

Chordal Sparsity, Decomposing SDPs and the Lyapunov Equation

Richard P. Mason and Antonis Papachristodoulou

Abstract—Analysis questions in control theory are often formulated as Linear Matrix Inequalities and solved using convex optimisation algorithms. For large LMIs it is important to exploit structure and sparsity within the problem in order to solve the associated Semidefinite Programs efficiently. In this paper we decompose SDPs by taking advantage of chordal sparsity, and apply our method to the problem of constructing Lyapunov functions for linear systems. By choosing Lyapunov functions with a chordal graphical structure we convert the semidefinite constraint in the problem into an equivalent set of smaller semidefinite constraints, thereby facilitating the solution of the problem. The approach has the potential to be applied to other problems such as stabilising controller synthesis, model reduction and the KYP lemma.

I. INTRODUCTION

Many problems in control theory can be formulated as Linear Matrix Inequalities (LMIs) and solved using convex optimisation algorithms [1], [2]. In this paper we will focus on the Lyapunov LMI: given $A \in \mathbb{R}^{n \times n}$,

$$\text{find } P \succ 0 \text{ such that } Q = A^T P + P A \prec 0, \quad (1)$$

where $A \succ B$ denotes that $A - B$ is positive definite. It is well known that this problem can be solved using linear algebra by picking a $Q \prec 0$ and solving for P [3].

Our motivation for studying the Lyapunov LMI is that it appears as a block within many key LMIs in control theory. For example, it appears within the KYP-lemma and the simultaneous stabilisation problem. For some of these problems there is no analytical solution and so we turn to iterative algorithms such as interior-point methods to solve the associated SDPs.

The problem is that these interior-point methods do not scale well when P is a dense matrix. One approach to mitigate this problem is to restrict P to be a sparse matrix as this reduces the number of free variables in the LMI [4]. In general this is conservative, but it is known that for certain classes of systems a sparse P is sufficient to find a feasible solution to the Lyapunov LMI [5], [6].

In this paper we use two theorems from linear algebra that connect positive semidefinite matrices and chordal graphs and apply them to solve (1). These results first appeared in papers [7], [8] and have since been applied by several researchers in optimisation to decompose large SDPs, see [9]–[12].

R. P. M. and A. P. are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK. R. P. M. was supported by the Life Sciences Interface Doctoral Training Centre at the University of Oxford. A. P. was supported in part by the Engineering and Physical Sciences Research Council projects EP/J012041/1, EP/I031944/1 and EP/J010537/1. {richard.mason, antonis}@eng.ox.ac.uk

We apply these theorems to the Lyapunov LMI by choosing the sparsity pattern of P so that P and Q have chordal sparsity patterns. This enables us to decompose the constraint in the Lyapunov LMI into multiple constraints on submatrices of P and Q . When the resulting LMI is sparse this decomposition method allows us to solve the problem significantly faster than by using the standard dense method.

A. Solving the Lyapunov LMI by Interior-Point Methods

We wish to reformulate (1) as a standard SDP to see how the density of P affects the computational difficulty of the problem. First we write the Lyapunov LMI in the form

$$\begin{bmatrix} -P & 0 \\ 0 & A^T P + P A \end{bmatrix} \prec 0. \quad (2)$$

Note that since P is an $n \times n$ symmetric matrix it has up to $m = n(n+1)/2$ free variables. Let W_1, W_2, \dots, W_m be the standard basis matrices for \mathbb{S}^n and define the matrices $A_1, A_2, \dots, A_m \in \mathbb{S}^{2n}$ by

$$A_i = \begin{bmatrix} -W_i & 0 \\ 0 & A^T W_i + W_i A \end{bmatrix}, \quad i = 1, 2, \dots, m. \quad (3)$$

We may then formulate the LMI feasibility problem as the constraint of an SDP

$$\begin{aligned} & \text{maximise} && b^T y \\ & \text{subject to} && \sum_{i=1}^m y_i A_i + Z = A_0 \\ & && Z \succeq 0 \end{aligned} \quad (4)$$

where $y \in \mathbb{R}^m$, $Z \in \mathbb{S}^{2n}$, $b = 0$ and $A_0 = -\epsilon I$, $\epsilon > 0$. The dual SDP to (4) is

$$\begin{aligned} & \text{minimize} && A_0 \bullet X \\ & \text{subject to} && A_i \bullet X = b_i, \quad i = 1, \dots, m \\ & && X \succeq 0 \end{aligned} \quad (5)$$

where $X \in \mathbb{S}^{2n}$ and $A \bullet B = \text{Tr}(A^T B) = \sum_{i,j} A_{ij} B_{ij}$. The SDP above can then be solved using a primal-dual interior-point method (via a self-dual embedding) [13]–[16]. Primal-dual interior-point methods generate a sequence of points (X^k, y^k, Z^k) in the interior of the feasible set using iterations of the form

$$(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k, y^k, Z^k) + \alpha^k (\Delta X, \Delta y, \Delta Z)$$

where α^k is a step length and $(\Delta X, \Delta y, \Delta Z)$ is a Newton step direction that must be computed at each iteration.

