A distributed Kalman smoother

Bradley M. Bell * **Gianluigi Pillonetto** **

* *Applied Physics Laboratory*
University of Washington, Seattle, USA
(e-mail: bradbell@apl.washington.edu)
** *Department of Information Engineering*
University of Padova, Padova, Italy
(e-mail: giapi@dei.unipd.it)

Abstract: Kalman smoothers obtain state estimates in a system with stochastic dynamics and measurement noise. We consider the smoothing problem in a distributed setting, present a cooperative smoothing algorithm for Gauss-Markov linear models, and provide a convergence analysis for the algorithm. An extension of the algorithm that maximizes the likelihood with respect to a sequence of state vectors subject to inequality constraints, e.g. positivity conditions, is also described. Finally, a numerical experiment regarding cubic spline regression is included to test the new approach.

Keywords: decentralized and cooperative optimization; parallel processing; constrained optimization; interior point methods; estimation theory; smoothing splines

1. INTRODUCTION

Distributed estimation on graphical models, such as Markov networks, is an important subject of research in many scientific fields such as bioinformatics and signal processing (Jordan, 1998; Rue and Held, 2005). Popular methods for inference on such structures include e.g. junction tree algorithms and variational methods as described in (Wainwright and Jordan, 2008) as well as Markov chain Monte Carlo methods (Gilks et al., 1996; Hastings, 1970).

We consider simple graphical models representing state space models underlying Kalman smoothing. In particular, we study the problem of reconstructing the state of a Gauss-Markov linear system via a decentralized optimization scheme. The problem of distributed filtering and smoothing has been studied in the context of a sensor network by assuming that each node can measure a state which is common to all the measurements devices, e.g. a scalar signal evolving over time (Alriksson and Rantzer, 2006; Schizas et al., 2007; Speranzon et al., 2006). The aim of each node is to achieve a local state estimate by suitably combining its own measurements and the estimates coming from its neighbors. Such approaches rely upon e.g. consensus strategies (Carli et al., 2007) and optimization techniques such as alternating-direction methods and augmented Lagrangians (Schizas et al., 2008). Conversely, we assume that the nodes can be ordered in time or space and have access to noisy measurements relative to different but correlated states. The goal of each node is to compute the minimum variance estimate of its state conditional on the overall data acquired by the network. The problem could be solved by resorting to the well known Mayne-Fraser two-filter or the Rauch-Tung-Striebel algorithm, which are however essentially serial in nature (Gelb, 1974). An interesting parallel smoothing scheme can instead be found in (Tewfik et al., 1990). However, in our context this algorithm, after a series of parallel processing of data in subintervals, would require a serial and expensive exchange of information between all the nodes of the network. The distributed smoother presented in this paper is different since it is an iterative scheme where information exchange is limited to nodes which are close to each other. In addition, we show how interior point approaches (Fiacco and McCormick, 1990; Bell et al., 2009) can be exploited to provide an efficient solution also to the problem of distributed smoothing with states subject to inequality constraints, e.g. positivity conditions.

The paper is organized as follows. In section 2, the estimation problem is stated. In section 3, the distributed smoothing algorithm is presented. In section 4, the inequality constrained version is discussed. In section 5, we provide a convergence analysis of the algorithm by deriving explicitly the matrix which regulates the dynamics of the error, i.e. the distance between estimate and minimum variance one as a function of iterations number. In section 6 we discuss the dynamics of our smoother in presence of node failures in the network. In section 7, a numerical example regarding the reconstruction of a function and its derivative via cubic smoothing splines is used to test the new algorithm and the theoretical findings. Some conclusions then end the paper.

2. PROBLEM DESCRIPTION

We are given the information in Fig. 1 where the subscript k denotes the node index. The functions h_k and g_k in Fig. 1 are affine; i.e., there are know matrices $H_k \in \mathbf{R}^{m(k)\times n}$ and $G_k \in \mathbf{R}^{n\times n}$ such that

$$h_k(x_k) = h_k(0) + H_k x_k$$
$$g_k(x_{k-1}) = g_k(0) + G_k x_{k-1}$$

We use x_k to denote the state at index $k = 1, \ldots N$. There is no state x_0, $G_0 = 0$, and $g_0(x_0)$ plays the role of

n	dimension of state vector x_k
N	Number of nodes; i.e. $k = 1, \ldots, N$
$m(k)$	number of components in measure z_k
z_k	measurement, $z_k \in \mathbf{R}^{m(k)}$
p	Number of groups working in parallel, $p \in \mathbf{Z}_+$
Q_k	Transition variance, $Q_k \in \mathbf{R}^{n\times n}$
R_k	Measurement variance, $R_k \in \mathbf{R}^{m(k)\times m(k)}$
g_k	Transition function, $g_k : \mathbf{R}^n \to \mathbf{R}^n$
h_k	Measurement function, $h_k : \mathbf{R}^n \to \mathbf{R}^{m(k)}$
ε	Algorithm convergence criteria

