0}_{10\times 6} & \mathbf{0}_{10} & \mathbf{0}_{10} & \boldsymbol{I}_{10} \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{r}_k, \quad (29)$$

where $[\boldsymbol{h}(\boldsymbol{x}_k)]_i = \|\boldsymbol{p}_k - \boldsymbol{p}_k^i\|_2$ is the Euclidean distance between the receiver position $\boldsymbol{p}_k \in \mathbb{R}^3$ and the ith satellite, located at a position $\boldsymbol{p}_k^i \in \mathbb{R}^3$. Here, deterministic models of a subset of the biases are introduced as measurements of the corresponding biases. The measurement noise and unbiased estimate prior are characterized by

$$\boldsymbol{R}_k = \text{diag}(0.5\boldsymbol{I}_{10}, 0.05\boldsymbol{I}_{10}, 0.05\boldsymbol{I}_{10}, 0.05\boldsymbol{I}_{10})\ \ \forall k, \quad (30a)$$

$$\boldsymbol{P}_0 = \text{diag}(5, 1, 5, 1, 5, 1, 1\boldsymbol{I}_{10}, 0.1\boldsymbol{I}_{20}). \quad (30b)$$

Remark 2. In this setting, the M-step in: the regular EM requires the inversion of $\boldsymbol{\Phi} \in \mathbb{R}^{36\times 36}$; while the GraphEM requires the inversion of $(\boldsymbol{I} \otimes \boldsymbol{Q} + \boldsymbol{\Phi}^\top \otimes \boldsymbol{I}) \in \mathbb{R}^{1296\times 1296}$ if the model class is to encompass the true noise covariance, and a matrix in $\mathbb{R}^{36\times 36}$ if the noise is isotropic; and the GGEM requires the inversion of a matrix in $\mathbb{R}^{12\times 12}$.

For the comparison, we use Taylor expansions of the measurement model to compute the smoothing posterior by the extended RTS smoother (Särkkä, 2013). We consider:

(A) Smoothing with the true model (True);
(B) An EM with dense noise cov. (EM-A);
(C) An EM with isotropic noise cov. (EM-B);
(D) A GraphEM with dense noise cov. (GraphEM-A);
(E) A GraphEM with isotropic noise (GraphEM-B);
(F) A WGEM with dense noise cov. and sparsity pattern in $\boldsymbol{\Lambda}$ same as the model of the GGEM (WGEM-A);
(G) A WGEM with isotropic noise and sparsity pattern in $\boldsymbol{\Lambda}$ same as the model of the GGEM (WGEM-B);
(H) A GGEM algorithm with the basis in (25) and (26).

Here, (A) serves as a baseline for thew comparison; (B)-(C) are direct implementations of the EM in Särkkä (2013); (D)-(E) are two variants of the method proposed in (Chouzenoux and Elvira, 2020); (F)-(G) is the GraphEM with weighted regularization proposed in Sec. 3; and (H) is the generalized GraphEM method proposed in Sec. 4.

The methods are compared using metrics similar to those considered in (Chouzenoux and Elvira, 2020):

- Time-averaged position root-mean square error (RMSE) in centimeters, which should be no less than the smoothing posterior evaluated in the true model parameters (approximating the Cramér-Rao bound).
- The F1-score of the estimates $\{\boldsymbol{A}(\boldsymbol{\theta}^{(N)}), \boldsymbol{Q}(\boldsymbol{\theta}^{(N)})\}$, where an element is determined to be zero (negative) or non-zero (positive) with a threshold of 10^{-10}, the same metric as in (Chouzenoux and Elvira, 2020).
- The number of zero elements (NZ) in the maps $\{\boldsymbol{A}(\boldsymbol{\theta}^{(N)}), \boldsymbol{Q}(\boldsymbol{\theta}^{(N)})\}$ combined.

Table 1. Estimation and computational performance, arrows show direction of improvement.