Calculating the step direction involves finding the solution to a square system of linear equations called the Schur complement system. When P is dense the size of the Schur

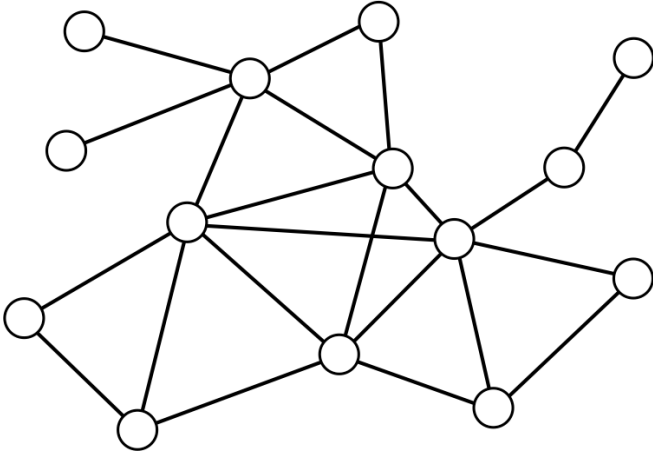


Fig. 1: An example of a chordal graph.

complement system is $m = n(n+1)/2$ and hence when n is large the time required to solve the LMI can be prohibitive. This motivates us to explore sparse P matrices, and in particular we will consider the case where P and Q have a chordal sparsity pattern.

II. PRELIMINARIES

Let $G = (V, E)$ be an undirected graph, with set of nodes $V = \{1, 2, \dots, N\}$ and set of edges $E \subseteq V \times V$. For a subset $V' \subset V$, the induced subgraph is the graph $G = (V', E')$ with vertex set V' and edge set $E' = E \cap (V' \times V')$. A clique is a subset $C \subset V$ such that for any $i, j \in C$, $(i, j) \in E$. A maximal clique is a clique that is not a subset of another clique. A cycle is a sequence of pairwise distinct vertices $\gamma = (v_1, v_2, \dots, v_s)$ having the property that

$$(v_1, v_2), (v_2, v_3), \dots, (v_{s-1}, v_s), (v_s, v_1) \in E$$

and s is the length of the cycle. A chord of the cycle γ is an edge $(v_i, v_j) \in E$ where $1 \leq i < j \leq s$, $(i, j) \neq (1, s)$, and $|i - j| \geq 2$. A graph is chordal if every cycle of length ≥ 4 has a chord, see Figure 1.

A vertex v is called simplicial if all its neighbours are adjacent to each other. A bijection $\alpha : V \mapsto \{1, \dots, N\}$ is called an ordering of G . For simplicity we write an ordering as $\alpha = \langle v_1, v_2, \dots, v_N \rangle$ where $v_i = \alpha^{-1}(i)$. An ordering α of G is a perfect elimination ordering if, for each $i \in \{1, \dots, N\}$, v_i is a simplicial vertex of the subgraph induced by $\{v_i, \dots, v_N\}$.

Theorem 1: [17] G is chordal if and only if G has a perfect elimination ordering.

Let G be a connected graph with set of maximal cliques $\mathcal{C} = \{C_1, C_2, \dots, C_p\}$. A clique tree, $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ with $\mathcal{E} \subseteq \mathcal{C} \times \mathcal{C}$, is a connected graph with no cycles. A clique tree is said to satisfy the clique intersection property if for every pair of distinct $C_i, C_j \in \mathcal{C}$, the set $C_i \cap C_j$ is contained in every clique on the unique path connecting C_i and C_j in the clique tree.

Theorem 2: [18] A connected graph G is chordal if and only if there exists a clique tree $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ for which the clique intersection property holds.

Let $J(C) = \{(i, j) \in C \times C \mid 1 \leq i \leq j \leq n\}$ for every $C \subseteq V$. In a connected graph the maximal cliques will overlap, i.e., $C_i \cap C_j \neq \emptyset$ for some $C_i, C_j \in \mathcal{C}$. Given a clique tree that satisfies the clique intersection property we denote the minimal set of overlapping elements by Λ , where

$$\Lambda = \{(i, j, k, l) \mid (i, j) \in J(C_k \cap C_l), (C_k, C_l) \in \mathcal{E}\}.$$