Fig. 1. Known values that define the problem

the initial state estimate for x_1. We assume the following statistical model:

$$\begin{aligned} x_k &= g_k(x_{k-1}) + w_k, \; w_k \sim \mathbf{N}[0, Q_k] \\ z_k &= h_k(x_k) + v_k, \quad v_k \sim \mathbf{N}[0, R_k] \end{aligned}$$

where $\{v_k\} \cup \{w_k\}$ are all mutually independent. We define the following terms

$$\begin{aligned} J &= (N-1)/p \\ K(j) &= 1 + (j-1)J \qquad (j = 1, \cdots, p+1) \\ L(j) &= 1 + J/2 + (j-1)J \quad (j = 0, \cdots, p+1) \end{aligned}$$

Let X denote the entire state sequence $\{x_k : k = 1, \ldots, N\}$. We use the following notation for specific subsets of the state sequence:

$$\begin{aligned} X_K &= \{x_{K(1)}, x_{K(2)}, \ldots, x_{K(p+1)}\} \\ X_L &= \{x_{L(1)}, x_{L(2)}, \ldots, x_{L(p)}\} \\ X_{K(j,k)} &= \{x_{K(j)}, x_{K(k)}\} \\ X_{L(j,k)} &= \{x_{L(j)}, x_{L(k)}\} \end{aligned}$$

In the case where $j = 0$ ($k = p+1$), the corresponding $x_{L(j)}$ ($x_{L(k)}$) is not included in the set $X_{L(j,k)}$. Let $Z = \{z_k : k = 1, \ldots, N\}$. We introduce a different type of subsets of Z 'contained' between, and not including, the indices j and k,

$$\begin{aligned} Z_{K(j,k)} &= \{z_{K(j)+1}, \cdots, z_{K(k)-1}\} \\ Z_{L(j,k)} &= \{z_{L(j)+1}, \cdots, z_{L(k)-1}\} \end{aligned}$$

As before, indices that are not between one and N are not included; i.e., the cases $j = 0$ and $k = p+1$ have the following definitions for $Z_{L(j,k)}$

$$\begin{aligned} Z_{L(0,k)} &= \{z_1, z_2, \cdots, z_{L(k)-1}\} \\ Z_{L(j,p+1)} &= \{z_{L(j)+1}, \cdots, z_{N-1}, z_N\} \end{aligned}$$

3. ALGORITHM

In the sequel, vectors are column vectors and $\mathbf{E}[\cdot]$ denotes the expectation operator. In addition, given the random vectors Y and W, $\mathbf{V}[Y, W]$ is their covariance; i.e.,

$$\mathbf{V}[Y, W] = \mathbf{E}\left[(Y - \mathbf{E}[Y])(W - \mathbf{E}[W])^T\right]$$

and we use the notation $\mathbf{V}[Y] = \mathbf{V}[Y, Y]$. The distributed smoothing algorithm is defined below:

(1) Set $\ell = 0$ and X_K^0 as follows, for $j = 1, \ldots, p+1$,
$$x^0_{K(j)} = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}]$$
(2) Compute $X_L^{\ell+1} = \mathbf{E}[X_L \mid Z \,,\, X_K = X_K^\ell]$.
(3) Compute $X_K^{\ell+2} = \mathbf{E}[X_K \mid Z \,,\, X_L = X_L^{\ell+1}]$.
(4) If $|x^{\ell+2}_{K(j)} - x^\ell_{K(j)}| \le \varepsilon$ for all $j = 1, \ldots, p+1$,
$$\text{return} \quad \mathbf{E}[x_k \mid Z \,,\, X_K = X_K^{\ell+2}]$$
as the state estimate for $k = 1, \ldots, N$.
(5) Set $\ell = \ell + 2$ and go to step 2

It follows from the Markov property for the state sequence that

$$\begin{aligned} &\mathbf{E}[x_{L(j)} \mid Z \,,\, X_K] \\ &\quad = \mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)} \,,\, X_{K(j,j+1)}] \end{aligned} \tag{1}$$

$$\begin{aligned} &\mathbf{E}[x_{K(j)} \mid Z \,,\, X_L] \\ &\quad = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)} \,,\, X_{L(j-1,j)}] \end{aligned} \tag{2}$$

In view of the above equations, the expectation in Step 2 can be computed using p parallel procedures. Each of these parallel procedures solves a smoothing problem over a set of J nodes where the state at the boundaries of the set of nodes is given. The computational complexity of each of these parallel procedures is $O(Jn^3)$. A similar conclusion holds for the expectations in Step 3 and Step 4.

4. CONSTRAINED SMOOTHING

Because the functions h_k and g_k are affine, the minimum variance algorithm above also minimizes the following sum of squares function

$$\begin{aligned} S(X) = \frac{1}{2} \sum_{k=1}^{N} \; & [z_k - h_k(x_k)]^{\mathrm{T}} R_k^{-1} [z_k - h_k(x_k)] \\ & + [x_k - g_k(x_{k-1})]^{\mathrm{T}} Q_k^{-1} [x_k - g_k(x_{k-1})] \end{aligned}$$

