

Scalable analysis of linear networked systems via chordal decomposition

Yang Zheng[†], Maryam Kamgarpour[‡], Aivar Sootla[†] and Antonis Papachristodoulou[†]

Abstract—This paper introduces a chordal decomposition approach for scalable analysis of linear networked systems, including stability, \mathcal{H}_2 and \mathcal{H}_∞ performance. Our main strategy is to exploit any sparsity within these analysis problems and use chordal decomposition. We first show that Grone’s and Agler’s theorems can be generalized to block matrices with any partition. This facilitates networked systems analysis, allowing one to solely focus on the physical connections of networked systems to exploit scalability. Then, by choosing Lyapunov functions with appropriate sparsity patterns, we decompose large positive semidefinite constraints in all of the analysis problems into multiple smaller ones depending on the maximal cliques of the system graph. This makes the solutions more computationally efficient via a recent first-order algorithm. Numerical experiments demonstrate the efficiency and scalability of the proposed method.

I. INTRODUCTION

Large-scale networked systems consisting of multiple subsystems over a network have received considerable attention [1]. This class of systems arises in many practical applications, such as the smart grid [2] and automated traffic systems [3]. One of the challenges arising in these systems is to develop scalable methods that are able to solve the associated analysis and synthesis problems efficiently. However, classical methods often suffer from lack of scalability to large-scale systems, since computational demand usually grows rapidly as the system’s dimension increases.

In the literature, there are two groups of scalable analysis techniques for large-scale networked systems: 1) *compositional analysis* [4]–[6]; and 2) *positive systems theory* [7]–[11]. The former method is usually carried out in the framework of dissipative systems, while the latter method aims to solve a special type of dynamical systems. The main strategy of compositional analysis is to find individual supply rates for each dissipative subsystem and then to establish a global storage function as a combination of the local storage functions [4]. Recently, Meissen *et al.* employed a first-order method to optimize the local supply rates for certifying stability of an interconnected system [5], which might reduce the conservatism brought by individual storage functions. Anderson and Papachristodoulou proposed a decomposition technique based on graph partitioning that facilitates the compositional analysis [6]. Another group of

scalable strategies focuses on a particular class of systems, *i.e.*, positive systems [7], where the system matrices only have nonnegative off-diagonal entries. It is well-known that stability and performance of positive systems can be verified using linear Lyapunov functions [8], which can be computed by more scalable linear programs (LPs) instead of traditional semidefinite programs (SDPs). Tanaka and Langbort showed that it is necessary and sufficient to use a diagonal Lyapunov function in the KYP lemma for positive systems [9]. Sootla and Rantzer proposed scalable model reduction techniques for positive systems using linear energy functions [10].

In contrast to the compositional analysis and positive system theory, our approach focuses on the inherent structure and sparsity of networked systems and uses sparse optimization techniques, particularly chordal decomposition, to solve the analysis problems efficiently. This idea is in line with some of the early results in the field [12]–[15]. Chordal decomposition is a celebrated result in linear algebra that connects sparse positive semidefinite matrices and chordal graphs. There is a broad literature regarding the applications of chordal graph properties in combinatorial problems, Cholesky factorization, matrix completion and sparse semidefinite optimization; see [16] for a comprehensive review.

In this paper, we introduce a chordal decomposition approach for scalable analysis of linear networked systems. We focus on the well-known convex formulations of the analysis problems, *i.e.*, stability, \mathcal{H}_2 and \mathcal{H}_∞ performance, and show how to decompose large positive semidefinite constraints in all of the analysis problems into multiple smaller ones, thus facilitating their solutions. Specifically, compared to [12]–[15], the contributions of this paper are: 1) we generalize Grone’s and Agler’s theorems to block matrices with arbitrary partition, and utilize the generalization for networked systems analysis; 2) we extend the scope of stability verification in [13] to \mathcal{H}_2 and \mathcal{H}_∞ analyses of networked systems. Our approach can potentially be applied to other analysis and synthesis problems, such as structured model reduction and stabilizing feedback design.

The rest of this paper is organized as follows. In Section II, we present the problem formulation. Chordal decomposition in sparse SDPs is introduced in Section III. Section IV presents the scalable analysis approach for stability, \mathcal{H}_2 and \mathcal{H}_∞ performance. Numerical results are shown in Section V, and we conclude the paper in Section VI.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. Preliminaries on graph-theoretic notions

A directed graph is denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and it consists of a set of nodes $\mathcal{V} = \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

The work of Y. Zheng is supported in part by the Clarendon Scholarship, and in part by the Jason Hu Scholarship. The work of M. Kamgarpour is supported by ERC Starting Grant CONENE.

[†]Y. Zheng, A. Sootla and A. Papachristodoulou are with Department of Engineering Science at the University of Oxford. (E-mail: {yang.zheng, aivar.sootla, antonis}@eng.ox.ac.uk)

[‡]M. Kamgarpour is with Department of Electrical Engineering and Information Technology at ETH Zurich. (E-mail: mkamgar@control.ee.ethz.ch)

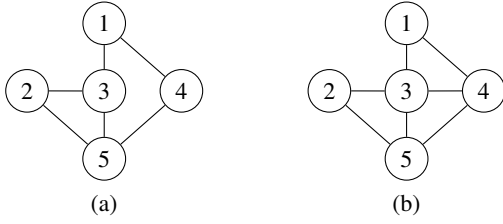


Fig. 1: (a) Nonchordal graph: the cycle (1-3-5-4) is of length four but with no chords. (b) Chordal graph by adding an undirected edge (3, 4): all cycles of length greater than three have a chord.

Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is called undirected if $(u, v) \in \mathcal{E} \Leftrightarrow (v, u) \in \mathcal{E}$. A graph is *complete* if there exists an edge between any pair of nodes. A clique is a subset of nodes $\mathcal{C} \subseteq \mathcal{V}$ that induces a complete subgraph. If \mathcal{C} is not contained by any other clique, then it is referred to as a *maximal clique*. A cycle of length k is a sequence $\{v_1, v_2, \dots, v_k\} \subseteq \mathcal{V}$ with $(v_k, v_1) \in \mathcal{E}$ and $(v_i, v_{i+1}) \in \mathcal{E}, \forall i = 1, \dots, k-1$. A *chord* in a cycle $\{v_1, v_2, \dots, v_k\}$ is an edge (v_i, v_j) that connects two nonconsecutive nodes in the cycle.

An undirected graph is called *chordal* if every cycle of length greater than three has a chord. Examples of chordal graphs include complete graphs and acyclic undirected graphs. Given a chordal graph, there are very efficient algorithms to find maximal cliques [17]. Non-chordal graphs $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can always be extended to a chordal graph $\hat{\mathcal{G}}(\mathcal{V}, \hat{\mathcal{E}})$ by adding edges to \mathcal{E} , i.e., $\mathcal{E} \subset \hat{\mathcal{E}}$, and this process is called *chordal extension*. There are several efficient heuristics to generate a good extension [16]. For example, the graph in Fig. 1 (a) is non-chordal, and it can be chordally extended to that in Fig. 1 (b) by adding an undirected edge (3, 4). The graph in Fig. 1 (b) has three maximal cliques: $\mathcal{C}_1 = \{1, 3, 4\}$, $\mathcal{C}_2 = \{2, 3, 5\}$ and $\mathcal{C}_3 = \{3, 4, 5\}$.

B. Sparse block matrices

A matrix $M \in \mathbb{R}^{N \times N}$ has $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ -partitioning with $N = \sum_{k=1}^n \alpha_k$ if

$$M = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1n} \\ M_{12} & M_{22} & \dots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \dots & M_{nn} \end{bmatrix}$$

where $M_{ij} \in \mathbb{R}^{\alpha_i \times \alpha_j}, i, j = 1, \dots, n$. We describe the sparsity pattern of α -partitioned M by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$:

$$\mathbb{R}_{\alpha}^{N \times N}(\mathcal{E}, 0) = \{M \in \mathbb{R}^{N \times N} | M_{ij} = 0 \text{ if } (j, i) \notin \mathcal{E}^*\}, \quad (1)$$

where $\mathcal{E}^* = \mathcal{E} \cup \{(i, i), \forall i \in \mathcal{V}\}$. If \mathcal{G} is undirected, we define the space of sparse symmetric block matrices as

$$\mathbb{S}_{\alpha}^N(\mathcal{E}, 0) = \{M \in \mathbb{S}^N | M_{ij} = M_{ji}^T = 0 \text{ if } (j, i) \notin \mathcal{E}^*\}, \quad (2)$$

and the cone of sparse block positive semidefinite (PSD) matrices as $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, 0) = \{M \in \mathbb{S}_{\alpha}^N(\mathcal{E}, 0) | M \succeq 0\}$. Also, we define a cone $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, ?)$ as the set of matrices in $\mathbb{S}_{\alpha}^N(\mathcal{E}, 0)$ that have a positive semidefinite completion, i.e., $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, ?) = \mathbb{P}_{\mathbb{S}_{\alpha}^N(\mathcal{E}, 0)}(\mathbb{S}_{\alpha,+}^N)$, where \mathbb{P} denotes the projection onto the space of sparse matrices.

Remark 1: The definitions (1) and (2) also allow the block entry $M_{ij} = 0$ if $(j, i) \in \mathcal{E}^*$. Then we have $M \in \mathbb{S}_{\alpha}^N(\hat{\mathcal{E}}, 0)$ if $M \in \mathbb{S}_{\alpha}^N(\mathcal{E}, 0)$ and $\hat{\mathcal{E}}$ is a chordal extension of \mathcal{E} . This fact will be used in Section III-B. The notation above is a natural extension of sparse scalar matrices to sparse block matrices with α partition. If $\alpha = \{1, 1, \dots, 1\}$, then the notations are reduced to the normal cases, as used in [16], [18], [19].

