Lasso-Kalman Smoother for Tracking Sparse Signals

Daniele Angelosante
Department of ECE
University of Minnesota
Minneapolis, MN 55455, USA
email: angel129@umn.edu

Stergios I. Roumeliotis
Department of CS
University of Minnesota
Minneapolis, MN 55455, USA
email: stergios@cs.umn.edu

Georgios B. Giannakis
Department of ECE
University of Minnesota
Minneapolis, MN 55455, USA
email: georgios@umn.edu

***Abstract*—Fixed-interval smoothing of time-varying vector processes is an estimation approach with well-documented merits for tracking applications. The optimal performance in the linear Gauss-Markov model is achieved by the Kalman smoother (KS), which also admits an efficient recursive implementation. The present paper deals with vector processes for which it is known *a priori* that many of their entries equal to zero. In this context, the process to be tracked is sparse, and the performance of sparsity-agnostic KS schemes degrades considerably. On the other hand, it is shown here that a sparsity-aware KS exhibits complexity which grows exponentially in the vector dimension. To obtain a tractable alternative, the KS cost is regularized with the sparsity-promoting ℓ_1 norm of the vector process – a relaxation also used in linear regression problems to obtain the least-absolute shrinkage and selection operator (Lasso). The Lasso (L)KS derived in this work is not only capable of tracking sparse time-varying vector processes, but can also afford an efficient recursive implementation based on the alternating direction method of multipliers (ADMoM). Finally, a weighted (W)-LKS is also introduced to cope with the bias of the LKS, and simulations are provided to validate the performance of the novel algorithms.**

I. Introduction

Sparsity is a well-documented attribute present in natural as well as in various man-made signals and systems encountered in engineering applications, as diverse as image processing [6], wireless communications [4], and biostatistics [8]. Recent advances in variable selection [8], and compressive sampling [6], have leveraged sparsity to a host of additional applications involving linear and nonlinear estimation problems.

One relevant paradigm where sparsity can facilitate the solution of nonlinear estimation problems, is that of localization. Using a grid of candidate points on which target(s) may lie in space, it is possible to approximate this nonlinear estimation problem with a linear one, at the price of increasing the number of unknowns [4]. Grid spacing depends on the desirable resolution, and exploitation of the sparsity present (since only a few grid points are occupied by targets) can effectively reduce the number of unknowns. The same approach can be envisioned for target tracking using e.g., range measurements, where instead of linearizing the nonlinear measurement equation [3, p. 381], one can use a grid-based partitioning of the state-space to obtain an approximate linear model of higher dimension, and subsequently rely on sparsity to extract the few nonzero entries of the state vector.

Exploiting sparsity to track time-varying signals has been considered in [10], where a sparsity-aware Kalman filter is proposed to track abrupt changes in the support of dynamic magnetic resonance imaging (MRI) signals. A sparsity-aware recursive estimator of un-modeled signal variations is developed in [2]. Relative to [2] and [10], the novelty of the present paper is a sparsity-aware fixed-interval smoother.

Fixed-interval smoothing of time-varying signals has been extensively used for post-processing of target tracks [1, p.187]. The optimal estimator of the linear Gauss-Markov state-space model is the Kalman smoother (KS), which also enjoys a computationally efficient recursive implementation [1, p. 189]. However, if many of the state vector entries are *a priori* known to be zero, the clairvoyant sparsity-agnostic KS is no longer optimal. On the other hand, optimal smoothing under sparsity constraints entails an exhaustive search over all possible positions where state entries may be nonzero. This search incurs computational complexity that scales exponentially with the state dimension. Such limitation motivates a convex relaxation of the KS cost using the ℓ_1-norm of the state. The latter is also invoked by the least-absolute shrinkage and selector operator (Lasso), a popular tool used in solving sparse linear regression problems. The resulting Lasso (L)KS criterion involves a non-smooth convex cost, whose minimization does not generally admit a closed-form solution. For this reason, an iterative solver is developed in this paper based on the alternating direction method of multipliers (ADMoM), which is amenable to an efficient recursive implementation. Numerical tests illustrate that LKS outperforms KS in estimating time-varying sparse state processes.

The rest of this paper is structured as follows. Section II contains preliminaries, and the problem statement. The novel LKS is developed in Section III, where a computationally efficient implementation via the ADMoM is also presented. Simulations are provided in Section IV, and conclusions are drawn in Section V.

Notation: Column vectors (matrices) are denoted with lower (upper) case boldface letters, and sets with calligraphic letters; $(\cdot)^T$ stands for transposition; $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian p.d.f. with mean μ, and variance σ^2; $\mathbf{0}_P$ and $\mathbf{0}_{P\times N}$ are the $P\times 1$ vector and the $P\times N$ matrix of zeros, respectively, and $\mathbf{I}_P$ is

Algorithm 1 Kalman filter.

