43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

A Decomposition Approach to Distributed Analysis of Networked Systems

Cédric Langbort Lin Xiao Raffaello D'Andrea Stephen Boyd

Abstract— **We present a simple distributed algorithm for analyzing well-posedness and stability of a system composed of different sub-units, interconnected over an arbitrary graph. The procedure consists in solving a set of coupled linear matrix inequalities via a subgradient method, with primal decomposition. The proposed algorithm can be implemented in parallel on the system's graph and should prove more efficient than conventional semidefinite programming solvers, for very large systems with a high number of states and interconnection variables.**

I. Introduction

As modern systems are getting more and more complex and modular, there is an increasing need for methods that can assess stability of large-scale interconnections. An even more challenging task is to design controllers for such systems. For example, the US power grid is composed of over 14,000 substations coupled over a dense network [1]. Likewise, it is expected that smart structures used to control turbulence on an airplane's wing will consist of hundred of thousands of distributed sensors and actuators, [6]. In both cases, we thus have to deal with a very large number of state, input and output variables; a situation where current optimal control design methods, based on semidefinite programming (SDP), are known to perform poorly.

A very natural idea, then, is to try to find local conditions which, when satisfied separately by the subsystems, guarantee that the global system performs as desired. This allows to replace the original large problem by a family of smaller, tractable ones that can be solved independently (see e.g. [9], [10], [12]). However, these decentralized conditions are often conservative or do not allow to formulate dynamic controller design as a convex program, when applied to the closed-loop.

In [7], we showed that, for a particular class of systems and interconnection relations, both these issues could be resolved when looking for distributed rather than decentralized conditions. The price to pay is that the obtained linear matrix inequalities (LMIs) are *not* uncoupled. Indeed, as we will review in Section 2, each edge of the system's interconnection graph contributes complicating variables to the LMIs of the adjacent subsystems. However, it is still possible to reduce the time needed to solve these coupled LMIs by using a primal decomposition method. This classical method, which we describe in Section 3, solves an optimization problem by holding the complicating variables fixed and optimizing on the remaining ones. The resulting subproblems can then also be solved independently.

In turn, we obtain an algorithm that not only allows us to tackle stability analysis of very large networked systems numerically, but can also be distributed directly over the system's network, using computing units localized at every node and the preexisting communication channels.

We believe that the present application of primal decomposition to distributed analysis is also of theoretical interest as, to the best of our knowledge, this is the first practical example where the resulting subproblems are semidefinite program themselves.

Notation. Most of the notation is standard. We will use the following shorthand: Given elements e_1,...,e_L in some sets E_1,...,E_L $\underset{k\leq i\leq l}{\mathbf{cat}} e_i$ will designate the element $(e_k, ..., e_l) \in E_k \times ... \times E_l$ when $1 \leq k < l \leq L$. Likewise, given matrices A_1,...,A_L, we let

$$\underset{k\leq i\leq l}{\mathbf{diag}} A_i := \begin{bmatrix} A_k & 0 & \dots & 0 \\ 0 & A_{k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_l \end{bmatrix}.$$

We will denote the $n\times m$ zero matrix by $0^{n\times m}$ and, when A and B are matrices, write $A < B$ (respectively $A \leq B$) to mean that $B - A$ is symmetric, positive-definite (respectively positive semidefinite). Finally, we define the following scalar product (that induces Frobenius norm): $A \bullet B := \mathbf{trace}(A^* B)$.

II. Preliminaries

A. Interconnected systems

In this section, we review the framework and basic results of [7], that will be used throughout the paper.

An interconnected system, as depicted in Figure 1, is composed of L possibly different linear time-invariant subsystems. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of an interconnected system is the complete undirected graph on L vertices $G_1, ..., G_L$. Each vertex $G_i \in \mathcal{V}$ represents a linear time-invariant finite dimensional subsystem. To every edge (G_i, G_j), $i \neq j$, we associate its size, an integer $n_{ij} \geq 0$, with the convention that $n_{ij} = n_{ji}$ for all i, j. By allowing $n_{ij} = 0$,

C. Langbort is with the Department of Theoretical & Applied Mechanics, Cornell University, 212 Kimball Hall, Ithaca NY 14853-1503, email: cl263@cornell.edu

L. Xiao is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, email: lxiao@stanford.edu

Prof. D'Andrea is with the Department of Mechanical & Aerospace Engineering, Cornell University, 101 Rhodes Hall, Ithaca, NY 14850, email: rd28@cornell.edu

Prof. Boyd is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305, email: boyd@stanford.edu