Because it is convex, optimizing this function is equivalent to solving the first order necessary condition $\nabla S(X) = 0$. Using (Bell et al., 2009, eq. 6), $\nabla S(X) = 0$ is equivalent to, for $k = 1, \ldots, N$

$$\begin{aligned} 0 = \quad & [x_k - g_k(0) - G_k x_{k-1}]^{\mathrm{T}} Q_k^{-1} \\ & - [z_k - h_k(0) - H_k x_k]^{\mathrm{T}} R_k^{-1} H_k \\ & - [x_{k+1} - g_{k+1}(0) - G_{k+1} x_k]^{\mathrm{T}} Q_{k+1}^{-1} G_{k+1} \end{aligned}$$

which is a set of linear N equations with respect to x_k for $k = 1, \ldots N$. Defining the terms

$$\begin{aligned} b_k &= Q_k^{-1} g_k(0) + H_k^{\mathrm{T}} R_k^{-1} [z_k - h_k(0)] - G_{k+1}^{\mathrm{T}} Q_{k+1}^{-1} g_{k+1}(0) \\ A_k &= -Q_k^{-1} G_k \quad , \quad A_{N+1} = 0 \\ C_k &= Q_k^{-1} + H_k^{\mathrm{T}} R_k^{-1} H_k + G_{k+1}^{\mathrm{T}} Q_{k+1}^{-1} G_{k+1} \end{aligned} \tag{3}$$

Proposition 4.1. Suppose that A_k, B_k, C_k are defined as in equation (3). It follows that the distributed algorithm above solves the following linear equations for the values of $x_1, \ldots, x_N$:

$$b_k = A_k x_{k-1} + C_k x_k + A_{k+1}^{\mathrm{T}} x_{k+1} \qquad (k = 1, \ldots, N)$$

Suppose that for $k = 1, \ldots, N$, $U_k \in \mathbf{R}^{n\times n}$ is symmetric positive definite and $u_k \in \mathbf{R}^{n\times n}$ is an arbitrary vector. We define the extended measurement function and measure vector by

$$\bar{h}_k(x_k) = \begin{pmatrix} x_k \\ h_k(x_k) \end{pmatrix} \,,\; \bar{z}_k = \begin{pmatrix} u_k \\ z_k \end{pmatrix} \,,\; \bar{R}_k = \begin{pmatrix} U_k & 0 \\ 0 & R_k \end{pmatrix}$$

Proposition 4.2. Suppose that A_k, b_k, C_k are defined as in equation (3). If we apply the distributed smoother algorithm to the extended measure model corresponding to $\bar{h}_k$, $\bar{z}_k$, and $\bar{R}_k$ defined above, it will solve the following linear equations for the values of $x_1, \ldots, x_N$:

$$b_k + U_k^{-1} u_k = A_k x_{k-1} + [C_k + U_k] x_k + A_{k+1}^{\mathrm{T}} x_{k+1}$$

for $k = 1, \ldots, N$

Suppose that we want to solve the affine inequality constrained smoother problem

$$\begin{aligned} &\text{minimize} \quad S(X) \quad \text{w.r.t } X \\ &\text{subject to } f_k(x_k) \le 0 \;\; (k = 1, \ldots, N) \end{aligned}$$

where $f_k : \mathbf{R}^n \to \mathbf{R}^p$ is affine; i.e., $f(x_k) = f(0) + F_k x_k$ for some $F_k \in \mathbf{R}^{p\times n}$. This is equivalent to (Bell et al., 2009, problem 11) (where $x_k = 0$ in problem 11 and y_k in problem 11 corresponds to x_k above). In Bell et al. (2009) it has been shown, by using interior point methods, that the solution of this problem calls for solving the following linear equations for the value of $x_1, \ldots, x_N$:

$$a_k = A_k x_{k-1} + [C_k + F_k D_k F_k] x_k + A_{k+1}^{\mathrm{T}} x_{k+1}$$

for $k = 1, \ldots, N$, where $a_k \in R^n$ is arbitrary and D_k is a diagonal matrix with positive elements along the diagonal. (Using the notation in Bell et al. (2009) this will enable us to multiply the term $[C + B^{\mathrm{T}} D(s/u) B]^{-1}$ in (Bell et al., 2009, equation 22) by an arbitrary vector.) Setting $U_k = F_k D_k F_k$ and $u_k = U_k(a_k - b_k)$ we see by the previous proposition that we can in fact use the distributed smoother to solve the linear equations necessary for the constrained problem.

5. CONVERGENCE ANALYSIS

The following lemma reports well known formulas about joint Gaussian vectors, see e.g. (Anderson and Moore, 1979).

Lemma 1. If Y, W are jointly Gaussian random variates, it holds that

$$\mathbf{E}(Y|W) = \mathbf{E}(Y) + \mathbf{V}(Y,W)\mathbf{V}(W)^{-1}[W - \mathbf{E}(W)] \quad (4)$$

$$\mathbf{V}(Y|W) = \mathbf{V}(Y) - \mathbf{V}(Y,W)\mathbf{V}(W)^{-1}\mathbf{V}(W,Y) \quad (5)$$

For future developments, it is useful to define the following notation

$$\begin{aligned}\Xi_j &= \mathbf{V}\left(x_{K(j)}, X_{L(j-1,j)} \mid Z_{L(j-1,j)}\right) \\ &\quad \times \mathbf{V}\left(X_{L(j-1,j)} \mid Z_{L(j-1,j)}\right)^{-1}, \quad j = 1, \ldots, p+1 \\ \Pi_j &= \mathbf{V}\left(x_{L(j)}, X_{K(j,j+1)} \mid Z_{K(j,j+1)}\right) \\ &\quad \times \mathbf{V}\left(X_{K(j,j+1)} \mid Z_{K(j,j+1)}\right)^{-1}, \quad j = 1, \ldots, p\end{aligned}$$

These matrices can be computed using (5) (for large values of J it may be more efficient and computationally stable to use (Bell and Pillonetto, 2008, corollary 7)).