Input: $\{\mathbf{y}_n, \mathbf{H}_n, \mathbf{R}_n, \mathbf{T}_n, \mathbf{Q}_n\}_{n=1}^N$, $\boldsymbol{\mu}_1$, and $\boldsymbol{\Sigma}_1$
Set $\boldsymbol{\mu}_{1|0} := \boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_{1|0} := \boldsymbol{\Sigma}_1$
for $n = 1, \ldots, N$ **do**

(Updating)
U1. $\mathbf{r}_n = \mathbf{y}_n - \mathbf{H}_n \boldsymbol{\mu}_{n|n-1}$
U2. $\mathbf{S}_n = \mathbf{H}_n \boldsymbol{\Sigma}_{n|n-1} \mathbf{H}_n^T + \mathbf{R}_n$
U3. $\mathbf{K}_n = \boldsymbol{\Sigma}_{n|n-1} \mathbf{H}_n^T \mathbf{S}_n^{-1}$
U4. $\boldsymbol{\mu}_n = \boldsymbol{\mu}_{n|n-1} + \mathbf{K}_n \mathbf{r}_n$
U5. $\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_{n|n-1} - \mathbf{K}_n \mathbf{S}_n \mathbf{K}_n^T$

(Prediction)
P1. $\boldsymbol{\mu}_{n+1|n} = \mathbf{T}_n \boldsymbol{\mu}_n$
P2. $\boldsymbol{\Sigma}_{n+1|n} = \mathbf{T}_n \boldsymbol{\Sigma}_n \mathbf{T}_n^T + \mathbf{Q}_n$
end for
Output: $\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{n=1}^N$

the $P \times P$ identity matrix. The ℓ_2 norm of $\mathbf{x} \in \mathbb{R}^P$ is $\|\mathbf{x}\|_2 := (\sum_{p=1}^P x_p^2)^{1/2}$. If $\{\mathbf{x}_n\}_{n=1}^N$ are $P \times 1$ vectors, then $\mathbf{x}_{1:N} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$. A diagonal matrix with $\mathbf{x} = [x_1, \ldots, x_P]^T \in \mathbb{R}^P$ on its diagonal is denoted as $\text{diag}(\mathbf{x}) \in \mathbb{R}^{P \times P}$; and a block diagonal matrix with g-th block equal to $\mathbf{T}_g \in \mathbb{R}^{P_g \times P_g}$ as $\text{bdiag}(\mathbf{T}_1, \ldots, \mathbf{T}_G) \in \mathbb{R}^{P \times P}$, with $\sum_{g=1}^G P_g = P$.

II. Preliminaries and problem statement

Let $\mathbf{x}_n := [x_{1,n}, \ldots, x_{P,n}]^T \in \mathbb{R}^P$ denote the P-dimensional unknown state vector at time n, observed through the linear transformation

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{x}_n + \mathbf{v}_n \tag{1}$$

where $\mathbf{y}_n := [y_{1,n}, \ldots, y_{M,n}]^T \in \mathbb{R}^M$, $\mathbf{H}_n \in \mathbb{R}^{M \times P}$, and $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}_M, \mathbf{R}_n)$ represents the additive, zero-mean, white Gaussian noise at time n.

The time evolution of the p-th entry is modeled through the first-order recursion

$$x_{p,n+1} = \beta_{p,n} x_{p,n} + e_{p,n} \tag{2}$$

for $n = 1, \ldots, N-1$; and $x_{p,1} \sim \mathcal{N}(\mu_{p,1}, \sigma_{p,1}^2)$, if p belongs to the support set $\mathcal{S}_x \subset \{1, \ldots, P\}$, which is unknown, while $x_{p,n} = 0$, if $p \notin \mathcal{S}_x$. Since $\mathbf{x}_n$ is assumed *sparse*, it follows that $|\mathcal{S}_x| \ll P$. Parameters $\{\beta_{p,n}, p = 1, \ldots, P, n = 1, \ldots, N-1\}$ are known, and $e_{p,n} \sim \mathcal{N}(0, q_{p,n}^2)$.

Given $\{\mathbf{y}_n\}$, $\{\mathbf{H}_n\}$, $\{\mathbf{R}_n\}$, $\{\beta_{p,n}\}$, $\{q_{p,n}^2\}$, and $\{\mu_{p,1}, \sigma_{p,1}^2\}$ for $n = 1, \ldots, N$ and $p = 1, \ldots, P$, the goal is to estimate the sparse sequence $\{\mathbf{x}_n\}_{n=1}^N$ obeying (1) and (2).