Algorithm	RMSE (↓)	F1 $\boldsymbol{A}$ (↑)	F1 $\boldsymbol{Q}$ (↑)	Tot. NZ.
True	1.288	–	–	2511.0
EM-A	23.074	0.109	0.118	1259.2
EM-B	16.296	0.106	0.923	1888.0
GraphEM-A	27.718	0.189	0.111	1752.0
GraphEM-B	42.411	0.111	0.923	2458.1
WGEM-A	**1.825**	**1.000**	0.122	1903.0
WGEM-B	3.383	0.938	0.923	2513.6
GGEM	1.999	0.984	**1.000**	**2509.7**
	Tot. CT	EM iters	CT/M-step	DR iters
True	0.2	–	–	–
EM-A	19.4	95.6	**0.014**	–
EM-B	18.4	89.5	**0.014**	–
GraphEM-A	207.4	100.0	1.88 0	860.3
GraphEM-B	8.0	30.8	0.065	603.4
WGEM-A	2458.8	46.0	53.256	24511.0
WGEM-B	26.4	47.7	0.349	3671.7
GGEM	**5.9**	**27.5**	0.022	**124.0**

The estimation performance is summarized in Table 1, along with (i) the total computational time including all E- and M-steps (CT) in seconds; (ii) the average number of EM iterations; (iii) the average computational time per M-step; and (iv) average number of DR-iterations per M-step (if applicable). The statistics are averaged over 50 Monte-Carlo runs, where the initial parameter errors in (28) are realized differently, and we only perform a single run with the GraphEM-A and WGEM-A due to their excessively long computation times (see Remark 2). The stopping criteria in the DR in Algorithm 2 is defined $\|\texttt{vec}(\boldsymbol{y}^{(i)} - \boldsymbol{x}^{(i)})\|_1 \leq 10^{-5}$, and the EM iterations are run until the decrease in the M-step objective is less than 10^{-2} or for a maximum number of $N \leq 100$ iterations.

From the results in Table 1, it appears that there is some benefit to allowing a dense noise covariance matrix, especially when adding the (weighted) ℓ_1 regularizers. It is also noteworthy that the sparsity induced by the ℓ_1 constraint in the GraphEM is not always beneficial, and that the number of non-zero elements in $\boldsymbol{A}$ greatly depends on λ. Setting $\lambda = 1$ in the GraphEM provides a level of sparsity in $\boldsymbol{A}$ roughly corresponding to that of the true model. The standard deviation of the number of zero elements (NZ) in $\boldsymbol{A}$ and $\boldsymbol{Q}$ was ≈ 40 for the GraphEM. It is also noteworthy that the convergence of the dense variants of EM-A and GraphEM-A required an excessive number of iterations to converge, almost always hitting the iteration upper bound before the numerical convergence criteria. Some realizations would require >1000 EM iterations to achieve performance comparable to the other EM variants.

In WGEM, a much more restrictive prior can be enforced over $\boldsymbol{A}$, with the weights in $\boldsymbol{\Lambda}$ corresponding to the elements that are assumed to be zero-valued set a factor of 10^3 higher. This results in parameter estimates that are very similar to the model parameters. This is best seen in the F1-score associated with $\boldsymbol{A}$ which approaches 1. As we get closer to the true model, without over-fitting the likelihood, we get far superior results in terms of RMSE.

The GGEM uses the correct model class for the process noise covariance. This enables a more accurate estimation

model to be inferred, and we get a F1-score of 1 with respect to $\boldsymbol{Q}$. We also see one of the highest F1-scores with respect to the system matrix $\boldsymbol{A}$. As the GGEM accurately estimates the model, we also expect its RMSE to be low. Only the WGEM-A outperforms the GGEM, and here we again emphasize that only one simulation was done with the WGEM-A while several were averaged for the GGEM.

In terms of computational properties, the use of a dense process noise in the context of a relatively large estimation model (recall, $D(\boldsymbol{x}) = 36$ with 10 satellites) results in a significant computational burden when evaluating the proximal operators in (10), effectively rendering the GraphEM-A and WGEM-A practically infeasible, despite the latter having one of the lowest RMSEs. Convergence is generally faster with the GGEM, both with respect to the number of EM-iterations used and the number of DR-iterations per M-step. One reason for this could be that we can introduce scaling in the DR (by α) when using the proximal operator in (23). We set $\alpha^{-1} = \|\boldsymbol{H}\|_2$, adapting this as $\boldsymbol{H}$ in (21c) changes over the GGEM iterations. For the GraphEM and WGEM using (10) and (12), it is more difficult to define an appropriate scaling parameter.