III. CHORDAL DECOMPOSITIONS OF POSITIVE SEMIDEFINITE MATRICES

Let $G = (V, E)$ be an undirected graph and assume that $(i, i) \in E$ for all $i \in V$, i.e., each node has a self-loop. A partial symmetric matrix, X , is a symmetric $N \times N$ matrix where element X_{ij} is specified if and only if $(i, j) \in E$. A completion of X is an $N \times N$ matrix M which satisfies $M_{ij} = X_{ij}$ for all $(i, j) \in E$. We say that M is a positive completion of X if and only if M is a completion of X and M is positive semidefinite. We will use the following notation:

$\mathbb{S}^N(E, ?)$ = the set of $N \times N$ partial symmetric matrices with elements defined on E

$$\mathbb{S}_+^N(E, ?) = \{X \in \mathbb{S}^N(E, ?) \mid \exists M \succeq 0, M_{ij} = X_{ij} \forall (i, j) \in E\}$$

$$\mathbb{S}^N(E, 0) = \{X \in \mathbb{S}^N \mid X_{ij} = 0 \text{ if } (i, j) \notin E\}$$

$$\mathbb{S}_+^N(E, 0) = \{X \in \mathbb{S}^N(E, 0) \mid X \succeq 0\}$$

$$\mathbb{S}^C = \{X \in \mathbb{S}^N \mid X_{ij} = 0 \text{ if } (i, j) \notin C \times C\} \text{ for every } C \subseteq V$$

$$\mathbb{S}_+^C = \{X \in \mathbb{S}^C \mid X \succeq 0\} \text{ for every } C \subseteq V$$

$$X(C) = \bar{X} \in \mathbb{S}^C \text{ such that } \bar{X}_{ij} = X_{ij} \text{ for } (i, j) \in C \times C \text{ for every } X \in \mathbb{S}^N \text{ and every } C \subseteq V.$$

U_{ij} = the $N \times N$ symmetric matrix with 1 in the (i, j) th and (j, i) th elements and 0 elsewhere.

Theorem 3: [7] Let $G = (V, E)$ be a chordal graph with set of maximal cliques $\mathcal{C} = \{C_1, \dots, C_p\}$. Suppose that $X \in \mathbb{S}^N(E, ?)$. Then $X \in \mathbb{S}_+^N(E, ?)$ if and only if $X(C_k) \succeq 0$ for $k = 1, 2, \dots, p$.

Dual to the cone $\mathbb{S}_+^N(E, ?)$ is the cone $\mathbb{S}_+(E, 0)$ and the following can be derived from Theorem 3 using this dual relationship [19].

Theorem 4: [8] Let $G = (V, E)$ be a chordal graph with set of maximal cliques $\mathcal{C} = \{C_1, \dots, C_p\}$. Suppose that $A \in \mathbb{S}^N(E, 0)$. Then $A \in \mathbb{S}_+^N(E, 0)$ if and only if there exists a set of matrices $\{A^1, A^2, \dots, A^p\}$ such that

$$A = \sum_{k=1}^p A^k, \quad A^k \in \mathbb{S}_+^{C_k}, \quad k = 1, \dots, p.$$

For our purposes we restate Theorem 4 using clique trees.

Theorem 5: [12] Let $G = (V, E)$ be a chordal graph with set of maximal cliques $\mathcal{C} = \{C_1, \dots, C_p\}$. Suppose that $A \in \mathbb{S}^N(E, 0)$. Then $A \in \mathbb{S}_+^N(E, 0)$ if and only if the system of LMIs

$$A^k - L^k(\theta) \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p,$$

is feasible, where θ is a vector of variables of the form $\theta = (\theta_{ijkl} \mid (i, j, k, l) \in \Lambda)$ and

$$L^k(\theta) = \sum_{(i, j, l) \mid (i, j, k, l) \in \Lambda} U_{ij} \theta_{ijkl} - \sum_{(i, j, h) \mid (i, j, h, k) \in \Lambda} \theta_{ijhk} U_{ij}$$

for every $\theta = (\theta_{ijkl} \mid (i, j, k, l) \in \Lambda)$, $k \in \{1, 2, \dots, p\}$.

IV. CHORDAL DECOMPOSITION OF THE LYAPUNOV LMI

In this section we return to the SDP formulation of the Lyapunov LMI (4)-(5) and explain how it can be decomposed using Theorems 3 and 4. Suppose that $A \in \mathbb{S}^N(E, 0)$, where $G = (V, E)$ is not necessarily chordal. A chordal extension of G , denoted by $G_{ch} = (V, F)$, is a chordal graph where the edge set satisfies $E \subseteq F$. Since $(i, j) \notin E \Rightarrow A_{ij} = 0$, for an arbitrary $X \in \mathbb{S}^N$

$$A \bullet X = \sum_{(i,j) \in E} A_{ij} X_{ij} = \sum_{(i,j) \in F} A_{ij} X_{ij}. \quad (6)$$

A given graph G may have many chordal extensions and the problem of finding the chordal extension that adds the minimum number of edges is NP hard, but fortunately there are effective heuristics for finding minimal chordal extensions. In this paper we find a chordal extension for G by permuting the adjacency matrix using a minimum degree ordering and then performing a symbolic Cholesky factorisation.

