Distributed Inequality Constrained Kalman Smoother

S. Del Favero[1], G. Pillonetto[1] and B.M. Bell[2]

Abstract— **This paper, which is the sequel of [1], considers smoothing of Gauss-Markov linear systems via distributed optimization. As an application we consider the distributed estimation problem over sensor networks and assume that each 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 collected by the network.**

In this paper, the convergence analysis in [1] is deepened, pointing out the importance in the algorithm design, of finding the right trade off between parallelism and convergence rate. Moreover an extension of the algorithm to the case of state sequence subject to inequality constraints is also provided. In particular, we show that the same algorithmic architecture and communication protocol used in the unconstrained case can be exploited in the constrained scenario. Hence, the network can efficiently include in the estimation process relevant a priori information on the state, such as nonnegativity. Numerical experiments regarding the distributed reconstruction of a function via spline regression is used to test the new approach.

I. Introduction

Kalman smoothers obtain state estimates in a system with stochastic dynamics and noisy measurements. Kalman smoothing can be addressed as a convex optimization problem, namely the maximization of the posterior of the state sequence conditional on the measurements. As recently shown in [2], additional constraints on the process can be efficiently handled by constrained optimization techniques relying on interior point methods. In this way, relevant a priori information on the state, such as nonnegativity, can be taken into account. This may lead to significant improvement in the estimation process, avoiding physically meaningless results.

In this work, we consider the unconstrained and inequality constrained smoothing problem in a distributed setting. In fact, the recently increased availability of cheap sensing devices makes more and more concrete the possibility of having large sensor networks spread over vast areas for monitoring and estimation purposes. Nevertheless, these cheap devices have tight communication and computation constraints that require the development of efficient algorithms to distribute the computations among nodes using only local communications. In this context, it is often assumed that each node collects noisy measurements of the same common state evolving over time [3], [4], [5]. The aim of each node is to obtain a local state estimate by suitably combining its own measurements and the estimates coming from its neighbors. However, the analysis of information dynamics which allows numerous cooperating units to work properly, is challenging [6]. Recent approaches rely upon e.g. consensus strategies [7] and optimization techniques such as alternating-direction methods and augmented Lagrangians [8]. Here, we instead assume that the nodes can be ordered in time or space and have access to noisy measurements relative to different but correlated states.

More generally, inference on graphical models such as Markov or Bayesian networks has been recently subject to much research in many different scientific fields such as bioinformatics and signal processing [9], [10]. Some popular methods for inference on such structures include e.g. junction tree algorithms and variational methods as described in [11] as well as Markov chain Monte Carlo methods [12], [13]. In this paper, we focus on simple graphical models defined by state space models underlying Kalman smoothing.

For what concerns the smoothing problem, in [1] an iterative cooperative smoothing algorithm for Gauss-Markov linear state space models has been proposed. The unconstrained case was addressed and an analysis of the estimation performance was also carried out, proving that the proposed scheme asymptotically returns the optimal estimate.

Simulations in [1] showed that a crucial role in the algorithm design is played by the choice of a parameter, denoted as J. In fact, augmenting the value of J increases the communications needed at each iteration but may improve dramatically the convergence rate of the algorithm. The contribution of this paper, which is a sequel of [1], is twofold. Firstly, we better investigate the role played by J in determining the speed of convergence of the algorithm to the optimal estimate. To do so, we restrict our analysis to a simple yet significant case, i.e. a scalar random walk. The convergence rate is derived as a function of J and of the process statistics such as model noise variance and measurement noise variance.

For what concerns the second contribution of this paper, while in [1] the extension of the distributed smoother to the inequality constrained scenario was only mentioned, here we obtain a numerically robust algorithm which implements the interior point method described in [2] in a distributed fashion. In particular, this goal is obtained by deriving a novel distributed matrix inversion algorithm that takes advantage of the block-tridiagonal structure of the matrices to be inverted and requires only local communications. The new approach is tested on a simulated example regarding the distributed reconstruction of a monotonic function via smoothing splines.

A. Paper Organization

The paper is organized as follows. Section II provides the problem statement and introduces the notation used in the sequel. In section III, for sake of clarity, we brefly recall the distributed unconstrained smoothing algorithm presented in [1] and summarize the convergence analysis of the algorithm. In the case where the state evolves according to a random walk, the convergence rate is explicitly derived as a function

[1] Department of Information Engineering, University of Padova, Italy.
[2] Applied Physics Laboratory University of Washington, Seattle, USA
E-mail: simone.delfavero@dei.unipd.it, giapi@dei.unipd.it
bradbell@apl.washington.edu

ISBN 978-963-311-370-7

of the design parameters of the algorithm. In section IV, the inequality constrained version is worked out while in section V the algorithm is used to reconstruct a monotonic function in a distributed way. Conclusions then end the paper.

II. Problem Description and Notation

Consider a graphical model underlying Kalman smoothing where the N non-observable states are denoted by $\{x_k\}$ while $\{z_k\}$ are vectors containing the noisy measurements. An example is given in Fig. 1 where $N = 9$. In the scenario of sensor networks, the components of z_k are the measurements collected by the k-th node whose aim is to estimate x_k. The a priori joint density of $\{x_k\}$ is defined by the following recursive equation:

$$x_k = G_k x_{k-1} + w_k, \qquad w_k \sim \mathcal{N}[0, Q_k], \tag{1}$$

where $G_k \in \mathbf{R}^{n\times n}$ and $w_k \in \mathbf{R}^n$ is zero-mean white Gaussian noise with autocovariance $Q_k \in \mathbf{R}^{n\times n}$. The measurements model is

$$z_k = H_k x_k + v_k, \qquad v_k \sim \mathcal{N}[0, R_k], \tag{2}$$

where $H_k \in \mathbf{R}^{m(k)\times n}$ and v_k is zero-mean white normal noise of autocovariance $R_k \in \mathbf{R}^{m(k)\times m(k)}$. We also assume that $\{v_k\}$ and $\{w_k\}$ are all mutually independent. Our aim is to obtain an efficient smoothing algorithm where the node in charge of estimating x_k can exchange information only with the adjacent nodes, i.e. with the nodes handling the states x_{k-1} and x_{k+1}, so as to obtain a high level of parallelism.