A. Fixed-interval Kalman Smoothing

Neglecting the sparsity information or equivalently assuming that $\mathcal{S}_x = \{1, \ldots, P\}$, the optimal minimum mean-square error estimator of $\{\mathbf{x}_n\}_{n=1}^N$, is given by the fixed-interval Kalman smoother (KS). Upon defining $\boldsymbol{\mu}_1 := [\mu_{1,1}, \ldots, \mu_{P,1}]^T$, $\boldsymbol{\Sigma}_1 := \text{diag}(\sigma_{1,1}^2, \ldots, \sigma_{P,1}^2)$, $\mathbf{T}_n := \text{diag}(\beta_{1,n}, \ldots, \beta_{P,n}) \in \mathbb{R}^{N \times N}$, and $\mathbf{Q}_n := \text{diag}(q_{1,n}^2, \ldots, q_{P,n}^2) \in \mathbb{R}^{N \times N}$, the KS amounts to estimating

Algorithm 2 Backward recursion.

Input:
Set $\boldsymbol{\mu}_N^S := \boldsymbol{\mu}_N$, $\boldsymbol{\Sigma}_N^S := \boldsymbol{\Sigma}_N$
for $n = N-1, \ldots, 1$ **do**
B1. $\mathbf{B}_n = \mathbf{T}_n \boldsymbol{\Sigma}_n \mathbf{T}_n^T + \mathbf{Q}_n$
B2. $\mathbf{C}_n = \boldsymbol{\Sigma}_n \mathbf{T}_n^T \mathbf{B}_n^{-1}$
B3. $\boldsymbol{\mu}_n^S = \boldsymbol{\mu}_n + \mathbf{C}_n(\boldsymbol{\mu}_{n+1}^S - \mathbf{T}_n \boldsymbol{\mu}_n^S)$
B4. $\boldsymbol{\Sigma}_n^S = \boldsymbol{\Sigma}_n + \mathbf{T}_n(\boldsymbol{\Sigma}_{n+1}^S - \mathbf{B}_n)\mathbf{T}_n^T$
end for
B5. $\mathbf{x}_{1:N}^{\text{KS}} = \boldsymbol{\mu}_{1:N}^S$
Output: $\mathbf{x}_{1:N}^{\text{KS}}$

the state as

$$\begin{aligned}\mathbf{x}_{1:N}^{\text{KS}} = \arg\min_{\mathbf{x}_{1:N}} \Bigg[&\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\ &+ \frac{1}{2} \sum_{n=1}^N (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n)^T \mathbf{R}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n) \\ &+ \frac{1}{2} \sum_{n=1}^{N-1} (\mathbf{x}_{n+1} - \mathbf{T}_n \mathbf{x}_n)^T \mathbf{Q}_n^{-1} (\mathbf{x}_{n+1} - \mathbf{T}_n \mathbf{x}_n) \Bigg]. \end{aligned} \tag{3}$$

The minimum of the linear-quadratic cost in (3) can be found in closed form. However, this solution is infeasible since the number of unknowns (NP) grows with the time horizon, and the matrix inversion required eventually becomes intractable. For this reason, recursive solvers of (3) offer the only practically viable alternative as N grows [1, p. 189]. The first step toward obtaining $\mathbf{x}_{1:N}^{\text{KS}}$ is a Kalman filtering iteration as tabulated under Algorithm 1, followed by a backward recursion outlined under Algorithm 2. The procedure described in Algorithms 1-2 yields the KS estimates with complexity that scales linearly with N.

III. Lasso-Kalman smoother

As mentioned earlier, the KS yields the optimal estimator when $\mathcal{S}_x = \{1, \ldots, P\}$. If it is a priori known that $\mathcal{S}_x \neq \{1, \ldots, P\}$, then KS is clearly suboptimal as it does not exploit all the available information. A possible remedy is to implement a KS for each candidate support $\mathcal{S}_x^c \in \mathcal{P}(\{1, \ldots, P\})$, where $\mathcal{P}(\cdot)$ denotes the power set operator, and choose the most *likely* candidate support. The limitation of this exhaustive search is that $|\mathcal{P}(\{1, \ldots, P\})| = 2^P$ and it necessitates running 2^P KSs in parallel; this is clearly infeasible for P sufficiently large. The exhaustive search approach just described can be formalized as

$$\begin{aligned}\hat{\mathbf{x}}_{1:N} = \arg\min_{\mathbf{x}_{1:N}} \Bigg[&\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\ &+ \frac{1}{2} \sum_{n=1}^N (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n)^T \mathbf{R}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n) \\ &+ \frac{1}{2} \sum_{n=1}^{N-1} (\mathbf{x}_{n+1} - \mathbf{T}_n \mathbf{x}_n)^T \mathbf{Q}_n^{-1} (\mathbf{x}_{n+1} - \mathbf{T}_n \mathbf{x}_n) \\ &+ \gamma \sum_{p=1}^P \delta(x_{p,1}, \ldots, x_{p,N}) \Bigg] \end{aligned} \tag{4}$$

where

$$\delta(x_{p,1},\ldots,x_{p,N}) = \begin{cases} 0, & \text{if } x_{p,n}=0 \text{ for } n=1,\ldots,N \\ 1, & \text{otherwise} \end{cases} . \tag{5}$$

Note that a larger $\gamma > 0$ in (4) gives rise to a sparser $\widehat{\mathbf{x}}_{1:N}$.