C. Problem statement: analysis of linear networked systems

We consider a network of linear heterogeneous subsystems interacting over a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Each node in \mathcal{V} represents a subsystem, and the edge $(i, j) \in \mathcal{E}$ means that subsystem i exerts dynamical influence on subsystem j . The dynamics of subsystem $i \in \mathcal{V}$ are written as

$$\begin{aligned} \dot{x}_i(t) &= A_{ii}x_i(t) + \sum_{j \in \mathbb{N}_i} A_{ij}x_j(t) + B_iw_i(t), \\ y_i(t) &= C_ix_i(t) + D_iw_i(t), \end{aligned} \quad (3)$$

where $x_i \in \mathbb{R}^{\alpha_i}, y_i \in \mathbb{R}^{d_i}, w_i \in \mathbb{R}^{m_i}$ represent the local state, output and disturbance, respectively, and \mathbb{N}_i denotes the neighbours of node i . By collecting the subsystems' states, the overall state-space model is then written concisely as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t), \\ y(t) &= Cx(t) + Dw(t), \end{aligned} \quad (4)$$

where $x = [x_1^T, x_2^T, \dots, x_n^T]^T$, and the vectors y, w are defined similarly. The matrix A is composed by blocks A_{ij} , and enjoys a block sparsity pattern $A \in \mathbb{R}_{\alpha}^{N \times N}(\mathcal{E}, 0)$. The matrices B, C, D have block-diagonal structures.

In this paper, we develop scalable methods for three analysis problems [20] of the linear networked system (4):

1) *Stability analysis:* System (4) with $w = 0$ is asymptotically stable if and only if the Lyapunov linear matrix inequality (LMI) is feasible

$$\begin{aligned} \text{find } P &\succ 0 \\ \text{subject to } A^T P + PA &\prec 0. \end{aligned} \quad (5)$$

2) *\mathcal{H}_2 performance:* The \mathcal{H}_2 performance of a stable system (4) with $D = 0$ can be computed as

$$\begin{aligned} \min_P \quad & \text{Tr}(B^T P B) \\ \text{subject to } & A^T P + PA + C^T C \prec 0, \\ & P \succ 0, \end{aligned} \quad (6)$$

where $\|C(sI - A)^{-1}B\|_{\mathcal{H}_2} = \sqrt{\text{Tr}(B^T P B)}$.

3) *\mathcal{H}_{∞} performance:* The \mathcal{H}_{∞} performance of a stable system (4) can be computed as

$$\begin{aligned} \min_P \quad & \gamma \\ \text{subject to } & \begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} \prec 0, \\ & P \succ 0, \end{aligned} \quad (7)$$

where $\|C(sI - A)^{-1}B + D\|_{\mathcal{H}_{\infty}} = \gamma$.

Problems (5)-(7) are convex, and ready to be solved via existing interior-point solvers, such as SeDuMi [21].

The main difficulty is that standard interior-point solvers suffer from scalability for large-scale problem instances. One major reason is that the constraints in (5)-(7) are imposed on the global system and consequently the computational complexity grows very quickly as the number of subsystems increases. Typically, the system graph \mathcal{G} is sparse for practical large-scale systems, meaning that each subsystem only has physical connections with a few other subsystems. In this paper, we aim to exploit this sparsity in the algorithmic level to solve (5)-(7) efficiently.

Remark 2: Note that there are also other efficient formulations to test stability, and compute \mathcal{H}_2 and \mathcal{H}_∞ performance [20]. An additional benefit of problems (5)-(7) is that we obtain an appropriate Lyapunov function $V(x) = x^T P x$. Also, problems (5)-(7) are helpful for some synthesis problems via a standard change of variables. In this paper, we will focus on (5)-(7), and introduce a scalable approach to solve them efficiently when the system graph \mathcal{G} is sparse.

III. CHORDAL DECOMPOSITION IN SPARSE SDPs

In this section, we focus on the SDP formulations of the optimization problems (5)-(7) and explain how to decompose them using chordal graph theory. The standard *primal form* of an SDP is

$$\begin{aligned} \min_X \quad & \langle A_0, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\ & X \succeq 0, \end{aligned} \quad (8)$$

and its standard *dual form* is

$$\begin{aligned} \max_{y, Z} \quad & \langle b, y \rangle \\ \text{subject to} \quad & Z + \sum_{i=1}^m y_i A_i = A_0, \\ & Z \succeq 0, \end{aligned} \quad (9)$$

where X is the primal variable, y, Z are the dual variables, and $b \in \mathbb{R}^m, A_i \in \mathbb{S}^N, i = 0, 1, \dots, m$ are problem data.

A. Chordal decomposition of sparse block PSD matrices

Here, we introduce two key theorems that decompose $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, ?)$ and $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, 0)$ into a set of smaller and coupled cones, respectively. Given a partition $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ and a clique \mathcal{C}_k of \mathcal{G} , we define an *index matrix* $E_{\mathcal{C}_k} \in \mathbb{R}^{|\mathcal{C}_k| \times N}$ with $|\mathcal{C}_k| = \sum_{i \in \mathcal{C}_k} \alpha_i$ and $N = \sum_{i=1}^n \alpha_i$ as

$$(E_{\mathcal{C}_k})_{ij} = \begin{cases} I_{\alpha_i}, & \text{if } \mathcal{C}_k(i) = j \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{C}_k(i)$ denotes the i -th node in \mathcal{C}_k , sorted in the natural ordering. Given a block matrix $X \in \mathbb{S}^N$ with α -partition, $E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T \in \mathbb{S}^{|\mathcal{C}_k|}$ extracts a principal submatrix defined by the clique \mathcal{C}_k , and the operation $E_{\mathcal{C}_k}^T Y E_{\mathcal{C}_k}$ inflates a $|\mathcal{C}_k| \times |\mathcal{C}_k|$ matrix into a sparse $N \times N$ matrix. Then, we have:

Theorem 1 (Generalized Grone's theorem): Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph with a set of maximal cliques $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$. Given a partition $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ and $N = \sum_{i=1}^n \alpha_i$, then, $X \in \mathbb{S}_{\alpha,+}^N(\mathcal{E}, ?)$ if and only if

$$E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T \in \mathbb{S}_+^{|\mathcal{C}_k|}, \quad k = 1, \dots, p. \quad (10)$$

Theorem 2 (Generalized Agler's theorem): Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph with a set of maximal cliques $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$. Given a partition $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ and $N = \sum_{i=1}^n \alpha_i$, then, $Z \in \mathbb{S}_{\alpha,+}^N(\mathcal{E}, 0)$ if and only if there exist matrices $Z_k \in \mathbb{S}_+^{|\mathcal{C}_k|}$ for $k = 1, \dots, p$ such that

$$Z = \sum_{k=1}^p E_{\mathcal{C}_k}^T Z_k E_{\mathcal{C}_k}. \quad (11)$$

For the scalar case, *i.e.*, $\alpha = \{1, 1, \dots, 1\}$, Theorems 1 and 2 reduce to the Grone's [22] and Agler's [23] theorems, respectively. Here, we show that these two celebrated theorems can be generalized into sparse block matrices with an arbitrary partition. Due to lack of space, the proofs are omitted here¹. One direct benefit of the generalized Grone's and Agler's theorems is the convenience for networked system analysis, *i.e.*, they allow us to solely focus on the physical connections characterized by \mathcal{G} to exploit scalability. In this paper, *chordal decomposition* refers to the application of Theorems 1 and 2 to decompose a large sparse block PSD cone into a set of smaller but coupled PSD cones. In the next section, we summarize a recent first-order algorithm [19] that can exploit Theorems 1 and 2.

B. Chordal decomposition in first-order methods

Suppose the data matrices in (8) and (9) have an aggregate sparsity pattern: $A_0, A_1, \dots, A_m \in \mathbb{S}_{\alpha}^N(\mathcal{E}, 0)$. It is assumed that the pattern \mathcal{E} is chordal (otherwise, a suitable chordal extension can be found; as stated in Remark 1, making an extension does not affect the problem data), with a set of maximal cliques $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p$. Note that the cost function and equality constraints in (8) only depend on the entries X_{ij} on its diagonal and $(i, j) \in \mathcal{E}$. The remaining elements simply guarantee that the matrix is PSD. Also, in (9) any feasible solution Z satisfies the sparsity pattern $\mathbb{S}_{\alpha}^N(\mathcal{E}, 0)$. Recalling the definition of $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, ?)$ and $\mathbb{S}_{\alpha,+}^N(\mathcal{E}, 0)$, and according to Theorems 1 and 2, we can equivalently reformulate the primal SDP (8) and the dual SDP (9), respectively, as

$$\begin{aligned} \min_{X, X_k} \quad & \langle A_0, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m, \\ & X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T, \quad k = 1, \dots, p, \\ & X_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, \quad k = 1, \dots, p, \end{aligned} \quad (12)$$

and

$$\begin{aligned} \max_{y, Z_k, V_k} \quad & \langle b, y \rangle \\ \text{subject to} \quad & \sum_{k=1}^p E_{\mathcal{C}_k}^T V_k E_{\mathcal{C}_k} + \sum_{i=1}^m y_i A_i = A_0, \\ & Z_k = V_k, \quad k = 1, \dots, p, \\ & Z_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, k = 1, \dots, p. \end{aligned} \quad (13)$$

In (12) and (13), the original single large PSD cone has been replaced by multiple smaller PSD cones, coupled by a

¹See proofs via <http://sysos.eng.ox.ac.uk/wiki/images/7/7c/ECC2018.pdf>

set of consensus variables. Then, first-order methods can be applied to the decomposed formulations (12) and (13) [24], or their homogeneous self-dual embedding [19], which lead to algorithms only involving parallel PSD projections onto p smaller cones and a projection onto an affine set at each iteration. If the size of the largest maximal clique is small, then the reduction of cone dimensions enables us to compute PSD projections much more efficiently. Consequently, the application of first-order methods in the decomposed problems (12) and (13) improves the scalability to solve sparse SDPs when seeking a solution of moderate accuracy. The interested reader is referred to [19], [24] for details. The MATLAB package CDCS [25] provides an efficient implementation of the above decomposition method.

Remark 3: Note that chordal sparsity has been identified as a key structure in large-scale sparse SDPs. In addition to the decomposition approach [19], there are several other methods to exploit chordal properties (Theorems 1 and 2) in solving sparse SDPs, *e.g.*, [18], [26]. In this paper, we use the decomposition approach [19] as an example to show the computational benefits brought by exploiting chordal sparsity in the analysis of large-scale networked systems.