The algorithm proposed in [1] divides the N nodes in two types of overlapping groups working in parallel. The first partition consists of p groups which, just for ease of notation, are assumed to contain the same number of nodes, $J + 1$, with J an even integer so that $N = Jp + 1$. The indices

$$K(j) = 1 + (j-1)J, \qquad j = 1, \cdots, p+1$$

define the boundaries of this first kind of groups. More precisely, $K(j)$ is a pointer both to the first node of the j-th group and to the last node of the $(j-1)$-th group. An example is reported in the top panel of Fig. 1.

Let X denote the entire state sequence $\{x_k : k = 1, \ldots, N\}$. It is useful also to define

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

whose meaning is graphically illustrated in the middle panel of Fig. 1.

The second partition contains $p+1$ groups whose boundaries are defined by

$$L(j) = 1 + J/2 + (j-1)J, \qquad j = 1, \cdots, p.$$

Notice that these are the bottom nodes of the groups present in the previous partition, as also illustrated in the top panel of Fig. 1. As in the previous case, it is useful to define

$$\begin{aligned} X_L &= \{x_{L(1)}, x_{L(2)}, \ldots, x_{L(p)}\} \\ X_{L(j,j+1)} &= \{x_{L(j)}, x_{L(j+1)}\}, \qquad j = 1, \cdots, p-1 \\ X_{L(0,1)} &= \{x_{L(1)}\} \\ X_{L(p,p+1)} &= \{x_{L(p+1)} \end{aligned}$$

Finally, let $Z = \{z_k : k = 1, \ldots, N\}$. Then, define the following subsets of Z 'contained' between, and not including, the indexes j and k

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

The bottom panel of Fig. 1 reports an example of these subsets.

III. Unconstrained 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 presented in [1] is the following:

Algorithm 1:

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 $j = 1, \ldots, p+1$,
$$\text{return} \qquad \mathbf{E}[x_k \mid Z\ ,\ X_K = X_K^{\ell+2}]$$
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

$$\mathbf{E}[x_{L(j)}|Z, X_K] = \mathbf{E}[x_{L(j)}|Z_{K(j,j+1)}, X_{K(j,j+1)}] \tag{3}$$
$$\mathbf{E}[x_{K(j)} \mid Z\ ,\ X_L] = \mathbf{E}[x_{K(j)} \mid Z_{L(j-1,j)}\ ,\ X_{L(j-1,j)}] \tag{4}$$

In view of the above equations, the expectation in Step 2 can be computed using p parallel procedures.

A. Convergence analysis

Let us define the following notation for $j = 1, \ldots, p$

$$\begin{aligned} \Xi_j &= \mathbb{V}\left(x_{K(j)}, X_{L(j-1,j)}|Z_{L(j-1,j)}\right) \mathbb{V}\left(X_{L(j-1,j)}|Z_{L(j-1,j)}\right)^{-1} \\ \Pi_j &= \mathbb{V}\left(x_{L(j)}, X_{K(j,j+1)}|Z_{K(j,j+1)}\right) \mathbb{V}\left(X_{K(j,j+1)}|Z_{K(j,j+1)}\right)^{-1} \end{aligned} \tag{5}$$

Moreover, let us use $\delta_K^\ell = X_K^\ell - \mathbb{E}[X_K|Z]$ to denote the error for even values of ℓ. and the matrices $E_1, E_2, \ldots, E_p$, where $E_j \in \mathbf{R}^{n\times 2n}$ if $j \in \{1, p\}$ and $E_j \in \mathbf{R}^{n\times 3n}$ otherwise, where

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

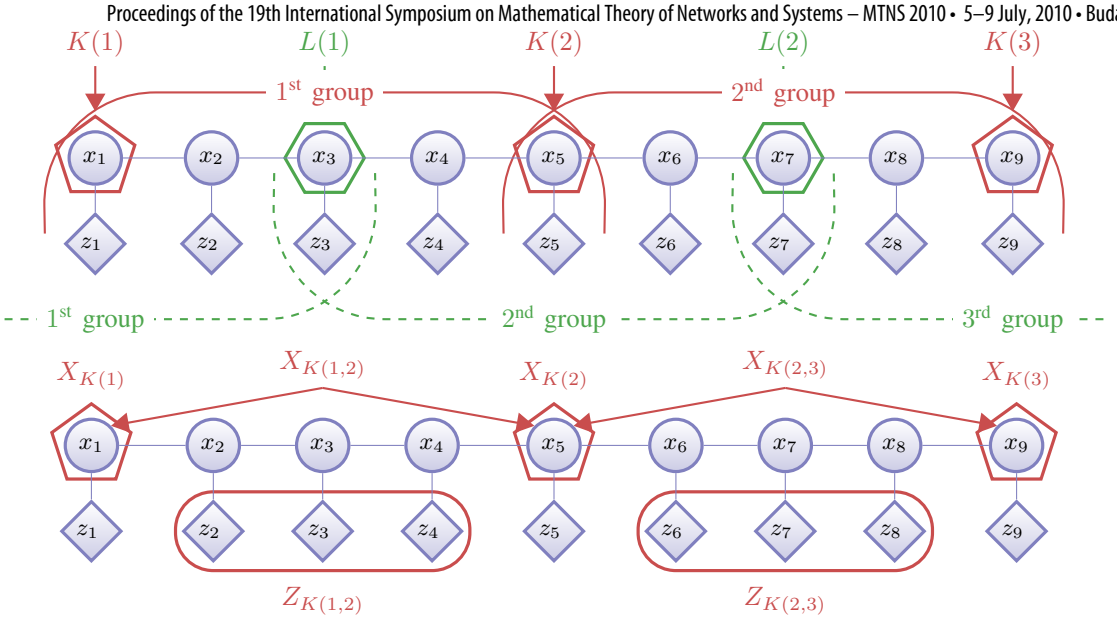

Fig. 1. Graphical Model of the spatially distributed random process under consideration (equations (1, 2)) for $N = 9$ and group division for $p = 2$, $J = 4$. In this example $K(j)$ may assume the values 1,5 and 9 and $L(j)$ may assume the values 3,7 .

Then, we define the block-tridiagonal matrix R associated with $\{E_i\}$ 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 following result is taken from [1].