Motivated by recent advances in variable selection and compressive sampling, an effective relaxation of the non-convex problem in (4) is given by the following convex optimization problem

$$\begin{aligned}\mathbf{x}_{1:N}^{\text{LKS}} = \arg\min_{\mathbf{x}_{1:N}} \Bigg[& \frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_1-\boldsymbol{\mu}_1) \\ & + \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n)^T\mathbf{R}_n^{-1}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n) \\ & + \frac{1}{2}\sum_{n=1}^{N-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n)^T\mathbf{Q}_n^{-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n) \\ & + \lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}x_{p,n}^2}\Bigg]. \end{aligned} \tag{6}$$

We will term the estimator in (6) Lasso-Kalman smoother (LKS), because the convex regularization term $\lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}x_{p,n}^2}$ resembles the one adopted by the so-called group-Lasso approach in [9]. The latter forces groups $\{x_{p,n}\}_{n=1}^N$ (as opposed to individual entries) to zero; also, as λ increases more groups $\{x_{p,n}\}_{n=1}^N$ tend to become zero.

Being convex but non-differentiable, the problem in (6) can be solved with standard techniques ranging from subgradient iterations for non-smooth costs to interior-point methods. The limitation of these general-purpose solvers is that the structure of the cost in (6) is not accounted for. In addition, without time-recursive solvers, only short-to-medium size intervals can be processed. An efficient time-recursive algorithm will be developed in the ensuing section to exploit the structure of the cost in (6).

A. LKS solver via ADMoM

The crux of ADMoM is revealed by reformulating (6) as follows:

$$\begin{aligned}[\mathbf{x}_{1:N}^{\text{LKS}}, \widehat{\boldsymbol{\chi}}_{1:N}] = \arg\min_{\mathbf{x}_{1:N},\boldsymbol{\chi}_{1:N}} \Bigg[& \frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_1-\boldsymbol{\mu}_1) \\ & + \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n)^T\mathbf{R}_n^{-1}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n) \\ & + \frac{1}{2}\sum_{n=1}^{N-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n)^T\mathbf{Q}_n^{-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n) \\ & + \lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}\chi_{p,n}^2}\Bigg] \\ & \text{subject to } x_{1:N} = \boldsymbol{\chi}_{1:N} \end{aligned} \tag{7}$$

where the constraint in $\mathbf{x}_{1:N} = \boldsymbol{\chi}_{1:N}$ corresponds to NP scalar equality constraints; that is, $x_{p,n} = \chi_{p,n}$ for $p = 1,\ldots,P$ and $n = 1,\ldots,N$, and $\{\chi_{p,n}\}$ are auxiliary variables such that $\boldsymbol{\chi}_n := [\chi_{1,n},\ldots,\chi_{P,n}]^T$, and $\boldsymbol{\chi}_{1:N} := [\boldsymbol{\chi}_1,\ldots,\boldsymbol{\chi}_N]$. Associating the Lagrangian multipliers $\mathbf{u}_{1:N}$ with the constraints $\mathbf{x}_{1:N} = \boldsymbol{\chi}_{1:N}$, the (quadratically) augmented Lagrangian for the problem in (7) is

$$\begin{aligned}\mathcal{L}(\mathbf{x}_{1:N},\boldsymbol{\chi}_{1:N},\mathbf{u}_{1:N}) := & \frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_1-\boldsymbol{\mu}_1) \\ & + \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n)^T\mathbf{R}_n^{-1}(\mathbf{y}_n-\mathbf{H}_n\mathbf{x}_n) \\ & + \frac{1}{2}\sum_{n=1}^{N-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n)^T\mathbf{Q}_n^{-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n) \\ & + \lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}\chi_{p,n}^2} + \sum_{n=1}^{N}\sum_{p=1}^{P}u_{p,n}(x_{p,n}-\chi_{p,n}) \\ & + \frac{c}{2}\sum_{n=1}^{N}\sum_{p=1}^{P}(x_{p,n}-\chi_{p,n})^2 \end{aligned}$$