0-7803-8682-5/04/$20.00 ©2004 IEEE

we also capture the case where subsystems G_i and G_j are *not* interconnected. For our purposes, all the relevant information regarding the topology of the interconnection can be summarized by a symmetric L by L matrix $\mathcal{N}$, with entries n_{ij}, the so-called (weighted) adjacency matrix of graph $\mathcal{G}$, [5]. We will denote by $\mathbb{R}^{\mathcal{N}}$ the vector space of partitioned vectors $\mathsf{v} = (\mathsf{v}_1, \cdots, \mathsf{v}_L)$, where each v_i can itself be further partitioned as $\mathsf{v}_i = (\mathsf{v}_{i1}, \cdots, \mathsf{v}_{iL})$, each v_{ij} being of size n_{ij}. When $\mathsf{v} \in \mathbb{R}^{\mathcal{N}}$, each v_i is of size $n_i := \sum_{j=1}^{L} n_{ij}$, for all $i = 1...L$. Each subsystem G_i is

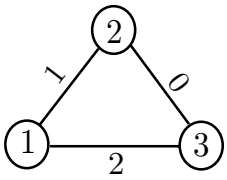

Fig. 1. An example of interconnected system with $L = 3$ subsystems. The integers indicated on the graph are the size n_{ij} of the edges. For example $n_{12} = 1$ and $n_{23} = 0$. i.e. subsystem 2 and 3 cannot communicate.

captured by the following state-space equations:

$$\begin{array}{l} \left[\begin{array}{c} \dot{x_i}(t) \\ w_i(t) \end{array} \right] = \left[\begin{array}{cc} A^i_{\mathrm{TT}} & A^i_{\mathrm{TS}} \\ A^i_{\mathrm{ST}} & A^i_{\mathrm{SS}} \end{array} \right] \left[\begin{array}{c} x_i(t) \\ v_i(t) \end{array} \right] \text{ for all } t \geq 0 \\ \qquad x_i(0) = x_i^0 \end{array} \tag{1}$$

where $x_i(t) \in \mathbb{R}^{m_i}$ and $v_i(t), w_i(t) \in \mathbb{R}^{n_i}$ for all $t \geq 0$. The latter signals are the overall interconnection signals used by subsystem G_i. For each given i, we partition these signals further into v_{ij} and w_{ij}— the interconnection signals G_i shares with G_j. By adding zero-components to the signals of smaller size and modifying the state-space matrices of Equation 1 if this is not the case, we can always assume that $v_{ij}(t),\ v_{ji}(t),\ w_{ij}(t)$ and $w_{ji}(t)$ all belong to $\mathbb{R}^{n_{ij}}$ for all $t \geq 0$. This, in turn, naturally defines two signals v and w such that $v(t), w(t) \in \mathbb{R}^{\mathcal{N}}$ for all $t \geq 0$. Note that v_{ij} and v designate a $\mathbb{R}^{n_{ij}}$ and $\mathbb{R}^{\mathcal{N}}$-valued signal, respectively while v_{ij} is a component of the vector $\mathsf{v} \in \mathbb{R}^{\mathcal{N}}$. This construction allows us to think of A^i_{SS} as a square, n_i by n_i, matrix without loss of generality.

Finally, we can get a description of the whole system by closing all loops and relating w_{ij} to v_{ji}, for all i, j. The simplest way this can be done is by imposing the so-called ideal interconnection relation

$$(w(t), v(t)) \in \mathcal{S}_{\mathcal{I}} \quad \text{for all } t \geq 0, \tag{2}$$

where the interconnection subspace $\mathcal{S}_{\mathcal{I}}$ is defined as

$$\mathcal{S}_{\mathcal{I}} := \left\{ (\mathsf{w}, \mathsf{v}) \in \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} : \mathsf{w}_{\mathsf{ji}} = \mathsf{v}_{\mathsf{ij}},\ \forall i, j = 1...L \right\}. \tag{3}$$

We refer to [7] for the study of systems with more general interconnection relations, defined by integral quadratic constraints.

B. Sufficient conditions for well-posedness and stability

We say that an interconnected system, defined by adjacency matrix $\mathcal{N} \in \mathbb{R}^{L \times L}$ and state-space matrices as per Equation 1 is well-posed if the two subspaces $\mathcal{S}_{\mathcal{I}}$ and $\mathcal{S}_{\mathcal{B}}$ satisfy

$$\mathcal{S}_{\mathcal{I}} \cap \mathcal{S}_{\mathcal{B}} = \{0\},$$

where

$$\begin{array}{l} \mathcal{S}_{\mathcal{B}} := \left\{ (\mathsf{w}, \mathsf{v}) \in \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} : \left[\begin{array}{c} \mathsf{w_i} \\ \mathsf{v_i} \end{array} \right] \in \mathcal{S}^i_{\mathcal{B}}\ \forall i = 1...L \right\} \\ \mathcal{S}^i_{\mathcal{B}} := Im \left[\begin{array}{c} A^i_{\mathrm{SS}} \\ I \end{array} \right]. \end{array} \tag{4}$$