We use $\delta_K^\ell = X_K^\ell - \mathbf{E}[X_K|Z]$ to denote the error for even values of ℓ. We also use $\delta_{K(j,k)}^\ell$ for the column vector

$$\delta_{K(j,k)}^\ell = (\delta_{K(j)}^\ell, \delta_{K(k)}^\ell)^{\mathrm{T}}$$

We use a similar notation for δ_L^ℓ.

We define matrices $E_1, E_2, \ldots, E_{p+1}$, where $E_j \in \mathbf{R}^{n\times 2n}$ if $j \in \{1, p+1\}$ and $E_j \in \mathbf{R}^{n\times 3n}$ otherwise. The block-tridiagonal matrix Γ associated with $\{E_j\}$ is such that E_1 specifies the nonzero-entries in the first n rows, E_2 the nonzero-entries in the second n rows and so on. The next proposition characterizes the error dynamics of the distributed smoothing algorithm.

Proposition 5.1. Let Γ be the block-tridiagonal matrix associated with the blocks

$$\begin{aligned}E_1 &= \Xi_1 \Pi_1 \\ E_j &= \Xi_j \begin{pmatrix} \Pi_{j-1} & 0_{n\times n} \\ 0_{n\times n} & \Pi_j \end{pmatrix} \qquad j = 2, \ldots, p \\ E_{p+1} &= \Xi_{p+1} \Pi_p\end{aligned}$$

Then, it holds that

(1) the error dynamics of the distributed smoothing algorithm at the nodes $\{K(j)\}$ are regulated by the equation

$$\delta_K^{\ell+2} = \Gamma \delta_K^\ell \quad (6)$$

(2) Γ is asymptotically stable, i.e. all its eigenvalues are inside the complex unit circle

Proof: Let's start focusing on the error propagation at a generic node $K(j)$, with $1 < j < p+1$, when ℓ is 0 or an even number. Step 2 of the algorithm computes

$$X_L^{\ell+1} = \mathbf{E}[X_L \mid Z\ , X_K = X_K^\ell]$$

by means of local computations; see (1). Exploiting (4), with $Y = x_{L(j)}$ and $W = X_{K(j,j+1)}$ and both conditional on $Z_{K(j,j+1)}$, the linear projection (1) admits the following decomposition

$$\begin{aligned}&\mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}\ , X_{K(j,j+1)}] = \mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}] \\ &+\mathbf{V}(x_{L(j)}, X_{K(j,j+1)} \mid Z_{K(j,j+1)}) \\ &\times \mathbf{V}^{-1}(X_{K(j,j+1)} \mid Z_{K(j,j+1)}) \\ &\times \left(X_{K(j,j+1)} - \mathbf{E}[X_{K(j,j+1)} \mid Z_{K(j,j+1)}]\right)\end{aligned}$$

Using our definition for Π_j, this becomes

$$\begin{aligned}&\mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}\ , X_{K(j,j+1)}] = \mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}] \\ &+\Pi_j \left(X_{K(j,j+1)} - \mathbf{E}[X_{K(j,j+1)} \mid Z_{K(j,j+1)}]\right)\end{aligned} \quad (7)$$

Taking the expected value $\mathbf{E}[\cdot|Z]$ of both sides of the equation above, we obtain

$$\begin{aligned}&\mathbf{E}[x_{L(j)} \mid Z] = \mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}] \\ &+\Pi_j \left(\mathbf{E}[X_{K(j,j+1)} \mid Z] - \mathbf{E}[X_{K(j,j+1)} \mid Z_{K(j,j+1)}]\right)\end{aligned} \quad (8)$$

In the places where $X_{K(j,j+1)}$ is a fixed value, substitute $x_{K(j)} = x_{K(j)}^\ell$, and $x_{K(j+1)} = x_{K(j+1)}^\ell$. Note that for these choices,

$$x_{L(j)}^{\ell+1} = \mathbf{E}[x_{L(j)} \mid Z_{K(j,j+1)}\ , X_{K(j,j+1)}]$$

Now, with this choice, subtracting equation (8) from (7), we obtain

$$\begin{aligned}x_{L(j)}^{\ell+1} - \mathbf{E}[x_{L(j)} \mid Z] &= \Pi_j \left(X_{K(j,j+1)}^\ell - \mathbf{E}[X_{K(j,j+1)} \mid Z]\right) \\ \delta_{L(j)}^{\ell+1} &= \Pi_j \delta_{K(j,j+1)}^\ell\end{aligned} \quad (9)$$