Now consider (5). Suppose that $A_0, A_1, \dots, A_m \in \mathbb{S}^N(E, 0)$ and replace the inner products using (6). Then we can write the primal form of the SDP in an equivalent form

$$\begin{aligned} & \text{minimize} && \sum_{(i,j) \in F} [A_0]_{ij} X_{ij} \\ & \text{subject to} && \sum_{(i,j) \in F} [A_q]_{ij} X_{ij} = b_q, \quad q = 1, 2, \dots, m \quad (7) \\ & && X \in \mathbb{S}_+^N(F, ?), \end{aligned}$$

where we have used the fact that the values of the objective and constraint functions are not affected by completing X . From Theorem 3 the semidefinite constraint will be satisfied if and only if the submatrices of X corresponding to the maximal cliques of $G = (V, F)$ are positive semidefinite. Let $\mathcal{C} = \{C_1, C_2, \dots, C_p\}$ be the maximal cliques of $G = (V, F)$, then (7) can be written as:

$$\begin{aligned} & \text{minimize} && \sum_{(i,j) \in F} [A_0]_{ij} X_{ij} \\ & \text{subject to} && \sum_{(i,j) \in F} [A_q]_{ij} X_{ij} = b_q, \quad q = 1, 2, \dots, m \quad (8) \\ & && X^k \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p. \end{aligned}$$

We may decompose any $A_q \in \mathbb{S}(E, 0)$ into a summation of matrices of the form $A_q^k \in \mathbb{S}^{C_k}$ since $E \subset F$, i.e.,

$$A_q = \sum_{k=1}^p A_q^k, \quad A_q^k \in \mathbb{S}^{C_k}, \quad k = 1, 2, \dots, p. \quad (9)$$

Using (9) we can write the inner product of A_q with an arbitrary matrix $X \in \mathbb{S}^N$ as

$$A_q \bullet X = \sum_{(i,j) \in F} [A_q]_{ij} X_{ij} = \sum_{k=1}^p \left(\sum_{(i,j) \in F} [A_q^k]_{ij} X_{ij} \right).$$

Since $A_q^k \in \mathbb{S}^{C_k}$ we may replace this equality with

$$A_q \bullet X = \sum_{k=1}^p \left(A_q^k \bullet X(C_k) \right) \quad \text{for every } X \in \mathbb{S}^N. \quad (10)$$

By using (10) we have

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^p \left(A_0^k \bullet X(C_k) \right) \\ & \text{subject to} && \sum_{k=1}^p \left(A_q^k \bullet X(C_k) \right) = b_q, \quad q = 1, 2, \dots, m \quad (11) \\ & && X(C_k) \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p. \end{aligned}$$

Unfortunately this is not a standard SDP as the submatrices in the semidefinite constraints $X(C_k) \succeq 0$ share elements and so are not independent. To convert this into a standard SDP Fukuda *et. al.* introduce new independent variables X^1, X^2, \dots, X^p where $X^k \in \mathbb{S}_+^{C_k}$ and new constraints that ensure equality between the overlapping elements [9]. The SDP problem then becomes

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^p \left(A_0^k \bullet X^k \right) \\ & \text{subject to} && \sum_{k=1}^p \left(A_q^k \bullet X^k \right) = b_q, \quad q = 1, 2, \dots, m \quad (12) \\ & && U_{ij} \bullet X^k - U_{ij} \bullet X^l = 0, \quad (i, j, k, l \in \Lambda) \\ & && X^k \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p \end{aligned}$$

where the constraint $U_{ij} \bullet X^k - U_{ij} \bullet X^l = 0$ enforces that the variable X_{ij} shared by maximal cliques k and l must be equal¹. Note that the single large semidefinite constraint in (5) has been replaced by multiple semidefinite constraints on smaller matrix variables and that the SDP is now in a block diagonal form. This is important because SDP solvers such as SeDuMi can exploit this block diagonal form to calculate the next iterate of the Newton step more efficiently.