Proposition 3.1: It holds that

1) the error dynamics of the distributed smoothing algorithm at the nodes $\{K(j)\}$ are regulated by the equation
$$\delta_K^{\ell+2} = R\delta_K^{\ell} \tag{6}$$
2) R is asymptotically stable, i.e. all its eigenvalues are inside the complex unit circle

Moreover, simulations in [1] showed that a crucial role in the algorithm design is played by the parameter J. In fact, augmenting the value of J increases the communications needed at each iteration but may improve dramatically the convergence rate of the algorithm. Next section is devoted to the analysis of this fact.

B. Specialization to the random walk model

In this subsection, we apply our convergence analysis to a special, yet significant, model given by the scalar random walk. In particular, (1,2) become

$$\begin{aligned} x_{k+1} &= x_k + w_k, \qquad x_0 = x^0 \\ z_k &= x_k + v_k, \qquad k = 1, \dots N, \end{aligned} \tag{7}$$

where $\mathbb{V}(w_k) = \lambda$, $\mathbb{V}(v_k) = \sigma$ and the initial condition x^0 is zero mean Gaussian of variance $\mathbb{V}(x^0)$. Then, the following result holds.

Proposition 3.2: As J goes to infinity and p remains constant, all the eigenvalues of R exhibit the same asymptotic behavior, i.e.

$$\lambda(R) \sim \text{const} \cdot \nu^{-J}, \qquad \nu = 1 + \frac{\lambda}{2\sigma} + \frac{1}{2}\sqrt{\frac{4\lambda}{\sigma} + \frac{\lambda^2}{\sigma^2}} \geq 1.$$

The above proposition is a demonstration of the usefulness of the convergence analysis relative to the distributed smoother. It shows that the estimator dynamics may be strongly nonlinear as a function of the size J of the blocks working in parallel. Remarkably, for large J, in the random walk case a small increment in J, associated with a small increment in the communication cost, may cause an exponential improvement of the convergence rate.

Notice that for more complex models, closed form/asymptotic expression of the eigenvalues, like that derived above, are generally hard to obtain. However, an exact convergence analysis can always be obtained since the eigenvalues of R can be computed in a numerically robust way exploiting its block-tridiagonal structure.

IV. Inequality constrained distributed Smoothing

A. The inequality constrained distributed algorithm

Define

$$\begin{aligned} S(X) = \frac{1}{2}\sum_{k=1}^{N} \quad & [z_k - Hx_k]^{\mathrm{T}} R_k^{-1} [z_k - Hx_k] \\ & + [x_k - Gx_{k-1}]^{\mathrm{T}} Q_k^{-1} [x_k - Gx_{k-1}] \end{aligned} \tag{8}$$

Then, Kalman smoothing can be interpreted as a convex optimization problem and one has

$$\mathbb{E}[X \mid Z] = \arg\min_X S(X) \tag{9}$$

This is an unconstrained quadratic problem that can be solved in a distributed way by means of Algorithm 1 reported in Section III. Now, let's consider the inequality constrained version of (9) defined by

$$\begin{aligned} &\hat{\mathbb{E}}[X|Z, \{B_k x_k + b_k \leq 0\}_{k=1}^N] := \\ &\arg\min_X S(X) \quad \text{s.t.} \quad B_k x_k + b_k \leq 0, \quad k = 1, \dots, N \end{aligned} \tag{10}$$

where $B_k \in \mathbf{R}^{l \times n}$ and $b_k \in \mathbf{R}^l$. Notice that the symbol $\hat{\mathbb{E}}$ above does not represent a minimum variance estimator but, by definition, returns a maximum a posteriori estimate conditional on Z and the inequality constraints on the state. Let's consider two kinds of problems obtained from (10)

where the conditioning variables include also either X_K or X_L. Using the Markov property, one obtains

$$\hat{\mathbb{E}}[\, x_{L(j)} \mid Z\ ,\ X_K, \{B_k x_k + b_k \leq 0\}_{k=1}^{N}] = \quad (11)$$
$$\hat{\mathbb{E}}[x_{L(j)} \mid\ Z_{K(j,j+1)}\ ,\ X_{K(j,j+1)}, \{B_k x_k + b_k \leq 0\}_{k=K(j)}^{K(j+1)}]$$

and

$$\hat{\mathbb{E}}[x_{K(j)} \mid Z\ ,\ X_L, \{B_k x_k + b_k \leq 0\}_{k=1}^{N}] = \quad (12)$$
$$\hat{\mathbb{E}}[x_{K(j)} \mid Z_{L(j-1,j)}\ ,\ X_{L(j-1,j)}, \{B_k x_k + b_k \leq 0\}_{k=L(j-1)}^{L(j)}]$$

We notice that, as in the unconstrained case, the problems (11,12) can be solved using p or $p+1$ parallel procedures that require only the exchange of local data. Hence, the following pseudo-code defines an inequality constrained distributed smoothing algorithm where the solution of the problem (10) is obtained solving a sequence of these subproblems.

Algorithm 2:

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

In the next subsection it is shown that the solutions to the problems (11,12) can be obtained by exploiting the same algorithmic architecture and communication protocol used in the unconstrained scenario.

B. Computing $\hat{\mathbb{E}}$ using interior point methods

For a given value of the iteration counter ℓ, we need to compute[1]

$$\hat{\mathbb{E}}\left[x_{L(j)} \,\middle|\, \begin{array}{c} Z_{K(j,j+1)}, \\ X_{K(j,j+1)} = X_{K(j,j+1)}^\ell, \\ \{B_k x_k + b_k \leq 0\}_{k=K(j)}^{K(j+1)} \end{array} \right] \quad (13)$$

that corresponds to obtaining

$$\begin{array}{cc} \arg\min & S(X) \\ \text{w.r.t} & x_{K(j)}, \ldots, x_{K(j+1)} \\ \text{s.t.} & X_{K(j,j+1)} = X_{K(j,j+1)}^\ell \\ & B_k x_k + b_k \leq 0, \\ & k = K(j)+1, \ldots, K(j+1)-1 \end{array} \quad (14)$$

[1]We restrict the attention to solving (13) since the problem of computing $\hat{\mathbf{E}}[x_{K(j)} | Z_{L(j-1,j)}, X_{L(j-1,j)} = X_{L(j-1,j)}^{l+1}, \{B_k x_k + b_k \leq 0\}_{k=L(j-1)}^{L(j)}]$ is analogous.