IV. SCALABLE PERFORMANCE ANALYSIS OF SPARSE SYSTEMS

This section applies the chordal decomposition techniques in stability, \mathcal{H}_2 and \mathcal{H}_∞ analyses of linear networked systems. Our strategy is to restrict the sparsity pattern of P , such that the sparsity structure of the dynamical system is preserved in the SDP formulations of (5)-(7). This allows one to decompose a single large PSD constraint into multiple smaller ones using chordal decomposition, thus facilitating their solutions using sparse optimization techniques [25]. We note that the proposed method of this section may introduce certain conservatism for general networked systems since we use a sparse Lyapunov function.

A. Stability verification

We first show that the sparsity pattern of $A^T P + PA$ reflects the aggregate sparsity pattern of the resulting SDP formulation. Let us write the Lyapunov LMI as

$$\begin{bmatrix} -P & 0 \\ 0 & A^T P + PA \end{bmatrix} \prec 0. \quad (14)$$

There are up to $m = \frac{N(N+1)}{2}$ free variables in matrix P . We denote W_1, W_2, \dots, W_m as the standard basis matrices for \mathbb{S}^N , and define the matrices $A_1, A_2, \dots, A_m \in \mathbb{S}^{2N}$ as

$$A_i = \begin{bmatrix} -W_i & 0 \\ 0 & A^T W_i + W_i A \end{bmatrix}, i = 1, 2, \dots, m. \quad (15)$$

Then, (14) can be reformulated into a standard dual SDP

$$\begin{aligned} & \max_{y, Z} \quad \langle b, y \rangle \\ & \text{subject to} \quad Z + \sum_{i=1}^m y_i A_i = A_0, \\ & \quad \quad \quad Z \succeq 0, \end{aligned} \quad (16)$$

where $y \in \mathbb{R}^m$, $Z \in \mathbb{S}_+^{2N}$, $A_0 = -\epsilon I$, $\epsilon > 0$, $b = 0$, and A_i is defined in (15). At this point, we know that the aggregate sparsity pattern of (16) is

$$\mathcal{P}(A_0) \cup \mathcal{P}(A_1) \cup \dots \cup \mathcal{P}(A_m) = \mathcal{P} \left(\begin{bmatrix} -P & 0 \\ 0 & A^T P + PA \end{bmatrix} \right),$$

where $\mathcal{P}(\cdot)$ denotes the sparsity pattern of a matrix. The aggregate sparsity pattern of (16) directly depends on the sparsity patterns of P and $A^T P + PA$.

For a networked system (4), matrix A has an inherent structure described by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, *i.e.*, $A \in \mathbb{R}_\alpha^{N \times N}(\mathcal{E}, 0)$, where $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ denotes the dimensions of local states. Apparently, a dense P has no conservatism in certifying stability, but leads to a full pattern of $A^T P + PA$. To preserve the sparsity structure \mathcal{G} , we consider the following problem.

Q* Given a sparse $A \in \mathbb{R}_\alpha^{N \times N}(\mathcal{E}, 0)$, find a sparsity pattern of P , such that the sparsity pattern of $A^T P + PA$ inherits the original pattern \mathcal{E} (more favorably, the resulting pattern is chordal with small maximal cliques).

We note that a complete answer to Q* is difficult for general systems, especially considering the relationship between sparsity (*i.e.*, efficiency) and conservativeness. One trivial choice is a block-diagonal P with block sizes compatible with the subsystem sizes α_i . Then, the graph structure in the dynamical system (3) is naturally inherited in (16), *i.e.*,

$$A^T P + PA \in \mathbb{S}_\alpha^N(\mathcal{E} \cup \mathcal{E}_r, 0), \quad (17)$$

where \mathcal{E}_r denotes a set of reverse edges obtained by reversing the order of the pairs in \mathcal{E} . We note that the existence of block-diagonal P is investigated in [11], and diagonal P is necessary and sufficient to certify stability of positive systems [8]. Other choices of P are available for special graphs \mathcal{G} , such as trees and cycles [13].

In this paper, we assume the pattern of $A^T P + PA$ can be described by a chordal graph $\mathcal{G}_c(\mathcal{V}, \mathcal{E}_c)$ with a set of maximal cliques $\mathcal{C}_1, \dots, \mathcal{C}_p$. As mentioned above, one basic choice is a block-diagonal P . If $\mathcal{E} \cup \mathcal{E}_r$ is non-chordal, then we can make a chordal extension to get \mathcal{E}_c . In (16), the single large PSD cone $Z \succeq 0$ has two blocks, where the upper-left one corresponds to block-diagonal P and the bottom-right one can be replaced by $\mathbb{S}_{\alpha,+}^N(\mathcal{E}_c, 0)$. Then, $\mathbb{S}_{\alpha,+}^N(\mathcal{E}_c, 0)$ can be subsequently decomposed into multiple smaller cones using chordal decomposition (see Theorem 1 and the reformulations (12) and (13)). Consequently, if the largest maximal clique is small, the SDP formulation (16) can be expected to be solved efficiently for sparse systems using the first-order method described in Section III-B.