A. Dual SDP

We now decompose the dual SDP using Theorem 4. Consider (4). Under our assumption on the aggregate sparsity pattern of the LMI the feasibility constraints of the dual SDP are

$$\sum_{i=1}^q y_q A_q + Z = A_0, \quad Z \in \mathbb{S}_+^N(E, 0). \quad (13)$$

Let $G_{ch} = (V, F)$ be a chordal extension of $G = (V, E)$, with set of maximal cliques $\mathcal{C} = \{C_1, C_2, \dots, C_p\}$. Using this chordal extension we can write (13) in the equivalent form

$$\sum_{q=1}^m y_q A_q + Z = A_0, \quad Z \in \mathbb{S}_+^N(F, 0). \quad (14)$$

Note that the constraint $Z_{ij} = 0$ if $(i, j) \in F \setminus E$ is implicit in these constraints, since $A(y) = A_0 - \sum_{q=1}^m y_q A_q \in \mathbb{S}^N(E, 0)$ for all $y \in \mathbb{R}^m$. Let A_q^k for $k = 1, 2, \dots, p$ and $q = 0, 1, 2, \dots, m$ be matrices that satisfy

$$\sum_{k=1}^p \left(A_0^k - \sum_{q=1}^m y_q A_q^k \right) = A_0 - \sum_{i=1}^q y_q A_q.$$

¹Technically, Λ consists of two disjoint trees since the Lyapunov SDP is block diagonal with two blocks.

The following proposition uses Theorem 4 to decompose the constraints of the dual SDP.

Proposition 1: The set of constraints (14) is feasible if and only if there exist matrices $Z^k \in \mathbb{S}^{C_k}$ and $L^k(\theta) \in \mathbb{S}^{C_k}$ for $k = 1, 2, \dots, p$ that satisfy

$$Z^k = A_0^k - \sum_{q=1}^m y_q A_q^k - L^k(\theta), \quad k = 1, 2, \dots, p$$

$$Z^k \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p$$

where $L^k(\theta)$ are as defined in Theorem 5.

Proof: Let $Z = \sum_{k=1}^p Z_k$ and note that $\sum_{k=1}^p L^k(\theta) = 0$ to give the first condition. The second condition then follows from Theorem 4,

$$Z \in \mathbb{S}_+^N(F, 0) \Leftrightarrow Z = \sum_{k=1}^p Z_k, \quad Z_k \in \mathbb{S}_+^{C_k}.$$

With these equivalent constraints the decomposition of the dual SDP is

$$\begin{aligned} & \text{maximize} && b^T y \\ & \text{subject to} && \sum_{q=1}^m y_q A_q^k + L^k(\theta) + Z^k = A_0^k, \quad k = 1, 2, \dots, p \\ & && Z^k \in \mathbb{S}_+^{C_k}, \quad k = 1, 2, \dots, p. \end{aligned} \quad (15)$$

Note that the SDPs (12) and (15) are Lagrange duals of one another (derivation given in the Appendix). For each equality constraint $U_{ij} \bullet X^k - U_{ij} \bullet X^l = 0$ in the primal problem there is an associated dual variable θ_{ijkl} . We note that this decomposition has been exploited to decompose problems in distributed robust stability analysis using IQCs [20].

For further information on the conversion process we refer the reader to papers [9], [10], [12]. Solvers that exploit chordal sparsity in general SDPs are available, in particular SMCP [11] and SDPA-C [21]. Also available is SparseCoLO which automates the reformulation of SDPs with chordal sparsity to facilitate solution using standard solvers [12].

V. STRUCTURED SYSTEMS

Throughout this section let $\dot{x}(t) = Ax(t)$ be the system that we wish to construct a Lyapunov function for. We consider five classes of A matrices that allow us to choose P so that P and Q have a chordal sparsity pattern. By finding chordal sparsity patterns we are then able to efficiently test for Lyapunov functions of this form using the chordal decomposition of the SDP.

A. Banded Matrices

We say that $A \in \mathbb{R}^{n \times n}$ is a banded matrix of bandwidth d if there exists an integer $1 < d < n$ such that

$$A_{ij} \neq 0 \text{ if and only if } |i - j| \leq d.$$

This form of A matrix corresponds to the case where the system consists of a chain of overlapping subsystems. Note that banded matrices are a subset of chordal graphs. In this case we can choose P to also be a banded matrix of order $d = 1, 2, \dots, n - 1$ and this gives a sequence of banded Q matrices of increasing order up to a complete graph.

B. Cyclical Matrices

Another class of A matrices for which we can find P and Q with chordal sparsity patterns is the class of cyclical matrices. In a cyclical matrix the nodes are connected in a ring or a cycle topology which can be represented by the following sparsity pattern

$$A_{ij} = \begin{cases} 1 & \text{if } |i - j| \leq 1 \\ 1 & \text{if } (i, j) = (1, n) \text{ or } (n, 1) \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Let the sparsity pattern of P be defined by

$$P_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } i + j = n + 1 \\ 1 & \text{if } i + j = n + 2 \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Theorem 6: Given $A \in \mathbb{R}^{n \times n}$ with a cyclic sparsity pattern, the sparsity pattern of P defined in (17) is chordal and results in a $Q = A^T P + PA$ with a chordal sparsity pattern. (Proof given in the appendix).