Well-posedness guarantees that all signals circulating in the loops of the interconnected system are uniquely determined and belong to $\mathbf{L}_2$ for any initial conditions x_i^0. We will say that a well-posed interconnected system is stable if for any set $\{x_i^0,\ i = 1...L\}$ of initial conditions, x_i is a smooth function of time that approaches zero as time goes to infinity, for all i. The main analysis result of [7] is the following

Theorem 1: An interconnected system is well-posed and stable if there exist symmetric matrices $X^i_{\mathrm{T}} \in \mathbb{R}^{m_i \times m_i}$ and $X^{11}_{ij} \in \mathbb{R}^{n_{ij} \times n_{ij}}$ for all $i, j = 1...L$, and $X^{12}_{ij} \in \mathbb{R}^{n_{ij} \times n_{ij}}$ for all $i \geq j$, with X^{12}_{ii} skew-symmetric, such that $X^i_{\mathrm{T}} > 0$ and

$$\begin{array}{l} \left[\begin{array}{cc} I & 0 \\ \hline A^i_{\mathrm{TT}} & A^i_{\mathrm{TS}} \\ A^i_{\mathrm{ST}} & A^i_{\mathrm{SS}} \\ \hline 0 & I \end{array} \right]^* \left[\begin{array}{cc|cc} 0 & X^i_{\mathrm{T}} & 0 & 0 \\ X^i_{\mathrm{T}} & 0 & 0 & 0 \\ \hline 0 & 0 & Z^{11}_i & Z^{12}_i \\ 0 & 0 & (Z^{12}_i)^* & Z^{22}_i \end{array} \right] \times \\ \qquad \left[\begin{array}{cc} I & 0 \\ \hline A^i_{\mathrm{TT}} & A^i_{\mathrm{TS}} \\ A^i_{\mathrm{ST}} & A^i_{\mathrm{SS}} \\ \hline 0 & I \end{array} \right] < 0 \end{array} \tag{5}$$

for all $i = 1...L$, where

$$\begin{array}{l} Z^{11}_i := -\underset{1 \leq j \leq L}{\mathbf{diag}}\, X^{11}_{ij}, \\ Z^{22}_i := \underset{1 \leq j \leq L}{\mathbf{diag}}\, X^{11}_{ji}, \\ Z^{12}_i := \mathbf{diag}(-\underset{1 \leq j \leq i}{\mathbf{diag}} X^{12}_{ij}, \underset{i < j \leq L}{\mathbf{diag}} (X^{12}_{ji})^*). \end{array}$$

Theorem 1 follows from simple dissipativity arguments, each LMI meaning that the corresponding subsystem is dissipative with respect to supply rate

$$\sum_{j=1}^{L} \left[\begin{array}{c} \mathsf{w_{ij}} \\ \mathsf{v_{ij}} \end{array} \right]^* \left[\begin{array}{cc} X^{11}_{ij} & X^{12}_{ij} \\ X^{12*}_{ij} & -X^{11}_{ji} \end{array} \right] \left[\begin{array}{c} \mathsf{w_{ij}} \\ \mathsf{v_{ij}} \end{array} \right],$$

with storage function $x_i^* X^i_{\mathrm{T}} x_i$. Once again, we refer to [7] for more details on this and related results. Our goal, in the remainder of the paper, is to introduce an efficient distributed algorithm for checking feasibility of LMI (5), using the particular structure of these inequalities. For easy later reference, we rewrite each one as

$$\mathcal{L}_i(X^i_{\mathrm{T}}, \mathcal{X}_i) < 0 \tag{6}$$

where, for all $i = 1...L$, $\mathcal{L}_i$ is an affine mapping and variable $\mathcal{X}_i$ is defined as

$$\mathcal{X}_i := \left(\underset{1\le j\le L}{\mathbf{cat}} X_{ij}^{11}, \underset{1\le j\le L}{\mathbf{cat}} X_{ji}^{11}, \underset{1\le j\le i}{\mathbf{cat}} X_{ij}^{12}, \underset{i<j\le i}{\mathbf{cat}} X_{ji}^{12} \right).$$

We also let $\mathcal{X} := \underset{1\le i\le L}{\mathbf{cat}} \mathcal{X}_i$.