with $S(X)$ defined in (8). Except for the inequality constraints, the objective reduces to

$$\begin{aligned} &\tilde{S}(\tilde{X}) \\ &= \frac{1}{2}\left[\sum_{k=K(j)}^{K(j+1)-1} \begin{array}{c} [z_k - Hx_k]^{\mathrm{T}} R_k^{-1} [z_k - Hx_k] \\ + [x_k - Gx_{k-1}]^{\mathrm{T}} Q_k^{-1} [x_k - Gx_{k-1}] \end{array} \right] + \\ &\quad + [z_k - Hx_k]^{\mathrm{T}} R_k^{-1} [z_k - Hx_k] \\ &\quad + [x_{K(j)+1} - Gx_{K(j)}^\ell]^{\mathrm{T}} Q_k^{-1} [x_{K(j)+1} - Gx_{K(j)}^\ell] \\ &+ [x_{K(j+1)}^\ell - Gx_{K(j+1)-1}]^{\mathrm{T}} Q_{K(j+1)}^{-1} [x_{K(j+1)}^\ell - Gx_{K(j+1)-1}] \end{aligned} \quad (15)$$

where $\tilde{X} = [x_{K(j)+1}, \ldots, x_{K(j+1)-1}]$.
Define

$$\begin{aligned} A_k &= -Q_k^{-1} G_k \\ C_k &= Q_k^{-1} + H_k^{\mathrm{T}} R_k^{-1} H_k + G_{k+1}^{\mathrm{T}} Q_{k+1}^{-1} G_{k+1}, \end{aligned} \quad (16)$$

and

$$\begin{aligned} d_k &= H_k^{\mathrm{T}} R_k^{-1} z_k \qquad \text{for } k = K(j)+2, \ldots, K(j+1)-2 \\ d_{K(j)+1} &= H_k^{\mathrm{T}} R_k^{-1} z_k + x_{K(j)}^\ell G^T Q_{K(j)+1}^{-1} \\ d_{K(j+1)-1} &= H_k^{\mathrm{T}} R_k^{-1} z_k + x_{K(j+1)}^\ell Q_{K(j+1)}^{-1} G \end{aligned} \quad (17)$$

Then, simple computations show that the Hessian and the gradient of $\tilde{S}(\tilde{X})$ computed at the origin are given respectively by

$$C = \quad (18)$$
$$\begin{pmatrix} C_{K(j)+1} & A_{K(j)+2}^T & 0 & & \\ A_{K(j)+2} & C_{K(j)+2} & A_{K(j)+3}^T & 0 & \\ & \ddots & \ddots & \ddots & \\ & & A_{K(j+1)-2} & C_{K(j+1)-2} & A_{K(j+1)-1}^T \\ & & & A_{K(j+1)-1} & C_{K(j+1)-1} \end{pmatrix}$$

and

$$d = (d_{K(j)+1}^{\mathrm{T}}, \ldots, d_{K(j+1)-1}^{\mathrm{T}})^{\mathrm{T}} \quad (19)$$

Thus, we obtain that the optimization problem (14) is equivalent to obtaining

$$\begin{array}{cc} \arg\min & \frac{1}{2}\tilde{X}^{\mathrm{T}} C \tilde{X} + d^T \tilde{X} \\ \text{subject to} & B\tilde{X} + b \leq 0 \end{array} \quad (20)$$

where

$$\begin{aligned} B &= \text{blkdiag}(B_{K(j)+1}, \ldots, B_{K(j+1)-1}) \\ b &= [b_{K(j)+1}^T, \ldots, b_{K(j+1)-1}^T]^T \end{aligned}$$

We are now in a position to exploit the interior point approach developed in [2] to solve the problem (20). Here, we just recall that the interior point technique applies a damped Newton's method to a relaxation of the Karush-Kuhn-Tucker (KKT) conditions. In our case, the relaxed subproblem, containing a log barrier, is given by:

$$\begin{array}{c} \text{minimize } \frac{1}{2}\tilde{X}^{\mathrm{T}} C \tilde{X} + d^T \tilde{X} \\ -\mu \sum_{i=1}^{l \times (K(j+1)-K(j)-2)} \log(s_i) \\ \text{w.r.t} \quad (\tilde{X}, s) \quad \text{s.t.} \quad s + b + B\tilde{X} = 0 \end{array} \quad (21)$$

where μ denotes the relaxation parameter and s is the vector containing the slack variables.
Let u denote the vector of Lagrange multipliers associated with problem (21). From [2] we know that the minimizer can be obtained by computing a sequence of Newton steps that exhibit a particular structure. In fact, to compute the increments $(\Delta u, \Delta\tilde{X}, \Delta s)$ around the current value of $(u, \tilde{X}, s)$, one has to successively solve the following equations

$$Cz = B^{\mathrm{T}}u + d, \tag{22}$$
$$\left[BC^{-1}B^{\mathrm{T}} + D(s/u)\right]\Delta u = [\mu(e/u) + b - Bz] \tag{23}$$
$$C(\Delta\tilde{X} + \tilde{X}) = d + B^{\mathrm{T}}(u + \Delta u) \tag{24}$$
$$\Delta s + s = -b - B(\Delta\tilde{X} + \tilde{X}) \tag{25}$$

where e is the vector (of proper dimension) with all its components equal to one, $D(w)$, with w a vector, is a diagonal matrix with entry $D_{ii} = w_i$ and the i-th entry of the vector s/u is s_i/u_i.
To evaluate Δu, let us reformulate (23) using the matrix inversion lemma as follows

$$\Delta u = D(s/u)^{-1}[\mu(e/u) + b - Bz] - D(s/u)^{-1}B^{T}\cdot$$
$$\cdot\,[C + B^{T}D(s/u)^{-1}B]^{-1}BD(s/u)[\mu(e/u) + b - Bz]$$

It is interesting to notice that multiplications by $D(s/u)^{-1}$, $D(s/u)^{-1}B^T$ and $BD(s/u)$ can be performed completely in parallel by the nodes, due to the block-diagonal nature of D and B. In addition, $C + B^TD(s/u)^{-1}B$ is the sum of a block-tridiagonal and a block-diagonal matrix. Now, recall that, in the unconstrained case, each group of nodes solves its smoothing subproblem inverting the block-tridiagonal matrix C. In the inequality constrained scenario, the above equations show that the core of the computational load of the group is still the solution of block-tridiagonal systems (defined by C or $C+B^TD(s/u)^{-1}B$). Hence, the same communication protocol used in the unconstrained case can be used to solve the constrained problem. Numerical issues related to the inversion of the type of block-tridiagonal matrices present in (22-25) are discussed in the next subsection.