We now move to consider Step 3 of the algorithm given by (2). We have

$$\begin{aligned}&\mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}\ , X_{L(j-1,j)}] = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}] \\ &+\mathbf{V}(x_{K(j)}, X_{L(j-1,j)} \mid Z_{L(j-1,j)}) \\ &\times \mathbf{V}^{-1}(X_{L(j-1,j)} \mid Z_{L(j-1,j)}) \\ &\times \left(X_{L(j-1,j)} - \mathbf{E}[X_{L(j-1,j)} \mid Z_{L(j-1,j)})\right)\end{aligned}$$

Using our definition for Ξ_j, this becomes

$$\begin{aligned}&\mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}\ , X_{L(j-1,j)}] = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}] \\ &+\Xi_j \left(X_{L(j-1,j)} - \mathbf{E}[X_{L(j-1,j)} \mid Z_{L(j-1,j)})\right)\end{aligned} \quad (10)$$

Taking the expected value $\mathbf{E}[\cdot|Z]$ of both sides of the equation above, we obtain

$$\begin{aligned}&\mathbf{E}[x_{K(j)} \mid Z] = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}] \\ &+\Xi_j \left(\mathbf{E}[X_{L(j-1,j)} \mid Z] - \mathbf{E}[X_{L(j-1,j)} \mid Z_{L(j-1,j)}]\right)\end{aligned} \quad (11)$$

In the places where $X_{L(j-1,j)}$ is a fixed value, substitute $x_{L(j-1)} = x_{L(j-1)}^{\ell+1}$, and $x_{L(j)} = x_{L(j)}^{\ell+1}$. Note that now for these choices,

$$x_{K(j)}^{\ell+2} = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}\ , X_{L(j-1,j)}]$$

Now, with this choice, subtracting equation (11) from (10), we obtain

$$x^{\ell+2}_{K(j)} - \mathbf{E}[x_{K(j)} \mid Z] = \Xi_j \left(X^{\ell+1}_{L(j-1,j)} - \mathbf{E}[X_{L(j-1,j)} \mid Z] \right)$$
$$\delta^{\ell+2}_{K(j)} = \Xi_j \delta^{\ell+1}_{L(j-1,j)}$$

Thus, using equation (9), we conclude that

$$\delta^{\ell+2}_{K(j)} = \Xi_j \begin{pmatrix} \Pi_{j-1} & 0 \\ 0 & \Pi_j \end{pmatrix} \begin{pmatrix} \delta^{\ell}_{K(j-1,j)} \\ \delta^{\ell}_{K(j,j+1)} \end{pmatrix}$$

Note that, for $1 < j < p+1$, the block matrix above is size $n \times 4n$. But since the values $\delta^{\ell}_{K(j)}$ are repeated, one can replace the block matrix by a $n \times 3n$ matrix (as is done in the definition of E_j). Considering $j = 1$ and $j = p+1$ as special cases (where the matrix above is $n \times 2n$) equation (6) is immediately obtained.

Viewing the algorithm as maximizing the likelihood function, it is a special version of coordinated gradient method, see e.g. (Bertsekas and Tsitsiklis, 1997; Luenberger, 2003), which is guaranteed to converge to $E[X|Z]$ for any initial point. Hence, all the eigenvalues of Γ must be less than one (Kailath, 1979).

6. SMOOTHER DYNAMICS IN PRESENCE OF NODE FAILURES

In this subsection, we think of Γ as a matrix function in place of a fixed matrix. To be more specific, we have

$$\Gamma : \{1, 2, \ldots, s\} \mapsto \mathbf{R}^{n(p+1) \times n(p+1)} \tag{12}$$

where the scalar s represents the number of possible scenarios, i.e. the number of configurations of the sensor network subject to possible node failures.

Let's start considering a simple situation where $s = 2$. In the first scenario all the nodes are assumed to work and we let $\Gamma(1)$ describe the error dynamics so that

$$\delta^{\ell+2}_{K1} = \Gamma(1)\delta^{\ell}_{K1}$$

where δ^{ℓ}_{K1} is the distance between X^{ℓ}_K and $\hat{X}_{K1} := \mathbf{E}[X_K|Z]$. In the second situation we assume that the i-th node does not work and that, for the sake of simplicity, does not belong to $\{K(j)\} \bigcup \{L(j)\}$. Let δ^{ℓ}_{K2} denote the distance between X^{ℓ}_K and the minimum variance estimate which, since z_i is not available, is given by

$$\hat{X}_{K2} := \mathbf{E}[X_K|Z \backslash z_i]$$

Thus, for a suitable matrix $\Gamma(2)$, error dynamics are

$$\delta^{\ell+2}_{K2} = \Gamma(2)\delta^{\ell}_{K2}$$

Let's now assume that at any iteration of the algorithm the probability that node i works is equal to a. By defining

$$d(1) := 0, \qquad d(2) = \hat{X}_{K2} - \hat{X}_{K1}$$

the model which describes the evolution of δ^{ℓ}_{K1} becomes

$$\begin{aligned} \delta^{\ell+2}_{K1} &= \Gamma(\xi_\ell) \left[\delta^{\ell}_{K1} + d(\xi_\ell) \right] - d(\xi_\ell) \\ &= \Gamma(\xi_\ell)\delta^{\ell}_{K1} + e(\xi_\ell) \end{aligned} \tag{13}$$

where

$$e(\xi_\ell) := \Gamma(\xi_\ell)d(\xi_\ell) - d(\xi_\ell) \tag{14}$$

and $\{\xi(\ell)\}$, $\ell = 0, 2, 4, \ldots$, are independent random variables which may assume values 1 with probability a and 2 with probability $1 - a$.