III. A Distributed Algorithm

In this section, we explain how to use a simple decomposition method to solve analysis conditions of Theorem 1 in a distributed fashion. We emphasize that the same technique is equally applicable to control design and that we only restricted ourselves to analysis for the sake of simplicity. For similar reasons, we purposely focused on a conceptually simple approach. We are more interested in pointing out the connection between distributed optimization algorithms and distributed control than in getting into the specifics and fine-tuning of a particular algorithm.

A. Subgradient methods and primal decomposition

We start by recalling the main features of the method of primal decomposition. In order to keep with the spirit of the rest of the paper, we do so in the context of optimization over graphs.

Let a graph $\mathcal{G}$ be given and assume that each vertex i has a set of "private variables" x_i that no other node can access, while another set of variables, y, is accessible to all vertices. We want to solve the following convex optimization problem

$$\begin{aligned} &min \sum_i f_i(x_i, y) \\ &\text{subject to } x_i \in \mathcal{C}_i \quad \text{for all } i \end{aligned} \tag{7}$$

where, for each i, f_i is a convex function and $\mathcal{C}_i$ is a convex set. More precisely, we want to design an optimization algorithm that respects the partition between variables, i.e. that any operation requiring knowledge of x_i should be performed by vertex i. Such an algorithm is easily parallelized, since all these operations can be executed *at the same time*, provided each vertex has computational power.

The first step of the primal decomposition method hinges on the fact that, since (7) is a convex problem, it can be solved by optimizing with respect to each variable separately. In other words, if we define

$$\phi_i(y) := \underset{x_i\in\mathcal{C}_i}{min} f_i(x_i, y) \text{ and } \phi(y) := \sum_{i=1}^{L} \phi_i(y)$$

for all y, then $(\bar{x}_1, ..., \bar{x}_L, \bar{y})$ is an optimal point for problem (7) if and only if

$$\bar{y} = argmin\ \phi(y) \tag{8a}$$

$$\bar{x}_i = \underset{x_i\in\mathcal{C}_i}{argmin}\ f_i(x_i, \bar{y}), \tag{8b}$$

and the optimal value is given by $\phi(\bar{y})$. Thus, solving problem (7) is equivalent to minimizing function ϕ. This new optimization problem is often referred to as the *master problem*.

Solving (8a) requires some care, as function ϕ is typically not differentiable and classical algorithms from smooth optimization theory, such as Newton's method, cannot be readily applied. However, each ϕ_i is convex because we assumed f_i to be *jointly* convex in x_i and y. As a result, ϕ is also a convex function of y and has a non-empty (convex, compact) subdifferential $\partial\phi(y)$ at each point in the interior of its domain, [2], [4]. It is thus possible to replace gradients by subgradients and use them as search directions to determine the minimum of ϕ. More precisely, we have the following result from [3]:

Theorem 2 (Subgradient Method): Let $\{\alpha_k\}_{k\in\mathbb{N}}$ be a *nonsummable diminishing* sequence of nonnegative numbers i.e.

$$\lim_{k\to\infty} \alpha_k = 0\ ;\ \sum_{k=1}^{\infty} \alpha_k = \infty\ ,$$

and let the sequence $\{y^{(k)}\}$ be defined by

$$y^{(k)} = y^{(k-1)} - \alpha_{k-1} g(y^{(k-1)}) \tag{9}$$

where, for each k, $g(y^{(k)})$ is a subgradient of ϕ at $y^{(k)}$. Assume further that there exists $G > 0$ such that $|g(y^k)| \le G$ for all $k \ge 0$. Then, the sequence $\{\phi_{best}^{(k)}\}$ defined by

$$\phi_{best}^{(k)} := min \left\{ \phi_{best}^{(k-1)}, \phi(y^{(k)}) \right\}$$

converges to the optimal value of master problem (8a).

Note that while there is some similarity between Newton's and the subgradient method, the latter is *not* a descent method. This is why one has to keep track of the best point found so far when defining $\phi_{best}^{(k)}$.

The great advantage of solving optimization problem (7) by resorting to a master problem and using subgradient iterations is that these subgradients can be calculated separately by each node, at each step of the algorithm, and that the whole procedure can thus be parallelized. Indeed, using the rules of subgradient calculus (see, e.g. [4]), we can write

$$\partial\phi(y) = \sum_{i=1}^{L} \partial\phi_i(y) \text{ for all } y,$$

which means that a subgradient of ϕ at point y can be computed by letting node i find a subgradient $g_i(y)$ of ϕ_i, for each i. Combining this observation with Theorem 2 yields the following distributed algorithm for solving problem (7):

Algorithm 1 Primal Decomposition

Initialize y to $y^{(0)}$
for $k = 1$ to # of steps **do**
1. (distributed calculation) Each vertex computes $g_i(y^{(k)})$ and broadcasts the value
2. Update $y^{(k+1)}$ according to (9) and broadcast the value
end for.