B. \mathcal{H}_2 performance

Similar to the stability analysis, the \mathcal{H}_2 optimization problem (6) can be reformulated into a standard SDP of primal form (8) or dual form (9). The aggregate sparsity pattern of the resulting SDP is determined by the pattern of $A^T P + PA + C^T C$. Considering the structure of networked system (3), we have

$$\mathcal{P}(A^T P + PA + C^T C) = \mathcal{P}(A^T P + PA). \quad (18)$$

Then, the argument for stability analysis can be applied to \mathcal{H}_2 analysis for the purpose of scalable computation. We assume the pattern of $A^T P + PA$ can be described by a chordal graph $\mathcal{G}_c(\mathcal{V}, \mathcal{E}_c)$, leading to $A^T P + PA + C^T C \in \mathbb{S}_\alpha^N(\mathcal{E}_c, 0)$. Consequently, the sparse optimization technique [25] is ready to solve the decomposed version of (6). Note that we can only obtain an approximated (upper bound) \mathcal{H}_2 performance in general due to using a sparse P .

C. \mathcal{H}_∞ performance

When reformulating the \mathcal{H}_∞ analysis problem (7) into a standard SDP, the aggregate sparsity pattern depends on the pattern of the following matrix

$$M = \begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix}. \quad (19)$$

According to the inherent structure in (3), we know $A \in \mathbb{R}_\alpha^{N \times N}(\mathcal{E}, 0)$ and B, C, D are block-diagonal. If we restrict P to be block-diagonal with compatible block sizes, then the entry PB is also block-diagonal. The sparsity pattern of the first block on the diagonal is $A^T P + PA \in \mathbb{S}_\alpha^N(\mathcal{E}_c, 0)$, where \mathcal{E}_c is the chordal extension of $\mathcal{E} \cup \mathcal{E}_r$, defined in Section IV-A. Then, we have the following result.

Theorem 3: Consider a networked system with dynamics (4) and a block-diagonal P . Suppose that the sparsity pattern of $A^T P + PA$ has p maximal cliques $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p$, and the cardinality of the largest maximal clique is h . Then,

- 1) the block matrix M in (19) has a partition $\hat{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n, m_1, m_2, \dots, m_n, d_1, d_2, \dots, d_n\}$;
- 2) the sparsity pattern of M , denoted as $M \in \mathbb{S}_{\hat{\alpha}}^{\hat{N}}(\hat{\mathcal{E}}, 0)$, has $p + n$ maximal cliques, and the cardinality of the largest maximal clique is $\max\{h, 3\}$.

Proof: According to (3), it is straightforward to see that the block matrix M in (19) has a partition $\hat{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n, m_1, m_2, \dots, m_n, d_1, d_2, \dots, d_n\}$, where $\alpha_i, m_i, d_i (i = 1, \dots, n)$ are the dimensions of local states, disturbances and outputs, respectively.

Let us first consider the following block

$$M_1 = \begin{bmatrix} A^T P + PA & PB \\ B^T P & -\gamma I \end{bmatrix} \in \mathbb{S}_{\hat{\alpha}_1}^{\hat{N}_1}(\hat{\mathcal{E}}_1, 0), \quad (20)$$

where the partition $\hat{\alpha}_1 = \{\alpha_1, \alpha_2, \dots, \alpha_n, m_1, m_2, \dots, m_n\}$ and $\hat{N}_1 = \sum_{i=1}^n (\alpha_i + m_i)$. Since PB and γI are block diagonal, every node $i \in \{n+1, \dots, n+n\}$ is only connected to one node $i - n$. Then, the edge set for M_1 is shown as

$$\hat{\mathcal{E}}_1 = \mathcal{E}_c \cup \{(i, i+n) \mid i = 1, \dots, n\}, \quad (21)$$

indicating that the maximal cliques of $\hat{\mathcal{E}}_1$ are given by

$$\mathcal{C}_1, \dots, \mathcal{C}_p, \mathcal{C}_{p+i} = \{i, i+n\}, i = 1, \dots, n. \quad (22)$$

Next, according to (19) and (20), we know

$$M = \begin{bmatrix} M_1 & H \\ H^T & -\gamma I \end{bmatrix} \in \mathbb{S}_{\hat{\alpha}}^{\hat{N}}(\hat{\mathcal{E}}, 0), \quad (23)$$

where $H^T = [C \ D]$ and $\hat{N} = \sum_{i=1}^n (\alpha_i + m_i + d_i)$. Since the matrices C, D and γI are block diagonal, every node

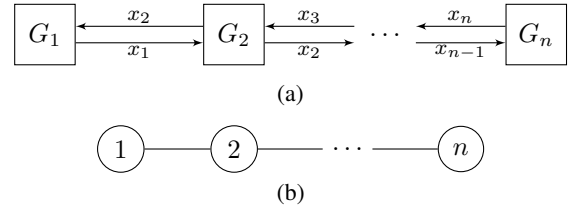


Fig. 2: (a) A chain of n subsystems; (b) a simplified line graph.

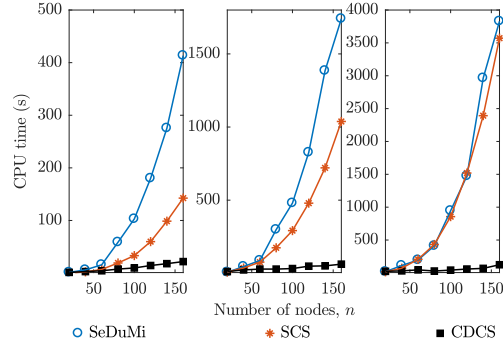


Fig. 3: CPU time (s) required by SeDuMi, SCS and CDCS to solve the SDPs of the analysis problems (5)-(7) of a chain of subsystems.