where c is any positive constant. Selecting arbitrary initial values $\boldsymbol{\chi}_{1:N}^{(0)}$, and $\mathbf{u}_{1:N}^{(0)}$, the ADMoM algorithm iteratively implements for $i = 1, 2, \ldots$ the following steps [5, p. 253]:

$$\mathbf{x}_{1:N}^{(i)} = \arg\min_{\mathbf{x}_{1:N}} \mathcal{L}(\mathbf{x}_{1:N},\boldsymbol{\chi}_{1:N}^{(i-1)},\mathbf{u}_{1:N}^{(i-1)}) \tag{8}$$

$$\boldsymbol{\chi}_{1:N}^{(i)} = \arg\min_{\boldsymbol{\chi}_{1:N}} \mathcal{L}(\mathbf{x}_{1:N}^{(i)},\boldsymbol{\chi}_{1:N},\mathbf{u}_{1:N}^{(i-1)}) \tag{9}$$

$$\mathbf{u}_{1:N}^{(i)} = \mathbf{u}_{1:N}^{(i-1)} + c(\mathbf{x}_{1:N}^{(i)} - \boldsymbol{\chi}_{1:N}^{(i)}). \tag{10}$$

Through this process, the original problem in (6) is divided into the sub-problems in (8) and (9), both of which admit a closed-form solution. Indeed, ignoring irrelevant constant terms, (8) can be re-written as

$$\begin{aligned}\mathbf{x}_{1:N}^{(i)} = \arg\min_{\mathbf{x}_{1:N}} \Bigg[& \frac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_1-\boldsymbol{\mu}_1) \\ & + \frac{1}{2}\sum_{n=1}^{N}(\widetilde{\mathbf{y}}_n-\widetilde{\mathbf{H}}_n\mathbf{x}_n)^T\widetilde{\mathbf{R}}_n^{-1}(\widetilde{\mathbf{y}}_n-\widetilde{\mathbf{H}}_n\mathbf{x}_n) \\ & + \frac{1}{2}\sum_{n=1}^{N-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n)^T\mathbf{Q}_n^{-1}(\mathbf{x}_{n+1}-\mathbf{T}_n\mathbf{x}_n)\Bigg] \end{aligned} \tag{11}$$

with

$$\widetilde{\mathbf{y}}_n := \begin{bmatrix} \mathbf{y}_n \\ \boldsymbol{\chi}_n^{(i-1)} - c^{-1}\mathbf{u}_n^{(i-1)} \end{bmatrix} \tag{12}$$

$$\widetilde{\mathbf{H}}_n := \begin{bmatrix} \mathbf{H}_n \\ \mathbf{I}_P \end{bmatrix} \tag{13}$$

$$\widetilde{\mathbf{R}}_n := \begin{bmatrix} \mathbf{R}_n & \mathbf{0}_{M\times P} \\ \mathbf{0}_{P\times M} & c^{-1}\mathbf{I}_P \end{bmatrix}. \tag{14}$$

Clearly, the problem in (11) can be solved efficiently by the KS as described in Algorithms 1-2 with input $\{\widetilde{\mathbf{y}}\}_{n=1}^N$, $\{\widetilde{\mathbf{H}}_n\}_{n=1}^N$, $\{\widetilde{\mathbf{R}}_n\}_{n=1}^N$, $\{\mathbf{T}_n\}_{n=1}^N$, $\{\mathbf{Q}_n\}_{n=1}^N$, $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$, and $\{\boldsymbol{\chi}_n^{(i-1)}, \mathbf{u}_n^{(i-1)}\}_{n=1}^N$ computed in the previous iteration. Thus, the computational burden to carry out the optimization in (8) scales linearly with N.

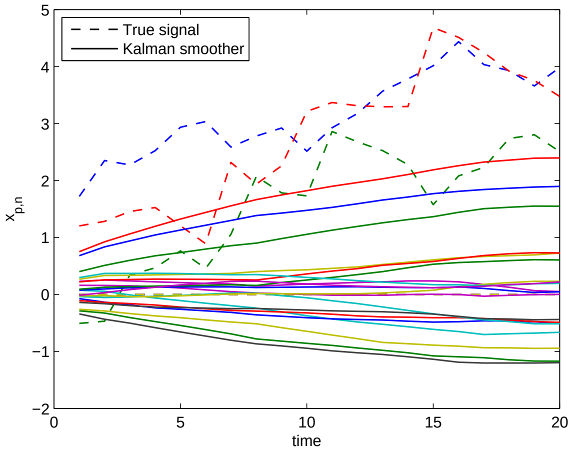

Fig. 1. KS estimates versus true signal: KS estimates are not sparse and true signal variations are not tracked.