C. A new solver for Block-Tridiagonal systems of linear equations

In this section we present an algorithm that solves a linear system of equations described by a block-tridiagonal matrix with M blocks. Suppose for $k = 1, \ldots, M$, $C_k \in \mathbf{R}^{n\times n}$, $e_k \in \mathbf{R}^{n\times\ell}$, $d_k \in \mathbf{B}^{n\times\ell}$, and for $k = 2, \ldots, M$, $A_k \in \mathbf{R}^{n\times n}$. We define the corresponding block tridiagonal system of equations

$$\begin{pmatrix} C_1 & A_2^T & 0 & \cdots & 0 \\ A_2 & C_2 & & & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & & A_{M-1} & C_{M-1} & A_M^T \\ 0 & \cdots & 0 & A_M & C_M \end{pmatrix}\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{M-1} \\ e_M \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{M-1} \\ d_M \end{pmatrix} \tag{26}$$

Subtracting $A_M^TC_M^{-1}$ times row M from row $M-1$, we obtain the following equivalent equation:

$$\begin{pmatrix} C_1 & A_2^T & 0 & \cdots & 0 \\ A_2 & C_2 & & & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & & A_{M-1} & C_{M-1} - A_M^TC_M^{-1}A_M & 0 \\ 0 & \cdots & 0 & A_M & C_M \end{pmatrix}\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{M-1} \\ e_M \end{pmatrix} =$$
$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{M-1} - A_M^TC_M^{-1}d_M \\ d_M \end{pmatrix}$$

Iterate this procedure till the first row of the matrix, using f_k to denote the resulting diagonal blocks, and s_k the corresponding right hand side of the equations; i.e.,

$$\begin{aligned} f_k &= C_k - A_{k+1}^T f_{k+1}^{-1} A_k \\ s_k &= d_k - A_{k+1}^T f_{k+1} d_{k+1} \end{aligned}$$

We obtain the following equivalent system:

$$\begin{pmatrix} f_1 & 0 & \cdots & & 0 \\ A_2 & f_2 & 0 & \cdots & 0 \\ & & \ddots & & \vdots \\ \vdots & & A_{M-1} & f_{M-1} & 0 \\ 0 & & & A_M & C_M \end{pmatrix}\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{M-1} \\ e_M \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{M-1} \\ d_M \end{pmatrix} \tag{27}$$

Thus we obtain the following block tridiagonal extension of Thomas algorithm, [14], :

Algorithm 3:

The inputs to this algorithm are $\{A_k\}$, $\{C_k\}$, and $\{d_k\}$. The output is a sequence $\{e_k\}$ that solves equation (26) and the λ which is equal to the log of the determinant of the block tridiagonal matrix in equation (26).

1) Set $f_M = C_M$ and $s_M = d_M$.
2) For $k = M-1, \ldots, 1$, set $f_k = C_k - A_{k+1}^T f_{k+1} A_k$, $s_k = d_k - A_{k+1}^T f_{k+1}^{-1} d_{k+1}$.
3) Set $e_1 = f_1^{-1}s_1$.
4) For $k = 2, \ldots, M$, set $e_k = f_k^{-1}(s_k - A_k e_{k-1})$.
5) Set $\lambda = \sum_{k-1}^{M} \log\det(f_k)$.

In the following proposition we give a result that is particularly useful when the above presented algorithm is applied to the inversion of the matrix C introduced in (18), in section IV-B.

For $M \in \mathbf{R}^{n\times n}$ we use the notation $|M|$ for the operator norm of the matrix M; i.e.,

$$|M| = \sup\{|Mw| \ : \ w \in \mathbf{R}^n \ , \ |w| = 1\}$$

Proposition 4.1: Consider the block–tridiagonal matrix introduced in (18) where the matrices A_k and C_k are given by

$$\begin{aligned} A_k &= -Q_k^{-1}G_k \\ C_k &= Q_k^{-1} + H_k^T R_k^{-1} H_k + G_{k+1}^T Q_{k+1}^{-1} G_{k+1} \end{aligned}$$

and each $Q_k \in \mathbf{R}^{n\times n}$ is positive definite, each $H_k^T R_k^{-1} H_k \in \mathbf{R}^{n\times n}$ is positive semi-definite and $G_{M+1} = 0$. It follows that $f_k - Q_k^{-1}$ is positive semi-definite for all k. Furthermore, if α is a bound for $|Q_k|$, $|Q_k^{-1}|$, $|H_k^T R_k^{-1} H_k|$, and $|G_k|$, Then the condition number of f_k is bounded by $\alpha^2 + \alpha^6$.

V. Inequality constrained numerical example

We reconsider the function estimation problem described in [1] where spline regression is used. It is equivalent to a Kalman smoothing problem in (1,2) where

$$G_k = \begin{pmatrix} 1 & 0 \\ \Delta t & 1 \end{pmatrix}, Q_k = 10^{-3} \begin{pmatrix} \Delta t & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \frac{\Delta t^3}{3} \end{pmatrix}, \; k > 1$$

with Δt denoting the distance between sampling points. The initial state estimate is defined by

$$x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad G_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad Q_1 = \begin{pmatrix} 100 & 0 \\ 0 & 10 \end{pmatrix}$$
$$H_k = \begin{pmatrix} 0 & 1 \end{pmatrix}, \qquad R_k = \sigma^2, \qquad k \geq 1$$

In comparison with [1], the problem is complicated by the fact that the function has be reconstructed also at time instants for which a measurement is not available, i.e.

$$\begin{cases} H_k = [0\ 1] & k = 1, M+1, 2M+1, \dots \\ H_k = 0 & \text{otherwise} \end{cases}$$

More importantly, the information about positivity and monotonicity of the function is available and needs to be incorporated in the model using the following inequality constraints

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x_k = B_k x_k \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \forall k = 1, \dots, N$$

Let the unknown function be $f(t) = e^{-t^2}$. We report in figure 2 some simulation results. Both the proposed algorithms for unconstrained and constrained smoothing has been used to estimate the function, in a distributed manner, from $N/M = 50$ noisy measurements at $N = 1000$ time instants with $\Delta t = 0.06$. Figure 2 shows a remarkable improvement in the estimation quality enabled by the a priori information on the function constraints.