At this stage, it is still unclear how broadcasting should be performed. It might also appear as though step 2 has to be executed by a central unit accessing all subgradients or through flooding. We will soon see that this basic algorithm can in fact be completely distributed when applied to stability analysis of networked systems.

B. Application to distributed analysis

In order to use primal decomposition for networked system analysis, we should first reformulate the LMI conditions of Theorem 1 as an optimization problem of the form (7). This is the content of the following

Proposition 1: LMI (5) are strictly feasible if and only if there exists $\epsilon > 0$ such that the optimal value φ of convex program

$$\begin{aligned} &min \sum_{i=1}^{L} t_i \\ \text{subject to } &\mathcal{L}_i(X_{\mathrm{T}}^i, \mathcal{X}_i) \leq t_i I \\ &X_{\mathrm{T}}^i > 0 \ ; \ t_i \geq -\epsilon \end{aligned} \tag{10}$$

is strictly negative and, at optimum, $t_i < 0$ for all i.

Proof: The necessity part is clear. For sufficiency, assume LMI (5) are strictly feasible. Then there exists $\bar{\mathcal{X}}$, $\epsilon_0 > 0$ and $\bar{X}_{\mathrm{T}}^i$ for all i such that $\mathcal{L}_i(\bar{X}_{\mathrm{T}}^i, \bar{\mathcal{X}}_i) \leq -\epsilon_0 I$ for all i. Hence, if $0 < \epsilon < \epsilon_0$, the point $X_{\mathrm{T}}^i = \bar{X}_{\mathrm{T}}^i,\ \mathcal{X}_i = \bar{\mathcal{X}}_i,\ t_i = -\epsilon\ (i = 1...L)$ is feasible for problem (10). Hence

$$-L\epsilon \leq \varphi \leq -L\epsilon,$$

which implies $\varphi = -L\epsilon < 0$ and $t_i < 0$ for all i at optimum. ∎

According to Proposition 1, we can thus deduce feasibility of LMI (5) by solving convex program (10) for a fixed, sufficiently small, value of ϵ and checking the corresponding optimal value. We can perform this minimization via the decomposition method outlined above, provided we know how to compute a subgradient of function $\phi_i(\mathcal{X}_i)$, defined as the optimal value of convex program $P(\mathcal{X}_i)$:

$$min\ t_i \tag{11a}$$

$$\text{subject to } \mathcal{L}_i(X_{\mathrm{T}}^i, \mathcal{X}_i) \leq t_i I \tag{11b}$$

$$X_{\mathrm{T}}^i > 0 \ ; \ t_i \geq -\epsilon. \tag{11c}$$

A subgradient of ϕ_i at $\mathcal{X}_i$ will be denoted $g(\mathcal{X}_i)$. If we decompose $\mathcal{X}_i$ into its components, we can define *partial subgradients* $g_{X_{ij}^{11}}(\mathcal{X}_i)$ for all $j = 1...L$, by considering ϕ_i as a function of X_{ij}^{11}, all other components being held fixed. Of course, this definition generalizes to all other components of $\mathcal{X}_i$.

With the notation of the preceding section, we see that, for our problem, the private variable x_i of each vertex i is the pair (t_i, X_{T}^i), while $\mathcal{X}$ corresponds to the shared variable y. Note that, in contrast with the situation of section III-A, vertex i does not need to access all the components of $\mathcal{X}$ but only $\mathcal{X}_i$. More precisely, neighboring vertices i and j $(i \geq j)$ share variables X_{ij}^{11}, X_{ji}^{11} and X_{ij}^{12} but non-communicating vertices do not share any variables. As a result, $\mathcal{X}$ never needs to be updated as a whole when taking a step in the subgradient direction. Each vertex can update the relevant components independently according to

$$\left(X_{ij}^{11}\right)^{(k+1)} = \left(X_{ij}^{11}\right)^{(k)} - \alpha_k \left(g_{X_{ij}^{11}}(\mathcal{X}_i^{(k)}) + g_{X_{ij}^{11}}(\mathcal{X}_j^{(k)})\right) \tag{12a}$$

$$\left(X_{ji}^{11}\right)^{(k+1)} = \left(X_{ji}^{11}\right)^{(k)} - \alpha_k \left(g_{X_{ji}^{11}}(\mathcal{X}_i^{(k)}) + g_{X_{ji}^{11}}(\mathcal{X}_j^{(k)})\right) \tag{12b}$$