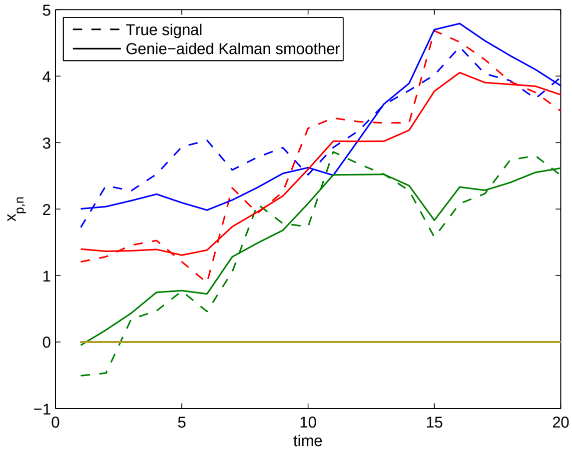

Fig. 2. GAKS estimates versus true signal: the GAKS knows the locations of the active entries and performs KS only on the active support.

Given $\mathbf{x}_n^{(i)}$ evaluated from (11), and $\mathbf{u}_n^{(i-1)}$) (computed in the previous iteration), and after neglecting constant terms, (9) can be re-written as

$$\boldsymbol{\chi}_{1:N}^{(i)} = \arg\min_{\boldsymbol{\chi}_{1:N}} \left[\lambda \sum_{p=1}^{P} \sqrt{\sum_{n=1}^{N} \chi_{p,n}^2} - \sum_{n=1}^{N}\sum_{p=1}^{P} u_{p,n}^{(i-1)} \chi_{p,n} + \frac{c}{2}\sum_{n=1}^{N}\sum_{p=1}^{P} (x_{p,n}^{(i)} - \chi_{p,n})^2 \right]. \quad (15)$$

Observe that the minimization in (15) can be performed on a per-component basis; that is,

$$[\chi_{p,1}^{(i)}, \ldots, \chi_{p,N}^{(i)}]^T = \arg\min_{\chi_{p,1},\ldots,\chi_{p,N}} \left[\lambda \sqrt{\sum_{n=1}^{N} \chi_{p,n}^2} - \sum_{n=1}^{N} u_{p,n}^{(i-1)} \chi_{p,n} + \frac{c}{2}\sum_{n=1}^{N} (x_{p,n}^{(i)} - \chi_{p,n})^2 \right]. \quad (16)$$

Upon defining $\boldsymbol{x}_p := [x_{p,1}, \ldots, x_{p,N}]^T$ and $\boldsymbol{u}_p := [u_{p,1}, \ldots, u_{p,N}]^T$, it is possible to show that (16) admits the following closed-form solution:

$$[\chi_{p,1}^{(i)}, \ldots, \chi_{p,N}^{(i)}]^T = \frac{\boldsymbol{x}_p^{(i)} + c^{-1}\boldsymbol{u}_p^{(i-1)}}{\|\boldsymbol{x}_p^{(i)} + c^{-1}\boldsymbol{u}_p^{(i-1)}\|_2} \times \max\left(\|\boldsymbol{x}_p^{(i)} + c^{-1}\boldsymbol{u}_p^{(i-1)}\|_2 - c^{-1}\lambda, 0\right). \quad (17)$$

The solution in (17) has the form of a *shrinkage* operator, which means that if the ℓ_2 norm of the vector $\boldsymbol{x}_p^{(i)} + c^{-1}\boldsymbol{u}_p^{(i-1)}$ is less than $c^{-1}\lambda$, then $\chi_{p,n}^{(i)} = 0$ for $n = 1, \ldots, N$; otherwise, the solution is obtained by reducing the ℓ_2 norm of $\boldsymbol{x}_p^{(i)} + c^{-1}\boldsymbol{u}_p^{(i-1)}$ by a factor $c^{-1}\lambda$. Thus, the computational burden for solving (9) scales linearly with N.

Exploiting the properties of the ADMoM [5, p. 253], the following proposition can be proved.

Proposition 1. *For any* $c > 0$, $\boldsymbol{\chi}_{1:N}^{(0)}$, *and* $\mathbf{u}_{1:N}^{(0)}$, *for* $i = 1, 2, \ldots$, *the iterates* $\mathbf{x}_{1:N}^{(i)}$ *in* (11), $\boldsymbol{\chi}_{1:N}^{(i)}$ *in* (17), *and* $\mathbf{u}_{1:N}^{(i)}$ *in* (10) *are all convergent; i.e.,* $\lim_{i\to\infty} \mathbf{x}_{1:N}^{(i)} = \mathbf{x}_{1:N}^{\text{LKS}}$.