A straightforward generalization of the situation described above for a generic integer s allows us to conclude that, in presence of node failures, the dynamics of the distance between X^{ℓ}_K and $\mathbf{E}[X_K|Z]$ are described by the following Markov chain

$$\delta^{\ell+2}_{K1} = \Gamma(\xi_\ell)\delta^{\ell}_{K1} + e(\xi_\ell) \tag{15}$$

where $\{\xi(\ell)\}$, $\ell = 0, 2, 4, \ldots$, are independent random variables taking values on $\{1, 2, \ldots, s\}$ while $e : \{1, 2, \ldots, s\} \mapsto \mathbf{R}^{n(p+1)}$ is a suitable deterministic map.

7. NUMERICAL EXAMPLE

We consider the problem of estimating in a distributed way the derivative of an unknown function f from a finite set of noisy measurements of f. Each measurement is taken by distinct nodes. In particular, let $X_1(t)$ denote the derivative of the unknown function while $X_2(t)$ is its value. Our prior model for f is given by the stochastic differential equation (see (Oksendal, 2003) or (De Nicolao and Ferrari Trecate, 2003))

$$\mathbf{d}X(t) = SX(t)\,\mathbf{d}t + T\,\mathbf{d}B(t) \tag{16}$$

where $B(t)$ is Brownian motion (its derivative is white noise) and

$$S = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \; T = \begin{pmatrix} 10^{-3} \\ 0 \end{pmatrix}$$

This model provides the basis for a Bayesian interpretation of cubic smoothing splines (Wahba, 1990).

For $s \leq t$, we use $\mathbf{V}(t|s)$ to indicate the covariance of $X(t)$ given the value of $X(s)$ which, prior to knowing the system output measurements, satisfies the differential Lyapunov equation (see (Jazwinski, 1970, page 133 equation 4.138))

$$\begin{aligned} \mathbf{V}(s|s) &= 0 \\ \partial_t \mathbf{V}(t|s) &= S\mathbf{V}(t|s) + \mathbf{V}(t|s)S^{\mathrm{T}} + TT^{\mathrm{T}}. \end{aligned}$$

For our particular choice of S and T, we have

$$\mathbf{V}(t|s) = 10^{-3} \begin{pmatrix} (t-s) & \dfrac{(t-s)^2}{2} \\ \dfrac{(t-s)^2}{2} & \dfrac{(t-s)^3}{3} \end{pmatrix}.$$

The distance between sampling points where each sensor is located is denoted by Δt. Thus, the transition model for $k > 1$ is represented by $g_k : \mathbf{R}^2 \to \mathbf{R}^2$ and $Q_k \in \mathbf{R}^{2\times 2}$ where

$$g_k(x_{k-1}) = \begin{pmatrix} x_{1,k-1} \\ x_{2,k-1} + x_{1,k-1}\Delta t \end{pmatrix}$$
$$Q_k = 10^{-3} \begin{pmatrix} \Delta t & \dfrac{\Delta t^2}{2} \\ \dfrac{\Delta t^2}{2} & \dfrac{\Delta t^3}{3} \end{pmatrix}.$$

The initial state estimate is given by

$$g_1(x_0) = X(t_1), \qquad Q_1 = \begin{pmatrix} 100 & 0 \\ 0 & 10 \end{pmatrix}$$

The measurement variance σ^2 is known and direct measurements of $X_2(t)$ are represented by $h_k : \mathbf{R}^2 \to \mathbf{R}^1$ and $R_k \in \mathbf{R}^{1\times 1}$ where

$$h_k(x_k) = x_{2,k}\ , \ R_k = \sigma^2\ .$$

The specifications for the example are completed by the following choices:

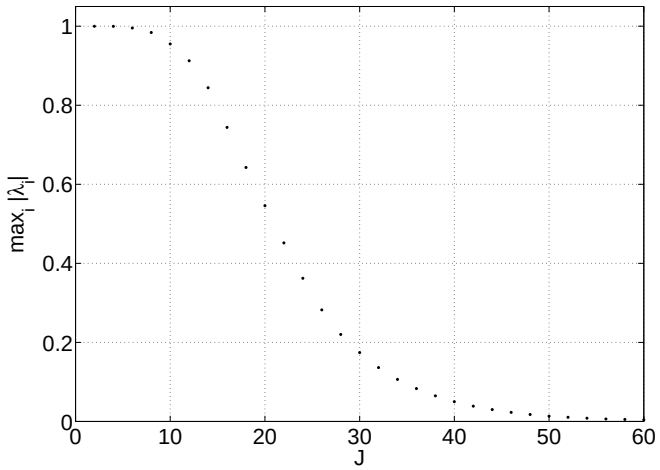

Fig. 2. Cubic smoothing spline example: maximum absolute value of the eigenvalues $\{\lambda_i\}$ of matrix Γ regulating error dynamics as a function of J defining the blocks size.