$$\left(X_{ij}^{12}\right)^{(k+1)} = \left(X_{ij}^{12}\right)^{(k)} - \alpha_k \left(g_{X_{ij}^{12}}(\mathcal{X}_i^{(k)}) + g_{X_{ij}^{12}}(\mathcal{X}_j^{(k)})\right), \tag{12c}$$

when it receives the current value of the subgradients from its neighbors. Hence, we see that when using Algorithm 1 for networked systems analysis, we can not only run the calculations of step 1 in parallel, each subsystem computing its own subgradient, but also restrict broadcasting of step 1 and 2 to direct neighbor-to-neighbor communication. However, some kind of global communication (or flooding) is still needed to determine when the algorithm terminates, because every node has to know whether $\phi_i(y^{(k)}) < 0$ *for all* i.

We now explain how to calculate the subgradients.

Proposition 2: Let $1 \leq i \leq L$. Let S_i be the dual variable associated with constraint (11b) in minimization problem $P(\mathcal{X}_i)$ and $\bar{S}_i$ its value at optimum. Then, the components of a subgradient of ϕ_i are given by

$$g_{X_{ij}^{11}}(\mathcal{X}_i) = -E_j^* M_i \bar{S}_i M_i^* E_j, \tag{13a}$$

$$g_{X_{ji}^{11}}(\mathcal{X}_i) = F_j^* M_i \bar{S}_i M_i^* F_j, \tag{13b}$$

$$g_{X_{ij}^{12}}(\mathcal{X}_i) = -E_j^* M_i \bar{S}_i M_i^* F_j - F_j^* M_i \bar{S}_i M_i^* E_j \text{ if } j \leq i, \tag{13c}$$

$$g_{X_{ji}^{12}}(\mathcal{X}_i) = E_j^* M_i \bar{S}_i M_i^* F_j + F_j^* M_i \bar{S}_i M_i^* E_j \text{ if } j > i, \tag{13d}$$

where

$$M_i = \begin{bmatrix} A_{\mathrm{ST}}^i & A_{\mathrm{SS}}^i \\ 0 & I \end{bmatrix}\ E_j = \begin{bmatrix} T_j \\ 0^{n_i \times n_{ij}} \end{bmatrix},$$

and

$$F_j = \begin{bmatrix} 0^{n_i \times n_{ij}} \\ T_j \end{bmatrix},\ T_j = \begin{bmatrix} 0^{n_{i1} \times n_{ij}} \\ \vdots \\ 0^{n_{i(j-1)} \times n_{ij}} \\ I_{n_{ij}} \\ 0^{n_{i(j+1)} \times n_{ij}} \\ \vdots \\ 0^{n_{iL} \times n_{ij}} \end{bmatrix}.$$

It is thus very easy to obtain a subgradient of ϕ_i if we use a primal-dual algorithm, such as SeDuMi [11], to solve semidefinite program (11a).

Proof: The proof is an adaptation of results of [2], that we give for completeness. Let $\mathcal{X}_i$ be fixed. Consider

the Lagrangian function of problem $P(\mathcal{X}_i)$,

$$\begin{aligned} L(X_{\mathrm{T}}^i, t_i, S_i, Y_i, \lambda_i) :=& t_i + S_i \bullet \left(\mathcal{L}_i(X_{\mathrm{T}}^i, \mathcal{X}_i) - t_i I\right) \\ & - Y_i \bullet X_{\mathrm{T}}^i - \lambda_i(t_i + \epsilon) \end{aligned} \tag{14}$$

and its Lagrange dual function, defined for $S_i > 0$, $Y_i > 0$ and $\lambda_i \geq 0$ by

$$K(S_i, Y_i, \lambda_i) := \inf_{X_{\mathrm{T}}^i, t_i} L(X_{\mathrm{T}}^i, t_i, S_i, Y_i, \lambda_i).$$

Since $P(\mathcal{X}_i)$ is clearly strictly feasible (just choose t_i large enough), strong duality holds for this problem and its dual, which means that both the primal and dual optimal values are attained and that they are equal. In other words, there exists $\bar{S}_i > 0$, $\bar{Y}_i > 0$ and $\bar{\lambda}_i \geq 0$ such that

$$\phi_i(\mathcal{X}_i) = K(\bar{S}_i, \bar{Y}_i, \bar{\lambda}_i).$$

Using the definition of K, we then get that

$$\phi_i(\mathcal{X}_i) \leq t_i + \bar{S}_i \bullet \left(\mathcal{L}_i(X_{\mathrm{T}}^i, \mathcal{X}_i) - t_i I\right) - \bar{Y}_i \bullet X_{\mathrm{T}}^i - \bar{\lambda}_i(t_i + \epsilon)$$