$i \in \{2n+1, \dots, 2n+n\}$ is connected to another two nodes $i-n, i-2n$. Consequently, the edge set for M is given by

$$\hat{\mathcal{E}} = \hat{\mathcal{E}}_1 \cup \{(i, i+2n), (i+n, i+2n) \mid i = 1, \dots, n\}.$$

According to the edge set $\hat{\mathcal{E}}_1$ (21), we know that in the edge set $\hat{\mathcal{E}}$, $\{i, i+n, i+2n\}$ forms a maximal clique. This implies that the maximal cliques of $\hat{\mathcal{E}}$ are

$$\mathcal{C}_1, \dots, \mathcal{C}_p, \mathcal{C}_{p+i} = \{i, i+n, i+2n\}, i = 1, \dots, n. \quad (24)$$

Therefore, $\mathbb{S}_{\hat{\alpha}}^{\hat{N}}(\hat{\mathcal{E}}, 0)$ has $p + n$ maximal cliques, and the cardinality of the largest maximal clique is $\max\{h, 3\}$. ■

Although \mathcal{H}_∞ analysis problem (7) appears to be more complex than the Lyapunov LMI (5), Theorem 3 shows that the underlying maximal cliques are similar and that the cardinality of the largest maximal clique for (7) and (5) is almost identical. Therefore, the strategy for stability analysis can be applied to \mathcal{H}_∞ analysis problem (7): the single large PSD cone can be decomposed into $p + n$ smaller ones, and the sparse optimization technique [25] can be used to solve the decomposed problem in a scalable fashion.

V. NUMERICAL SIMULATIONS

To show the efficiency of the chordal decomposition approach, we consider a chain of n subsystems where each subsystem has physical interactions with its two neighbouring ones except the first and last subsystem, which only interacts with one neighbouring subsystem; see Fig. 2 (a) for illustration. A simplified version of this chain is shown in Fig. 2 (b). In this case, the maximal cliques of the graph in Fig. 2 (b) are $\{i, i+1\}, i = 1, \dots, n-1$, and the cardinality of the largest maximal clique is only 2.

We solved the SDP formulations of stability analysis (5), \mathcal{H}_2 performance (6), and \mathcal{H}_∞ performance (7) using standard

TABLE I: Approximated \mathcal{H}_2 and \mathcal{H}_∞ performance of a chain of subsystems computed by different solvers.

n	\mathcal{H}_2				\mathcal{H}_∞			
	\dagger	sedumi	SCS	CDCS	\ddagger	sedumi	SCS	CDCS
20	9.70	17.73	17.73	17.73	3.65	3.66	3.70	3.66
40	11.66	20.07	20.07	20.07	3.67	3.68	3.74	3.68
60	14.72	25.78	25.79	25.78	3.75	3.77	3.85	3.77
80	16.85	28.70	28.71	28.69	4.32	4.34	4.37	4.34
100	18.08	29.88	29.91	29.88	3.91	3.92	3.96	3.92
120	19.71	32.10	32.12	32.09	4.02	4.03	4.10	4.04
140	21.51	35.59	35.64	35.58	4.09	4.10	4.16	4.11
160	24.64	40.65	40.73	40.65	4.07	4.08	4.18	4.09

\dagger : Accurate \mathcal{H}_2 performance returned by the MATLAB routine `norm(sys,2)`.

\ddagger : Accurate \mathcal{H}_∞ performance returned by the MATLAB routine `norm(sys,inf)`.

dense solvers: SeDuMi [21] and SCS [27], as well as using the sparse conic solver CDCS [25] that exploits chordal sparsity. Block-diagonal P was used in the formulations. For the interior-point solver SeDuMi, we used its default parameters, and the first-order solvers SCS and CDCS were called with termination tolerance 10^{-4} and number of iterations limited to 2000. All simulations were run on a PC with a 2.8 GHz Intel Core i7 CPU and 8GB of RAM. In the simulations, the state dimension n_i was chosen randomly from 5 to 10, and the dimensions of output and disturbance (d_i, m_i) were chosen randomly from 1 to 5. Then, we generated random matrices A_{ii}, A_{ij}, B_i, D_i and imposed the global state matrix A with negative eigenvalues by setting $A := A - (\lambda_{\max} + 5)I$, where λ_{\max} denotes the maximum real part of the eigenvalues of A .

Fig. 3 shows the CPU time in seconds required by the solvers for testing stability, and computing approximated \mathcal{H}_2 and \mathcal{H}_∞ performance. The chordal decomposition approach (via CDCS) took significantly less time than standard dense methods (using either SeDuMi or SCS). Moreover, the CPU time required by CDCS seems to grow linearly as the system size increases. This is expected since the size of the largest maximal clique is fixed for a line graph, indicating that the size of the PSD cones after decomposition is fixed and only the number of PSD cones increases linearly as growth of the graph size. Finally, Table I lists the \mathcal{H}_2 and \mathcal{H}_∞ performance computed by different solvers. We can see that using block-diagonal P indeed brought certain conservatism when searching for performance bounds.