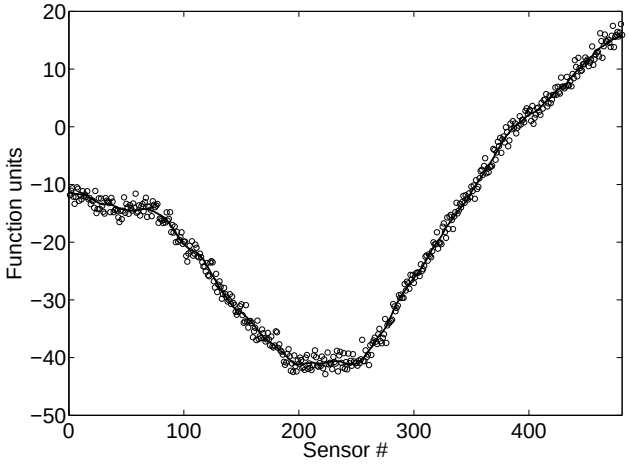

Fig. 3. Cubic smoothing spline example: true function f (solid line) and noisy measurements (circles).

N	$= 480$	number of measurements
Δt	$= 1$	spacing between nodes
σ^2	$= 1$	measurement noise variance
v_k	$\sim \mathbf{N}(0, R_k)$	simulated measurement noise
z_k	$= h_k[X(t_k)] + v_k$	simulated measurement value

Given such specifications, Fig. 2 reports the maximum absolute value of the eigenvalues $\{\lambda_i\}$ of the matrix Γ as a function of J which defines the size of the sensor blocks working in parallel. As a matter of fact, the developed analysis makes available a clear picture on the error dynamics also pointing out the importance of establishing a good trade off between level of parallelism and convergence rate of the iterative algorithm. For instance, when the value of J is low, say $J \leq 8$, a node is stressed with very high frequency to produce a new estimate of its own state and to send it to its adjacent node. However, even if this choice leads to a high level of parallelism, Fig. 2 suggests the need of a very large number of iterations for obtaining an acceptable level of accuracy in the estimate. In fact, the maximum modulus of the eigenvalue is very close to 1. The situation changes when J increases and a good trade-off appears between 16 and 30 where eigenvalues vary from 0.7 to 0.2.

To corroborate the theoretical analysis, let's consider a realization of f drawn from the prior and plotted in Fig. 3 (solid line). The aim is to reconstruct the derivative of such function from the noisy measurements displayed as circles in the same figure. Fig. 4 reports the estimates of the derivative of f obtained by the distributed Kalman smoother when J is 2 (top panels), 16 (middle) or 30 (bottom), for ℓ equal to 0,2 and 4. It is apparent that, in practice, when $J = 2$ the algorithm will never converge to the minimum variance estimate in reasonable time. On the other hand, when $J = 16$, already for $\ell = 2$ the algorithm returns an estimate sufficiently close to the optimum.

8. CONCLUSIONS

We have considered smoothing of Gauss-Markov linear systems via distributed optimization. In the context of a sensor network, our problem amounts to assuming that any node has access to noisy measurements of different but correlated states. Then, the aim is to reconstruct the overall state sequence in a cooperative way, by taking advantage of all the data obtained by the network. A parallel smoothing scheme has been presented together with a convergence analysis. The latter points out the importance, in the algorithm design, of finding the right trade off between parallelism and rate of convergence towards the optimal estimate. Extension of the algorithm to the case of state sequence subject to inequality constraints has been also provided. Future developments of this work may regard the extension of the method, and relative convergence analysis, to more general and complex graphical models.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack

REFERENCES

Alriksson, P. and Rantzer, A. (2006). Distributed kalman filtering using weighted averaging. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto.*

Anderson, B.D.O. and Moore, J.B. (1979). *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, N.J., USA.

Bell, B., Burke, J., and Pillonetto, G. (2009). An inequality constrained nonlinear kalman-bucy smoother by interior point likelihood maximization. *Automatica*, 45, 25–33.

Bell, B. and Pillonetto, G. (2008). Approximating the bayesian inverse for nonlinear dynamical systems. *Journal of Physics: Conference Series 124.*

Bertsekas, D. and Tsitsiklis, J. (1997). *Parallel and distributed computation: numerical methods.* Athena Scientific, Belmont, Massachusets.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2007). Distributed kalman filtering based on consensus strategies. *Proc. of IEEE Conf. on Decision and Control, New Orleans, USA.*

De Nicolao, G. and Ferrari Trecate, G. (2003). Regularization networks for inverse problems: a state space approach. *Automatica*, 39, 669–676.