for all t_i and X_{T}^i. In particular, if (t_i, X_{T}^i) is feasible for $P(\mathcal{X}_i + \Delta\mathcal{X}_i)$, then

$$\begin{aligned} \phi_i(\mathcal{X}_i) &\leq t_i - \bar{S}_i \bullet \mathcal{L}_i(0, \Delta\mathcal{X}_i) - \bar{Y}_i \bullet X_{\mathrm{T}}^i - \bar{\lambda}_i(t_i + \epsilon) \\ &\leq t_i - \bar{S}_i \bullet \mathcal{L}_i(0, \Delta\mathcal{X}_i), \end{aligned} \tag{15}$$

where, in the first inequality, we have used that $\mathcal{L}_i$ is affine and, in the second, that $t_i + \epsilon \geq 0$, $X_{\mathrm{T}}^i > 0$. Taking the infimum over such feasible pairs, we deduce

$$\phi_i(\mathcal{X}_i) + \bar{S}_i \bullet \mathcal{L}_i(0, \Delta\mathcal{X}_i) \leq \phi_i(\mathcal{X}_i + \Delta\mathcal{X}_i).$$

The result then follows from expressing $\mathcal{L}_i(0, \Delta\mathcal{X}_i)$ in terms of the components of $\mathcal{X}_i$. ■

IV. Example

In this section, we give preliminary results comparing the performance of the proposed distributed algorithm to that of a centralized scheme, which does not exploit the structure of LMIs (5) and solves them directly with SeDuMi. Our test-system is depicted in Figure 2. Each subsystem has $m_i = 10$ states and the dimension n_{ij} of each interconnection signal is as indicated on the system's graph. Subsystems have been chosen at random, with the constraint that they all be stable and contractive, i.e.

$$\|w_i\| < \|v_i\| \text{ for all } i = 1...L,$$

where $\|.\|$ denotes the $\mathbf{L}_2$ norm of a signal. While it is shown in [7] that LMIs (5) can be used to prove stability of an interconnection of unstable or non-contractive subsystems, these two conditions ensure that the LMIs are always strictly feasible (just take $X_{ij}^{11} = I_{n_{ij}}$, $X_{ij}^{12} = 0$ for all i, j and, for X_{T}^i, any solution to the Kalman-Yakubovich-Popov LMI condition ensuring that subsystem i has small gain). According to Proposition 1, we thus know that both algorithms converge in theory and we can evaluate performance by focusing solely on speed of convergence.

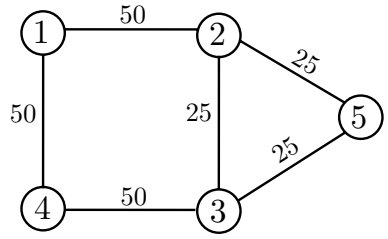

Fig. 2. 'Large' interconnected system with $L = 5$ subsystems. The integers indicated on the graph are the size n_{ij} of the edges.

When applied to the interconnected system of Figure 2, the centralized method simply fails. The program runs out of memory before it can complete a single iteration of the interior-point algorithm. This is due to the very large dimension of the interconnection signals and corresponding variables X_{ij}^{11} and X_{ij}^{12}. On the other hand, the size of these scales should not affect the running time of the primal decomposition method, since they are held constant during every iteration. We thus expect our algorithm to perform well on this example and, more generally, for systems with dense networks and high communication rates.

This is indeed the case, as can be seen on Figure 3. Starting with a random value of $\mathcal{X}^{(0)}$, the subgradient algorithm converges and stops at a negative value of ϕ_{best}. Note that we stopped the iterations as soon as $\phi_{best}^{(k)} < 0$ and $\phi_i(y^{(k)}) < 0$ for all i, which is all we need to establish feasibility of LMI (5), and did not wait to attain the minimum of problem (10). The whole process took approximately 30 minutes on a 1.6 GHz workstation, when calculating all subgradients on the same machine, one after the other, at each step. Provided transmission delays and idle time are negligible, we can thus expect that a fully distributed implementation of the algorithm, where the subgradients are computed in parallel, can certify well-posedness and stability of the system in about 6 minutes.

V. Concluding Remarks

In this paper, we have set the stage for the use of distributed optimization methods for analysis of large-scale networked systems. In particular, we have developed a simple distributed algorithm based on primal decomposition and subgradient methods.