C. Tree Matrices

Let $G = (V, E)$ be a tree, i.e., a graph that is connected and has no cycles. We pick one node to be the root of the tree and label it r . We say that node u is the parent of node v and v is the child of u if u is the unique node adjacent to v on the unique path from r to v . We define the siblings of a node u to be the set of all nodes with the same parent as u (including u). Given node $i \in V$, we denote its parent by $\text{Par}(i)$, its children by $\text{Ch}(i)$ and its siblings by $\text{Sib}(i)$. We number the nodes in a topological ordering, meaning that each parent has a higher number than its children. Let $M(i) = \{i\} \cup \text{Par}(i) \cup \text{Ch}(i)$.

Given a tree T , define the sparsity pattern of A by

$$A_{ij} = \begin{cases} 1 & \text{if } j \in M(i) \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Let the sparsity pattern of P to be defined by

$$P_{ij} = \begin{cases} 1 & \text{if } j \in \text{Sib}(i) \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

Theorem 7: Given $A \in \mathbb{R}^{n \times n}$ with a sparsity pattern defined by the tree T , the sparsity pattern of P defined in (19) is chordal and results in a $Q = A^T P + PA$ with a chordal sparsity pattern. (Proof given in the appendix).

D. Metzler Matrices and Triangular Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be Metzler if all its off diagonal elements are nonnegative i.e., $a_{ij} \geq 0 \forall i \neq j$.

Proposition 2: [6] Let $A \in \mathbb{R}^{n \times n}$ be a Metzler matrix. Then the following statements are equivalent:

- i) the matrix A is Hurwitz
- ii) there exists a positive definite, diagonal matrix $P \succ 0$ such that $Q = A^T P + PA \prec 0$.

If we further assume that A is sparse, then by choosing P to be diagonal we ensure that the structure of A is preserved in Q . Let $G = (V, E)$ be the graphical structure of Q , then

we can find a chordal extension $G = (V, F)$ where $E \subseteq F$. Therefore, given a sparse Metzler matrix we can apply the chordal decomposition to the SDP so that we can efficiently test whether the system is stable.

Remark 1: It is known that stable triangular matrices also admit a diagonal Lyapunov function [22]. Therefore when a matrix is triangular and sparse we can apply exactly the same decomposition as with the Metzler matrices.

E. Sparse Non-chordal Matrices

Chordal extensions allow us to generalise the chordal decomposition approach to non-chordal sparse systems. Given a sparse matrix $A \in \mathbb{R}^{n \times n}$, choose a sparsity pattern for P , i.e., $G(P) = (V, R)$. This implies a sparsity pattern for Q i.e., $G(Q) = (V, S)$ where $Q = A^T P + PA$. The sparsity patterns of P and Q will not be chordal in general, but we can chordal extend them to $G_{ch}(P) = (V, T)$ and $G_{ch}(Q) = (V, U)$ and then apply the chordal decomposition as described in the previous section.

This suggests a method to iteratively search for sparse Lyapunov functions:

Algorithm: Given $A \in \mathbb{R}^{n \times n}$. Initialise $m = 1$
while: $m \leq n$
 $P =$ banded matrix with bandwidth m .
 Compute $G(Q)$ where $Q = A^T P + PA$
 Chordal extend $G(P)$ and $G(Q)$ to $G_{ch}(P)$ and $G_{ch}(Q)$
 Solve the chordal SDP (12) or (15).
 if: SDP is feasible
 A Lyapunov function has been found
 else:
 $m = m + 1$
 end
end

VI. NUMERICAL RESULTS

In this section we give numerical results that demonstrate the improvements in efficiency possible when chordal sparsity in an LMI is taken advantage of. We make use of SparseCoLO which detects chordal sparsity in an LMI and preprocesses the data so that the matrix variables are block diagonal. The conversion is performed using the clique tree conversion method described in Section IV. After this preprocessing step, SparseCoLO then calls either SeDuMi, SDPA or SDPT3 to solve the problem [12].

To test the chordal decomposition method we generated random matrices with negative eigenvalues and imposed on them the properties described in Section V to create sparse Metzler, sparse triangular, banded, cyclical and tree structured A matrices. Then we compared the time taken for SeDuMi and SparseCoLO+SeDuMi to solve the Lyapunov LMI. For all of our experiments the SDP was considered to be solved when the primal-dual gap had been reduced to less

than $\varepsilon = 10^{-9}$. The experiments were run on a MacBook Pro with a 2.9GHz processor and 8GB of RAM.