Proposition 1 asserts that the ADMoM-based algorithm described in (11), (17), and (10) converges to the minimum of (6). Furthermore, the computational burden it incurs scales linearly with N per iteration; thus, long data intervals can be processed.

B. Selection of λ

The selection of λ critically affects performance. In fact, the larger λ is selected, the sparser $\mathbf{x}_{1:N}^{\text{LKS}}$ is obtained, but with a larger bias too. Defining $\mathbf{d}_n := \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{y}_n$, and $\mathbf{D} := [\mathbf{d}_1, \ldots, \mathbf{d}_N] = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_P]^T$, the following result can be established.

Proposition 2. *If* $\lambda \geq \lambda_{max} := \max_{p=1,\ldots,P} \|\boldsymbol{h}_p\|_2$, *then* $\mathbf{x}_{1:N}^{\text{LKS}} = \mathbf{0}_{P\times N}$.

Proposition 2 asserts that choosing λ greater than or equal to $\lambda_{\max}$ ensures that the LKS estimates are deterministically zero. This suggests choosing $\lambda \ll \lambda_{\max}$ to avoid this trivial solution. Extensive simulations have further indicated that selecting λ to be a small percentage of $\lambda_{\max}$ leads to a satisfactory tradeoff between sparsity and bias. Notice that $\lambda_{\max}$ can be evaluated from the available data.

Remark 1: The sparsity-aware estimation of time-varying signals advocated herein can be extended to encompass *group* sparsity. In this case, one has to assume that the unknown vector can be divided into G groups; that is, $\mathbf{x}_n = [\mathbf{x}_{n,1}^T, \ldots, \mathbf{x}_{n,G}^T]^T \in \mathbb{R}^P$ with $\mathbf{x}_{n,g} := [x_{1,n,g}, \ldots, x_{P_g,n,g}]^T \in \mathbb{R}^{P_g}$, where P_g is the dimension of the g-th group of variables, and $\sum_{g=1}^{G} P_g = P$. A pertinent model for group sparsity is

$$\mathbf{x}_{n+1,g} = \mathbf{T}_{n,g}\mathbf{x}_{n,g} + \mathbf{e}_{n,g} \quad (18)$$

with $\mathbf{e}_{n,g} \sim \mathcal{N}(\mathbf{0}_{P_g}, \mathbf{Q}_{n,g})$, and $\mathbf{Q}_{n,g}, \mathbf{T}_{n,g} \in \mathbb{R}^{P_g\times P_g}$, if the g-th group is active; or, $\mathbf{x}_{n,g} = \mathbf{0}_{P_g}$ for $n = 1, \ldots, N$, otherwise. Defining the $P \times P$ block-diagonal matrices $\mathbf{T}_n = \text{bdiag}(\mathbf{T}_{n,1}, \ldots, \mathbf{T}_{n,G})$, a sparsity-aware estimator tailored for this case can be obtained by substitut-

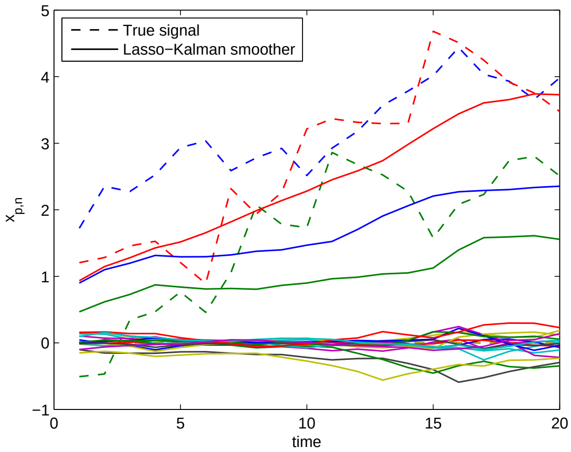

Fig. 3. LKS estimates versus the true signal: LKS outperforms the KS; furthermore, the inactive component amplitudes are reduced at the expense of bias in the estimates of the active components.

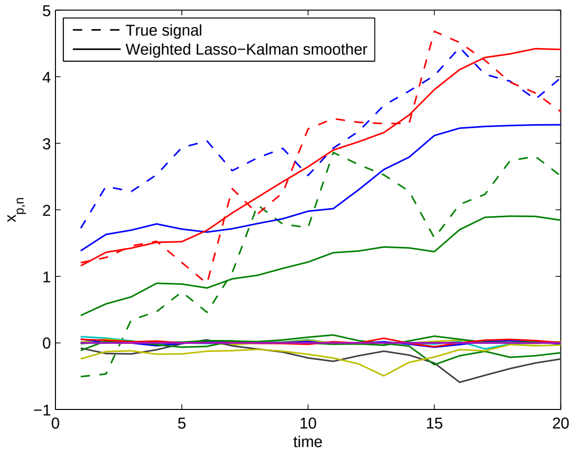

Fig. 4. WLKS estimates versus the true signal: WLKS outperforms the KS and LKS; the non-active components amplitudes, as well as the bias of the active components, are reduced.

ing the penalty function $\lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}x_{p,n}^2}$ is (6) with $\lambda\sum_{g=1}^{G}\sqrt{\sum_{p=1}^{P_g}\sum_{n=1}^{N}x_{p,n,g}^2}$.