There are many ways in which this basic algorithm can be improved. First, while it can handle systems with a large number of interconnection signals, which are completely out of reach for conventional methods, our algorithm converges rather slowly. The main reason for this is that one has to solve L semidefinite programs at each step, which requires an important computational effort. As explained before, parallelizing calculations should help reduce wall-clock time but only provided synchronization issues and communication delays between subsystems are properly addressed. An encouraging result in this direction can be

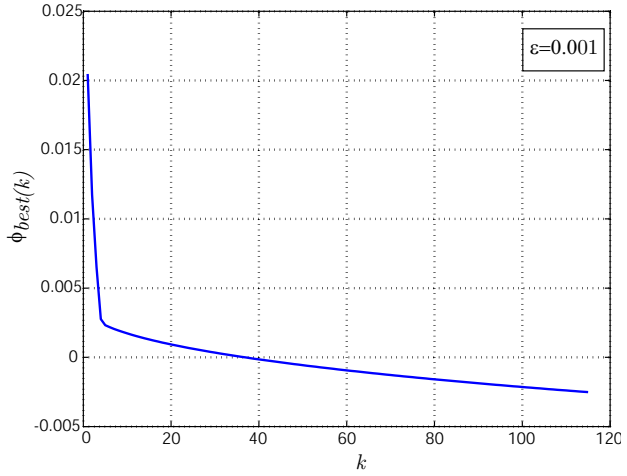

Fig. 3. Best value for problem (10) versus iteration number k, in the subgradient method with nonsummable diminishing step $\alpha_k = \frac{1}{\sqrt{k}}$. The algorithm stopped before reaching optimum, for the smallest value of k such that $\phi_i(y^{(k)}) < 0$ for all i. The corresponding values are: $\phi_1(y^{(k)}) = -0.00027853$, $\phi_2(y^{(k)}) = -5.7078 \times 10^{-6}$, $\phi_3(y^{(k)}) = -0.00023152$ and $\phi_4(y^{(k)}) = \phi_5(y^{(k)}) = -0.001$.

found in [8], where it is shown that updates (9) can be performed asynchronously, as new subgradients become available, as long as *all* partial subgradients are used with the same long-time frequency. We plan to investigate the role of synchronization in more details in future work.
Another drawback of our method is that it is difficult to discriminate between slow convergence of the algorithm and infeasibility of the original LMI (5). This is because the theoretical upper-bounds on the number of iterations needed to approximate the optimal value of problem (10) with a prescribed tolerance are hard to use in practice. We are currently exploring *primal-dual* subgradient methods, which should provide a better convergence criterion than the present method.

ACKNOWLEDGMENTS

This work was funded in part by a NSF grant for "Design of Robust, Networked Control Systems via Convex Optimization" and AFOSR grant F49620-01-1-0119. C.L. would like to thank Ramu Chandra for many helpful discussions.

REFERENCES

[1] R. Albert, I. Albert and G. L. Nakarado. Structural Vulnerability of the North American Power Grid. To appear in *Physical Review E*, available at `http://xxx.lanl.gov/abs/cond-mat/0401084(2004)`.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2003.
[3] S. Boyd, L.Xiao and A. Mutapcic, Lecture Notes for EE392o, Stanford University, Autumn 2003.
[4] F. H. Clarke, *Nonsmooth Analysis and Optimization*, Wiley-Interscience, 1983.
[5] R. Diestel. *Graph Theory*. Second edition, Graduate Texts in Mathematics, Springer Verlag.
[6] J. Kim, Control of turbulent boundary layers. *Physics of Fluids*, vol. 15 no. 5, May 2003.
[7] C. Langbort and R. D'Andrea, Distributed Control of Heterogeneous Systems Interconnected over an Arbitrary Graph. In *Proc. IEEE Conference on Decision and Control*, 2003.
[8] A. Nedic, D. Bertsekas and V. Borkar. Distributed Asynchronous Incremental Subgradient Methods. In *Proceedings of the March 2000 Haifa Workshop on "Inherently Parallel Algorithms in Feasibility and Optimization and their Applications", D. Butnariu, Y. Censor, and S. Reich (Eds.)*, Elsevier, Amsterdam, 2001.
[9] M. Rotkowitz and S. Lall. Decentralized control information structures preserved under feedback. In *Proc. IEEE Conference on Decision and Control*, 2002.
[10] G. Scorletti and G. Duc. An LMI approach to decentralized control. *International Journal of Control*, vol. 74, no. 3, pp. 211-224, 2001.
[11] J. F. Sturm. Using SeDuMi, a Matlab Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Sofwares*, vol. 11-12, pp.625-653, 1999.
[12] P. Voulgaris. A convex characterization of classes of problems in control with specifi c interaction and communication structures. *UIUC Technical Report, AAE 01 06 UILU ENG 01-05-06*, 2001.