VI. Conclusions

This paper, which is the sequel of [1], considers smoothing of Gauss-Markov linear systems via distributed optimization. In the context of sensor networks, we assume that each 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 collected by the network.

The convergence analysis in [1] was deepened, pointing out the importance, in the algorithm design, of finding the right trade off between parallelism and convergence rate. Moreover, an extension of the algorithm to the case of state sequence subject to inequality constraints has been also provided. In particular, it has been shown that the same algorithmic architecture and communication protocol relative to the unconstrained case can be used in the constrained scenario. Hence, the network can efficiently include in the estimation process relevant a priori information on the state, such as nonnegativity. Numerical experiments regarding the distributed reconstruction of a function via spline regression have been used to test the new approaches. Future developments of this work will regard the extension of the method, and relative convergence analysis, to more general graphical models.

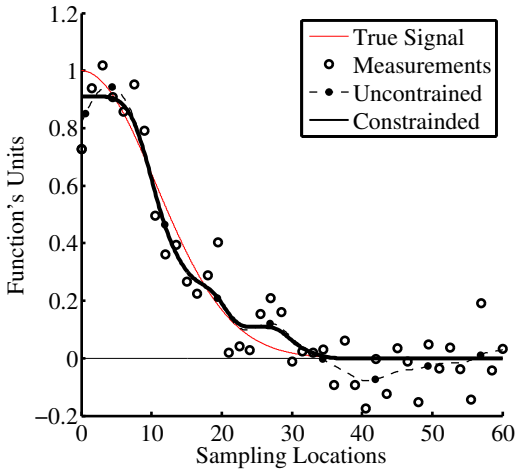

Fig. 2. Function reconstruction using the distributed inequality constrained and unconstrained smoother.

Acknowledgements

This research was partially supported by the Italian Ministry of University and Research through the PRIN Project entitled 'Tecniche ed applicazioni innovative di identificazione e controllo adattativo', by the Progetto di Ateneo CPDA090135/09 funded by the University of Padova and by the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

A. Appendix: Proof of Proposition 3.2.

Notation: We say that $f(J) \sim g(J)$ if $\lim f(J)/g(J) = 1$ for $J \to +\infty$. Moreover we will denote with $\mathscr{C}$ **a generic** constant whose exact amount is not of interest. To avoid cumbersome notations, we allow $\mathscr{C}$ to represent different constants, even in the same formula, since we are only interested to point out their being constant [2].

Proof:

Let us consider the the projector Π_j, defined in (5). Π_j tells us how much an error in the estimate of $x_{K(j)}$ and $x_{K(j+1)}$ affects the estimates of $x_{L(j)}$ given Z and an analogous role is played by Ξ_j. To prove Proposition 3.2 we first give, in the following lemma, an asymptotic results describing the behavior of the projector Π_j and Ξ_j a J goes to infinity. The lemma is proved then in section A-A of this appendix, where also we compute exactly Π_j and Ξ_j.

Lemma 1: For all $j = 1, ..., p$,

$$\Pi_j \sim \nu^{-\frac{J}{2}} \begin{bmatrix} \mathscr{C} & \mathscr{C} \end{bmatrix}^T$$

[2]Some care has to be payed by the reader, since this choice might at first result in some cases a little confusing: for instance $[\mathscr{C} \quad \mathscr{C}]^T$ might represent any vector in $\mathbf{R}^2$, not only vectors of the form $\mathscr{C}[1 \quad 1]$.

Analogously $\Xi_j \sim \nu^{-\frac{J}{2}} \begin{bmatrix} \mathscr{C} & \mathscr{C} \end{bmatrix}^T$ $j = 2, ..., p$ and $\Xi_1 \sim \Xi_{p+1} \sim \mathscr{C}\nu^{-\frac{J}{2}}$.
From lemma 1 it follows that

$$E_j = \Xi_j \begin{bmatrix} \Pi_{j-1} & 0 \\ 0 & \Pi_j \end{bmatrix} \sim \nu^{-\frac{J}{2}} \begin{bmatrix} \mathscr{C} & \mathscr{C} \end{bmatrix} \nu^{-\frac{J}{2}} \begin{bmatrix} \mathscr{C} & \mathscr{C} & 0 \\ 0 & \mathscr{C} & \mathscr{C} \end{bmatrix} \sim \nu^{-J} \begin{bmatrix} \mathscr{C} & \mathscr{C} & \mathscr{C} \end{bmatrix} \quad \forall j = 2, \dots, p$$

and analogously

$$E_1 = \Xi_1 \Pi_1 \sim \nu^{-J} \begin{bmatrix} \mathscr{C} & \mathscr{C} \end{bmatrix} \quad E_{p+1} = \Xi_{p+1} \Pi_p \sim \nu^{-J} \begin{bmatrix} \mathscr{C} & \mathscr{C} \end{bmatrix}$$

The matrix R of equation (6) is then a tridiagonal matrix having all entries $r_{i,j} \sim \mathscr{C}\nu^{-J}$. Then, the proof of Proposition 3.2 follows directly form the following general linear algebra fact

Lemma 2: Consider a matrix A whose entries are function of J, $a_{i,j}(J)$. If $\forall i, j$ there exists a constant $\mathscr{C}_{i,j}$, possibly zero, such that $a_{i,j}(J) \sim \mathscr{C}_{i,j} f(J)$ as J goes to infinity, then also $\lambda(A(J)) \sim \mathscr{C} f(J)$.