For the case of banded matrices we chose A and P to be banded with order $d = 5$. To generate the trees we simply started with the trivial tree $\{v\}$ and then added edges to a node not in the tree, repeating until all of the nodes were connected to the tree. For the Metzler and triangular cases we generated sparse graphs and then found a chordal extension to determine how to decompose the problem. To generate sparse non-chordal matrices we first generated sparse chordal graphs and then added in random elements using the MATLAB command `sprandsym(n, density)` where $density = 0.1/n$.

Table I shows the CPU time in seconds required to solve the SDPs for the five different sparsity patterns. The variable maxC corresponds to the largest maximal clique in the sparsity pattern of the LMI. MaxC was kept constant as the number of nodes was increased. The numbers in the brackets e.g., (230,17) are the size and number of blocks the Schur complement matrix respectively.

We see that the size of the Schur complement matrix is larger when we apply the chordal decomposition method. This is due to the extra variables that are introduced (the dual variables of the equality constraints (θ)). However, the problem is decomposed into a number of blocks of smaller size which SeDuMi can exploit to solve the problem more efficiently. Hence the chordal decomposition method is significantly more efficient in these sparse cases with small maximal cliques.

As the number of nonzero elements in the A matrix increases, this tends to increase the density of the matrices in the LMI. This leads to larger maximal cliques, with more overlapping elements, which in the chordal decomposition requires us to introduce more dual variables. This increases the size of the Schur complement matrix. Table II shows the effect of increasing maximal clique sizes on the time taken to solve the Lyapunov LMI for sparse Metzler matrices with $n = 400$. We see that for small maxC the chordal decomposition is more efficient than the standard dense method, but as maxC is made larger the increased size of the Schur complement matrix outweighs the advantages of decomposing the problem into blocks.

VII. CONCLUSION

We have shown that chordal sparsity in the Lyapunov LMI can be exploited to improve the efficiency with which the associated SDPs are solved. We applied the method to construct Lyapunov functions for randomly generated linear systems from five different classes. It was shown that when the size of the maximal cliques is small the chordal decomposition method is significantly quicker than the standard method. This makes it a promising approach when applied to large, sparse systems which are often found in applications.

In the future we will apply this methodology to other LMIs arising in systems and control theory, such as in

Banded Matrices (maxC = 11)		
n	SeDuMi	SparseCoLO+SeDuMi
200	10.7 (1185,2)	4.4 (1865,27)
400	65.3 (2385,2)	8.3 (3745,51)
600	197.9 (3585,2)	12.7 (5640,76)
800	679.4 (4785,2)	17.5 (7535,101)
Cyclical Matrices (maxC = 4)		
n	SeDuMi	SparseCoLO+SeDuMi
200	9.2 (399,2)	2.1 (489,33)
400	52.0 (799,2)	3.1 (986,65)
600	152.0 (1199,2)	5.2 (1476,95)
800	418.6 (1599,2)	7.1 (1973,127)
Tree Matrices (maxC = 15)		
n	SeDuMi	SparseCoLO+SeDuMi
200	13.6 (393,2)	3.3 (643,73)
400	88.3 (755,2)	7.7 (1063,146)
600	290.2 (1184,2)	9.9 (1684,217)
800	466.4 (1598,2)	13.9 (2502,289)
Sparse Metzler (maxC = 10)		
n	SeDuMi	SparseCoLO+SeDuMi
200	15.76 (200,2)	1.6 (230,17)
400	66.9 (400,2)	3.3 (496,37)
600	296.0 (600,2)	4.3 (732,53)
800	391.0 (800,2)	6.9 (982,71)
Sparse Triangular (maxC = 10)		
n	SeDuMi	SparseCoLO+SeDuMi
200	14.0 (200,2)	1.9 (291,24)
400	71.2 (400,2)	2.7 (533,41)
600	237.1 (600,2)	4.7 (931,57)
800	401.8 (800,2)	4.9 (995,70)
Sparse Non-chordal (maxC = 12)		
n	SeDuMi	SparseCoLO+SeDuMi
200	9.3 (200,2)	1.8 (280,31)
400	58.2 (400,2)	4.0 (680,73)
600	192.3 (600,2)	5.8 (986,97)
800	595.6 (800,2)	8.5 (1266,70)

TABLE I: SeDuMi CPU time vs SparseCoLO+SeDuMi CPU time in seconds to solve the Lyapunov LMI for different sparsity patterns, (size of Schur complement matrix, no. of blocks).