VI. CONCLUSION

In this paper, we have introduced scalable analysis techniques for sparse linear networked systems by exploiting chordal decomposition and using a recent first-order algorithm. The numerical results have shown that when the largest maximal clique is small, the chordal decomposition approach is significantly faster than the standard dense method. This makes it a promising approach for large sparse systems analysis. Future work will consider non-block diagonal Lyapunov functions that can preserve the sparsity pattern in the analysis problems. Also, there are several interior-point methods that are able to exploit chordal properties in solving sparse SDPs [18], [26]. It will be interesting to apply these solvers in sparse systems analysis and synthesis as well.

REFERENCES

[1] D. D. Siljak, *Decentralized control of complex systems*. Courier Corporation, 2011.

[2] F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo, “Sparsity-promoting optimal wide-area control of power networks,” *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2281–2291, 2014.

[3] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, “Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies,” *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 899–910, 2017.

[4] P. Moylan and D. Hill, “Stability criteria for large-scale systems,” *IEEE Trans. Automat. Contr.*, vol. 23, no. 2, pp. 143–149, 1978.

[5] C. Meissen, L. Lessard, M. Arcak, and A. K. Packard, “Compositional performance certification of interconnected systems using ADMM,” *Automatica*, vol. 61, pp. 55–63, 2015.

[6] J. Anderson and A. Papachristodoulou, “A decomposition technique for nonlinear dynamical system analysis,” *IEEE Trans. Automat. Contr.*, vol. 57, no. 6, pp. 1516–1521, 2012.

[7] L. Farina and S. Rinaldi, *Positive linear systems: theory and applications*. John Wiley & Sons, 2011, vol. 50.

[8] A. Rantzer, “Scalable control of positive systems,” *Eur. J. Control.*, vol. 24, pp. 72–80, 2015.

[9] T. Tanaka and C. Langbort, “The bounded real lemma for internally positive systems and H-infinity structured static state feedback,” *IEEE Trans. Automat. Contr.*, vol. 56, no. 9, pp. 2218–2223, 2011.

[10] A. Sootla and A. Rantzer, “Scalable positivity preserving model reduction using linear energy functions,” in *Proc. 51st IEEE Conf. Decision Control*. IEEE, 2012, pp. 4285–4290.

[11] A. Sootla, Y. Zheng, and A. Papachristodoulou, “Block-diagonal solutions to Lyapunov inequalities and generalisations of diagonal dominance,” in *Proc. 56th IEEE Conf. Decision Control*. IEEE, 2017, pp. 6561–6566.

[12] A. Rantzer, “Distributed performance analysis of heterogeneous systems,” in *Proc. 49th IEEE Conf. Decision Control*. IEEE, 2010, pp. 2682–2685.

[13] R. P. Mason and A. Papachristodoulou, “Chordal sparsity, decomposing SDPs and the Lyapunov equation,” in *Proc. American Control Conf.* IEEE, 2014, pp. 531–537.

[14] M. Andersen, S. Pakazad, A. Hansson, and A. Rantzer, “Robust stability analysis of sparsely interconnected uncertain systems,” *IEEE Trans. Autom. Control*, vol. 59, no. 8, pp. 2151–2156, 2014.

[15] Y. Zheng, R. P. Mason, and A. Papachristodoulou, “Scalable design of structured controllers using chordal decomposition,” *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 752–767, March 2018.

[16] L. Vandenberghe and M. S. Andersen, “Chordal graphs and semidefinite optimization,” *Found. Trends Optim.*, vol. 1, pp. 241–433, 2014.

[17] R. E. Tarjan and M. Yannakakis, “Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs,” *SIAM J. Comput.*, vol. 13, no. 3, pp. 566–579, 1984.

[18] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, “Exploiting sparsity in semidefinite programming via matrix completion I: General framework,” *SIAM J. Optimiz.*, vol. 11, no. 3, pp. 647–674, 2001.

[19] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, “Chordal decomposition in operator-splitting methods for sparse semidefinite programs,” *arXiv preprint arXiv:1707.05058*, 2017.

[20] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*. Springer Science & Business Media, 2013, vol. 36.

[21] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optim. Methods Softw.*, vol. 11, no. 1–4, pp. 625–653, 1999.

[22] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, “Positive definite completions of partial hermitian matrices,” *Linear algebra and its applications*, vol. 58, pp. 109–124, 1984.

[23] J. Agler, W. Helton, S. McCullough, and L. Rodman, “Positive semidefinite matrices with a given sparsity pattern,” *Linear Algebra Appl.*, vol. 107, pp. 101–149, 1988.

[24] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, “Fast ADMM for Semidefinite Programs with Chordal Sparsity,” in *Proc. American Control Conf.* IEEE, 2017, pp. 3335–3340.

[25] —, “CDCS: Cone decomposition conic solver, version 1.1,” <https://github.com/oxfordcontrol/CDCS>, Sep. 2016.

[26] M. S. Andersen, J. Dahl, and L. Vandenberghe, “Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones,” *Math. Prog. Computat.*, vol. 2, no. 3–4, pp. 167–201, 2010.

[27] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd, “SCS: Splitting conic solver, version 1.2.6,” <https://github.com/cvxgrp/scs>, Apr. 2016.