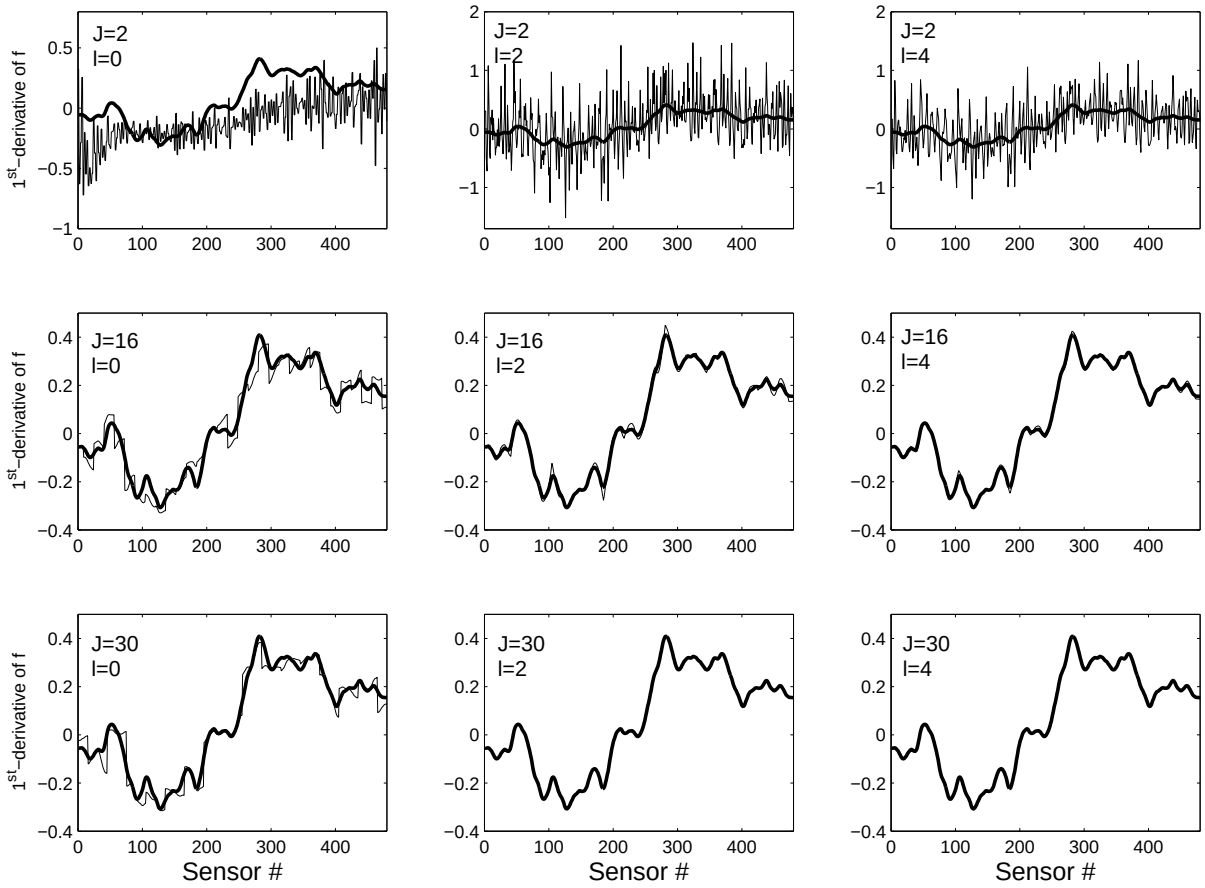

Fig. 4. Reconstruction of the first derivative of f: minimum variance estimate (thick line), estimates from the distributed smoother (thin line) as a function of iteration number ℓ and size of nodes working in parallel (defined by J).

Fiacco, A. and McCormick, G. (1990). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM.

Gelb, A. (1974). *Applied Optimal Estimation*. The M.I.T. Press, Cambridge, MA.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov chain Monte Carlo in Practice*. London: Chapman and Hall.

Hastings, W. (1970). Monte carlo sampling methods using markov chain and their applications. *Biometrika*, 57, 97–109.

Jazwinski, A.H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press, New York.

Jordan, M. (1998). *Learning in Graphical Models*. The M.I.T. Press, Cambridge, MA.

Kailath, T. (1979). *Linear Systems*. Prentice Hall.

Luenberger, D. (2003). *Linear and Nonlinear Programming*. Kluwer.

Oksendal, B. (2003). *Stochastic Differential Equations*. Springer.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall.

Schizas, I., Giannakis, G., Roumeliotis, S.I., and Ribeiro, A. (2007). Anytime optimal distributed kalman filtering and smoothing. In *Workshop on Statistical Signal Processing, 2007. SSP 07. IEEE/SP 14th*, 368372.

Schizas, I., Giannakis, G., Roumeliotis, S.I., and Ribeiro, A. (2008). Consensus in ad hoc wsns with noisy links - part ii: Distributed estimation and smoothing of random signals. *IEEE Transactions on Signal Processing*, 56, 1650–1666.

Speranzon, A., Fischione, C., and Johansson, K. (2006). Distributed and collaborative estimation over wireless sensor networks. In *Proceedings of the IEEE Conference on Decision and Control (CDC06)*, 10251030.

Tewfik, A., Willsky, A., and Levy, B. (1990). Parallel smoothing. *Systems and Control Letters*, 14, 253–259.

Wahba, G. (1990). *Spline models for observational data*. SIAM, Philadelphia.

Wainwright, M.J. and Jordan, M.I. (2008). Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1, 1–305.