IV. Simulation Results

A test case is now presented to assess the performance of the LKS estimator in tracking time-varying sparse signals. The simulation parameters are set to: $N = 20$, $P = 20$, $M = 1$, $e_{p,n} \sim \mathcal{N}(0, 0.5)$, and $\beta_{p,n} = 1$ for $n = 1, \ldots, N$, $p = 1, \ldots, P$, $\mu_{p,1} = 0$, $\sigma_{p,1}^2 = 1$ for $p = 1, \ldots, P$, $\mathbf{H}_n = [h_{1,n}, \ldots, h_{P,n}]$ such that $h_{p,n} \sim \mathcal{N}(0, 1)$, and $\mathbf{R}_n = 0.1$ (since $M = 1$, the matrices $\mathbf{H}_n \in \mathbb{R}^{M\times P}$ and $\mathbf{R}_n \in \mathbb{R}^{M\times M}$ become, respectively, a P-dimensional row vector and a scalar). The unknown active support is $\mathcal{S}_x = \{1, 2, 3\}$. Estimates of the sparsity-agnostic KS are depicted in Fig. 1. Clearly, the performance of the KS is poor, and its estimates are not sparse. Fig. 2 depicts estimates of a genie-aided (GA) KS that knows in advance the true support $\mathcal{S}_x$, and thus, it implements standard KS to estimate *only* the active components. As expected, the GAKS estimates closely capture the true signal variations. Next, the LKS is tested with $\lambda = \lambda_{\max}10^{-3}$, and its estimates are depicted in Fig. 3. Clearly, most of the inactive components have been shrunk to zero, and the non-zero components track the true signal well. Notice, however, that the active component amplitudes are under-estimated. A means of reducing this bias is to resort to *weighted* norms [7]. To this end, consider the regularization term $\lambda\sum_{p=1}^{P}w_p\sqrt{\sum_{n=1}^{N}x_{p,n}^2}$ replacing $\lambda\sum_{p=1}^{P}\sqrt{\sum_{n=1}^{N}x_{p,n}^2}$ in (6), with $w_p = \left(\sum_{n=1}^{N}(x_{p,n}^{\text{KS}})^2\right)^{-1/2}$ for $p = 1, \ldots, P$. In practice, the KS is run first and then the weights $\{w_p\}$ are obtained from its estimates. Figure 4 depicts state estimates with the resulting weighted (W)-LKS, which decreases the bias and outperforms the LKS and the sparsity-agnostic KS.

V. Concluding Remarks

Sparsity-aware fixed-interval smoothing of sparse time-varying signals was considered in this paper. Since the KS is sparsity-agnostic and the optimal smoother is computationally infeasible, a novel smoother, termed LKS, was developed by regularizing the KS cost with a sparsity-encouraging term. It was shown that such regularization supports recursive estimation of the LKS solution that was obtained using the ADMoM algorithm. Finally, to reduce the bias introduced due to regularization, the WLKS was introduced to process the KS estimates and obtain suitable weights. Numerical tests confirmed that the novel smoothers outperform the KS in estimating sparse time-varying signals. Future research directions will pursue application of sparsity-aware smoothers to nonlinear target tracking, and comparisons with existing alternatives.

References

[1] B. D. Anderson and J. B. Moore, *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.

[2] D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," in *Proc. IEEE Intl. Conf. Acoust. Sp. Sig. Proc.*, Taipei, Taiwan, Apr. 2009.

[3] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley-Interscience Publications, 2001.

[4] J.-A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radios by exploiting sparsity," *Proc. of 42nd Asilomar Conf.*, Pacific Grove, CA, Oct. 26-29, 2008.

[5] D. P. Bertsekas, and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed., Athena Scientific, 1999.

[6] E. Candès, and M. B. Wakin, "An introduction to compressive sampling," *IEEE Sign. Proc. Mag.*, vol. 25, pp. 21–30, Mar. 2008.

[7] E. Candès, M. Wakin and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2007.

[8] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[9] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Statist. Soc. Ser. B*, vol. 68, no. 1, pp. 49–67, 2006.

[10] N. Vaswani, "Kalman filtered compressed sensing," *Proc. of Intl. Conf. Image Proc*, San Diego, CA, Oct. 2008.