A. Proof of the lemma 1

Let us consider the j-th block of nodes and introduce a new nodes index, h, such that $k = K(j) + h$. h therefore describes the position of a node of the block j with respect to the initial node $K(j)$. The model for the states of the j–th block becomes then

$$\begin{aligned} x_{h+1}^j &= x_h^j + w_h^j \\ x_0^j &= x_{K(j)} \qquad\qquad h = 1, \dots, J+1 \\ z_h^j &= x_h^j + v_h^j \end{aligned} \tag{28}$$

Define

$$X^j = [x_1^j, \dots, x_{J+1}^j], \qquad Z^j = [z_1^j, \dots, z_{J+1}^j]$$

Denote with $\lambda_0 = \mathbb{V}(x_{K(j)}) = \mathbb{V}(x^0) + K(j)\lambda$ and let us define the matrix

$$B = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \qquad B^{-1} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \ddots & \ddots & \\ 1 & \dots & 1 & 1 \end{bmatrix}$$

It is easy to see that

$$\mathbb{V}(X^j) = \lambda_0 \mathbb{1}\mathbb{1} + \lambda B^{-1} B^{-T}, \tag{29}$$

where $\mathbb{1} = [1, \dots, 1]$. Using the matrix inversion lemma we obtain then

$$\mathbb{V}(X^j)^{-1} = (B^{-1}(\lambda_0 e_1 e_1^T + \lambda I)^{-1} B^{-T})^{-1}.$$

where with $e_1 \in \mathbf{R}^{J+1}$ we denote the vector $e_1 = [1, 0, \dots 0]$ and we exploited the fact that $B^{-1} e_1 = \mathbb{1}$. Since $(\lambda_0 e_1 e_1^T + \lambda I)$ is a diagonal matrix we get that

$$\mathbb{V}(X^j)^{-1} = -\left(\frac{\lambda_0}{\lambda(\lambda_0 + \lambda)}\right) e_1 e_1^T + \frac{1}{\lambda} B^T B$$

where we used the fact that $B^T e_1 = e_1$. The a–posteriori error variance becomes hence:

$$\mathbb{V}(X^j | Z^j) = (\mathbb{V}(X^j)^{-1} + \frac{1}{\sigma} I)^{-1} = \lambda \cdot$$

$$\cdot \begin{bmatrix} 2 - \frac{\lambda_0}{\lambda_0 + \lambda} + \frac{\lambda}{\sigma} & -1 & & & & & \\ -1 & 2 + \frac{\lambda}{\sigma} & -1 & & & & \\ & -1 & 2 + \frac{\lambda}{\sigma} & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 + \frac{\lambda}{\sigma} & -1 & \\ & & & & -1 & 1 + \frac{\lambda}{\sigma} \end{bmatrix}^{-1}$$

We are therefore interested to compute some elements of the inverse of a tridiagonal matrix of the form:

$$\begin{bmatrix} a & -1 & & & \\ -1 & b & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & b & -1 \\ & & & -1 & c \end{bmatrix}^{-1} \tag{30}$$

where $b \geq 2$, in particular to those in positions:$(1,1)$, $(J+1, J+1)$, $(J+1, 1)$, $(1, J/2)$, $(J/2, J+1)$, $(J/2, J/2)$.

As we will see, the computation of these elements of interest of A^{-1} reduces to the computation of the determinant of the following three $n \times n$ tridiagonal matrices defined below

$$\overline{M_n} = \begin{bmatrix} a & -1 & & \\ -1 & b & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & b & -1 \\ & & & -1 & c \end{bmatrix} \quad M_n = \begin{bmatrix} b & -1 & & \\ -1 & b & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & b & -1 \\ & & & -1 & c \end{bmatrix} \tag{31}$$

$$\widetilde{M_n} = \begin{bmatrix} a & -1 & & \\ -1 & b & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & b & -1 \\ & & & -1 & b \end{bmatrix}$$

One shows that, [15],

$$\det M_n = b \det(M_{n-1}) - \det(M_{n-2})$$

with initial conditions $\det M_1 = c$ and $\det M_2 = bc - 1$.

Hence, defining $\nu = \frac{1}{2}b + \frac{1}{2}\sqrt{b^2 - 4}$, $\nu \geq 1$ and $\eta = \frac{1}{\nu}$, we get

$$\det(M_n) = c_1 \nu^n + c_2 \eta^n \tag{32}$$

where c_1 and c_2 are two suitable constants, so that as n goes to infinity,

$$\det(M_n) \sim \mathscr{C}\nu^n.$$

After some computations [15], one gets also

$$\begin{aligned} \det \overline{M}_n &= a \det(M_{n-1}) - \det(M_{n-2}) \\ &\sim \mathscr{C} a \nu^{n-1} - \mathscr{C}\nu^n \sim \mathscr{C}\nu^n \end{aligned}$$

Analogously one notices that $\det \widetilde{M_n}$ obeys to the following recursion

$$\det \widetilde{M}_n = b \det \widetilde{M}_{n-1} - \det \widetilde{M}_{n-2}$$

with initial conditions $\det \widetilde{M}_1 = a$ and $\det \widetilde{M}_2 = ab - 1$. Therefore $\det \widetilde{M}_n = c_3 \nu^n + c_4 \eta^n$ with $c_1 \neq c_3$, $c_2 \neq c_4$,

since the initial conditions are different. Also in this case it holds that:

$$\det(\widetilde{M}_n) \sim \mathscr{C}\nu^n.$$

Let us evaluate now the elements of interest in A^{-1}. To this aim, first of all, note that

$$\begin{aligned}
\det A = \det \overline{M_{J+1}} &\sim \mathscr{C}\nu^J \\
\mathcal{A}_{1,1} = \det M_J &\sim \mathscr{C}\nu^J \\
\mathcal{A}_{J+1,J+1} = \det \widetilde{M}_J &\sim \mathscr{C}\nu^J \\
\mathcal{A}_{\frac{J}{2}+1,\frac{J}{2}+1} = \det \widetilde{M}_{\frac{J}{2}} \det M_{\frac{J}{2}} &\sim \mathscr{C}\nu^J,
\end{aligned}$$

see [15] for further details. Furthermore

$$\begin{aligned}
\mathcal{A}_{1,J+1} &= -(-1)(-1)^{1+1}(-1)^{J-1} = 1 \\
\mathcal{A}_{1,\frac{J}{2}+1} &= -\det\left[\begin{array}{c|c} \text{Upper Triangular} & \\ \hline & M_{\frac{J}{2}} \end{array}\right] + \\
&\quad \det\left[\begin{array}{c|c} \text{Upper Triangular} & * \\ \hline & \text{Singular Matrix} \end{array}\right] \\
&= (-1)(-1)^{\frac{J}{2}-1}\det(M_{\frac{J}{2}}) \sim \mathscr{C}(-1)^{\frac{J}{2}}\nu^{J/2}.
\end{aligned}$$