In summary, with respect to estimation performance, and in the context of the sparse estimation model, there appears to be significant benefits in including richer priors over the estimation model using the WGEM, or performing the optimization over a smaller, well chosen set of parameters in the GGEM. The update in the GGEM is comparable in speed to the update in the regular EM, and significantly faster than the construction of the $\mathcal{Q}$-function in the M-step. With the exception of the methods assuming dense process noise, the total computational time is largely dominated by the evaluation of the smoothing posterior. In this setting, the GGEM requires fewer smoothing passes than any other considered method to achieve the convergence criteria.

6. CONCLUSION

In this paper, we propose an extension to the GraphEM algorithm in two ways. First, we introduce a weighted ℓ_1 regularizer to encode partially known sparsity patterns in the estimation model. Next, we introduce the GGEM to reduce the number of parameter optimized in the M-step given this partially known sparsity structure, which comes with the added benefit of generalizing to an LTV setting.

We demonstrate that the choice of EM and regularizer has a significant impact on performance, especially for large and sparse estimation models. The numerical results indicate that the WGEM and GGEM should be preferred over both the original GraphEM and EM if we seek a sparse estimation model when we have prior knowledge of this sparsity structure, and that the GGEM should be preferred in terms of the computational properties when $D(\boldsymbol{\theta})$ is relatively small. Importantly, after computing the factorization in Proposition 1, and factoring $(\alpha\boldsymbol{H} + \boldsymbol{I})$, consecutive evaluations of the proximal operator (23) in the DR of the GGEM scale with $O(D(\boldsymbol{\theta}_{\boldsymbol{A}})^2)$, resulting in a fast and efficient M-step, which can be used in both LTI and LTV settings under the assumptions in Sec. 4.

Future work will apply the GGEM in global navigation satellite system post-processing, where it will be used to handle variable sample rates and the time-varying measurement equations arising from the satellite movement.

REFERENCES

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1), 183–202.

Berntorp, K., Weiss, A., and Di Cairano, S. (2020). Integer ambiguity resolution by mixture Kalman filter for improved GNSS precision. *IEEE Trans. Aerosp. Electron. Syst.*, 56(4), 3170–3181.

Chouzenoux, E. and Elvira, V. (2020). GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints. In *IEEE Int. Conf. on Acoustics, Speech and Sig. Processing*, 5840–5844. IEEE.

Combettes, P.L. and Pesquet, J.C. (2007). A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Selected Topics in Signal Processing*, 1(4), 564–574.

Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11), 1413–1457.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39, 1–22.

Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10), 1667–1682.

Giselsson, P. (2015). Tight linear convergence rate bounds for Douglas-Rachford splitting and ADMM. In *Conference on Decision and Control (CDC)*, 3305–3310.

Greiff, M., Berntorp, K., Di Cairano, S., and Kim, K.J. (2021). Mixed-integer linear regression Kalman filters for GNSS positioning. In *IEEE Conf. on Control Technology and Applications*, 980–985.

Higham, N.J. and Kim, H.M. (2000). Numerical analysis of a quadratic matrix equation. *IMA J. Numerical Analysis*, 20(4), 499–519.

Holmes, E.E. (2013). Derivation of an EM algorithm for constrained and unconstrained multivariate autoregressive state-space (MARSS) models. *arXiv preprint arXiv:1302.3919*.

Katayama, T. (2005). *Subspace methods for system identification*, volume 1. Springer.

Ljung, L. (1998). System identification. In *Signal analysis and prediction*, 163–173. Springer.

Neal, R.M. and Hinton, G.E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, 355–368. Springer.

Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and trends in Optimization*, 1(3), 127–239.

Petersen, K.B., Pedersen, M.S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15), 510.

Sarkar, T., Rakhlin, A., and Dahleh, M.A. (2021). Finite time LTI system identification. *J. Machine Learning Research*, 22.

Särkkä, S. (2013). *Bayesian filtering and smoothing*, volume 3. Cambridge university press.

Tsiamis, A. and Pappas, G.J. (2019). Finite sample analysis of stochastic system identification. In *IEEE 58th Conference on Decision and Control*, 3648–3654.

Viberg, M. (1995). Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12), 1835–1851.

Wills, A., Yu, C., Ljung, L., and Verhaegen, M. (2018). Affinely parametrized state-space models: Ways to maximize the likelihood function. *IFAC-PapersOnLine*, 51(15), 718–723.