Analogously,

$$\mathcal{A}_{\frac{J}{2},J+1} \sim \mathscr{C}(-1)^{J/2}\nu^{J/2-1}.$$

Therefore

$$\begin{aligned}
\Pi_j &= \mathbb{V}(x^j_{J/2+1}, [x^j_1\ x^j_{J+1}]|Z^j)\mathbb{V}^{-1}([x^j_1\ x^j_{J+1}]|Z^j) \\
&= \left[\begin{array}{cc} \dfrac{\det M_{J/2}}{\det\overline{M_{J+1}}} & \dfrac{\det\widetilde{M}_{J/2}}{\det\overline{M_{J+1}}} \end{array}\right] \cdot \left[\begin{array}{cc} \frac{\det M_J}{\det\overline{M_{J+1}}} & \frac{1}{\det\overline{M_{J+1}}} \\ \frac{1}{\det\overline{M_{J+1}}} & \frac{\det\widetilde{M}_J}{\det\overline{M_{J+1}}} \end{array}\right]^{-1} \\
&= \left[\det M_{J/2}\ \det\widetilde{M}_{J/2}\right] \cdot \frac{1}{\det M_J \det\widetilde{M}_J - 1}\left[\begin{array}{cc} \det\widetilde{M}_J & -1 \\ -1 & \det M_J \end{array}\right] \\
&= \frac{1}{\det M_J \det\widetilde{M}_J - 1}\left[\begin{array}{c} \det M_{J/2}\det\widetilde{M}_J - \det\widetilde{M}_{J/2} \\ \det\widetilde{M}_{J/2}\det M_J - \det M_{J/2} \end{array}\right]^T
\end{aligned}$$

which, for $J \to \infty$

$$\Pi_j \sim \mathscr{C}\nu^{-2N}\left[\begin{array}{c} \mathscr{C}\nu^{\frac{3}{2}J} \\ \mathscr{C}\nu^{\frac{3}{2}J} \end{array}\right]^T \sim \nu^{-\frac{J}{2}}\left[\begin{array}{c} \mathscr{C} \\ \mathscr{C} \end{array}\right]^T$$

Since the asymptotic behavior of Π_j is independent of the variance of the initial condition, it is also independent of j and of the index of the first node of the group $K(j)$. Thus:

$$\Xi_j \sim \Pi_j \quad \forall j = 2, \ldots, p.$$

An analogous behavior is observed at the boundaries, $\Xi_1 \sim \Xi_{p+1} \sim \mathscr{C}\nu^{-\frac{J}{2}}$.

B. APPENDIX: PROOF OF PROPOSITION 4.1.

We note that $f_M - Q_M^{-1} = H_M^T R_M^{-1} H_M + G_{M+1}^T Q_{M+1}^{-1} G_{M+1}$ so this conditions holds for $k = M$. We now complete the proof by induction; i.e., suppose $f_{k+1} - Q_{k+1}^{-1}$ is positive semi-definite

$$\begin{aligned}
f_k &= C_k - A_{k+1}^T f_{k+1}^{-1} A_k \\
f_k - Q_k^{-1} &= H_k^T R_k^{-1} H_k + G_{k+1}^T Q_{k+1}^{-1} G_{k+1} - \\
&\quad - G_{k+1}^T Q_{k+1}^{-1} f_{k+1}^{-1} Q_{k+1}^{-1} G_{k+1} \\
&= H_k^T R_k^{-1} H_k + \\
&\quad G_{k+1}^T Q_{k+1}^{-1}\left[Q_{k+1} - f_{k+1}^{-1}\right] Q_{k+1}^{-1} G_{k+1}
\end{aligned}$$

The assumption that $f_{k+1} - Q_{k+1}^{-1}$ is positive semi-definite implies that that $Q_{k+1} - f_{k+1}^{-1}$ is positive semi-definite. It now follows that $f_k - Q_k^{-1}$ is the sum of positive semi-definite matrices and hence is positive semi-definite which completes the induction and hence proves the first result in the Lemma.

Using the last equation above, we have

$$|f_k| \leq \quad |H_k^T R_k^{-1} H_k| + |G_{k+1}^T|^2 |Q_{k+1}^{-1}|^2 |Q_{k+1}| \leq \alpha + \alpha^5$$

Hence the maximum eigenvalue of f_k is less than or equal $\alpha + \alpha^5$. In addition, since $f_k - Q_k^{-1}$ is positive semi-definite, the minimum eigen-value of f_k is greater than or equal the minimum eigen-value of Q_k^{-1}, which is equal to the reciprocal of the maximum eigenvalue of Q_k. Thus the minimum eigenvalue of f_k is greater than or equal $1/\alpha$. Thus the condition number of f_k is bounded by $\alpha^2 + \alpha^6$.

REFERENCES

[1] B. Bell and G. Pillonetto, "A distributed kalman smoother," in *Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems - NecSys'09, Venice, Italy*, 2009.

[2] B. Bell, J. Burke, and G. Pillonetto, "An inequality constrained nonlinear kalman-bucy smoother by interior point likelihood maximization," *Automatica*, vol. 45, pp. 25–33, 2009.

[3] P. Alriksson and A. Rantzer, "Distributed kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto*, 2006.

[4] I. Schizas, G. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Anytime optimal distributed kalman filtering and smoothing," in *Workshop on Statistical Signal Processing, 2007. SSP 07. IEEE/SP 14th*, 2007, p. 368372.

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

[6] A. Giridhar and P. R. Kumar, "Towards a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, pp. 98–107, 2006.

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

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

[9] M. Jordan, *Learning in Graphical Models*. The M.I.T. Press, Cambridge, MA, 1998.

[10] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall, 2005.

[11] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, pp. 1–305, 2008.

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

[13] W. Hastings, "Monte carlo sampling methods using markov chain and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[14] S. Conte and C. deBoor, *Elementary Numerical Analysis*. McGraw-Hill, New York, 1972.

[15] S. Del Favero, "Analysis and development of consensus-based estimation schemes," Ph.D. dissertation, Universit di Padova, Department of Information Engineering, 2010.