Sparse Metzler (n= 400)			
maxC	nnz(A)	SeDuMi	SparseCoLO+SeDuMi
6	2022	57.5 (400,2)	3.3 (486,35)
10	2606	65.1 (400,2)	3.4 (554,36)
17	4874	69.4 (400,2)	5.3 (980,44)
20	5240	72.0 (400,2)	14.9 (3418,192)
27	9260	70.0 (400,2)	129.9 (7182,172)

TABLE II: SeDuMi CPU time vs SparseCoLO+SeDuMi CPU time in seconds to solve the Lyapunov LMI for Metzler matrices with $n = 400$ nodes whilst varying the size of the maximal cliques, (size of Schur complement matrix, no. of blocks).

the KYP lemma. This will allow us to bound the input-output properties of large, sparse systems. Another promising direction is to apply this approach to the stability analysis of systems with polynomial vector fields by using Sum of Squares. We will also investigate how to use information about the system, in terms of dynamics or spectral properties, in order to inform the choice of sparsity pattern for the Lyapunov function

VIII. APPENDIX

A. Derivation of the dual SDP from the primal SDP

The Lagrangian function for (12) is

$$L(X, y, \theta, Z) = \sum_{k=1}^p (A_0^k \bullet X^k) - \sum_{q=1}^m y_q \left(\sum_{k=1}^p (A_q^k \bullet X^k) - b_q \right) - \sum_{i,j,k,l \in \Lambda} \theta_{ijkl} (U_{ij} \bullet X^k - U_{ij} \bullet X^l) - \sum_{k=1}^p (Z^k \bullet X^k)$$

where $Z^k \in \mathbb{S}_+^k$ for $k = 1, 2, \dots, p$. The dual function is defined by $g(y, \theta, Z) = \inf_X L(X, y, \theta, Z)$. Define $L^k(\theta)$ to be

$$L^k(\theta) = \sum_{(i,j,l):(i,j,k,l) \in \Lambda} U_{ij} \theta_{ijkl} - \sum_{(i,j,h):(i,j,h,k) \in \Lambda} U_{ij} \theta_{ijhk}$$

Taking the partial derivative of the Lagrangian function with respect to X^k we have

$$\frac{\partial L}{\partial X^k} = A_0^k - \sum_{q=1}^m y_q A_q^k - L^k(\theta) - Z^k = 0, \quad (20)$$

and from this we see that the dual function is given by

$$g(y, \theta, Z) = \begin{cases} b^T y & \text{if (20) holds and } Z^k \in \mathbb{S}_+^k \quad \forall k, \\ -\infty & \text{otherwise.} \end{cases}$$

Maximising this dual function gives (15).

B. Proof of Theorem 6

Let A and P have sparsity patterns as defined in (16) and (17). Note that P can be rearranged to be a banded matrix and hence have a chordal sparsity pattern. The sparsity pattern of $Q = A^T P + P A$ is given by

$$Q_{ij} = \begin{cases} 1 & \text{if } |i-j| \leq 1 \\ 1 & \text{if } n \leq i+j \leq n+3 \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

The proof that Q as defined above is chordal is by induction. We use the notation that Q^n is the Q matrix for an n node cycle. For $n = 1, 2, 3$ there cannot be a cycle of length greater or equal to 4 so these graphs must be chordal. We first consider the even case and take Q^4 as our base case. Since it is a complete graph it is chordal. Next consider Q^n with n even and assume that it is chordal. Construct Q^{n+2} by relabelling the nodes of Q^n by incrementing $v_i := v_{i+1}$, for $i = 1, 2, \dots, n$ and then adding two new nodes $\{v_1, v_{n+2}\}$ to the periphery of the network and connecting them according to (21), see Figure 2.

Let the perfect elimination ordering for Q^n be denoted by α . If we can find a perfect elimination ordering for Q^{n+2} of the form $\langle v_1, v_{n+2}, \alpha \rangle$ then we have shown that Q^{n+2} is chordal. Consider the first row of Q^{n+2}

$$Q_{1,k}^{n+2} = 1, \text{ for } k \in \{1, 2, n+1, n+2\}, 0 \text{ otherwise.}$$

In order for node 1 to be simplicial, nodes $2, n+1$ and $n+2$ must form a clique. From (21) we see that $Q_{ij}^{n+2} = 1$ for $n+2 \leq i+j \leq n+5$ and hence these nodes do indeed form

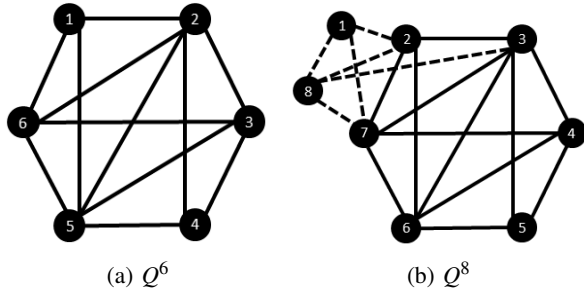


Fig. 2: Constructing Q^8 from Q^6 (Drawn without self-loops)

a clique. Similarly for node $n+2$, after eliminating node 1 the $n+2$ th row of Q^{n+2} is

$$Q_{n+2,k}^{n+2} = 1, \text{ for } k \in \{2, 3, n+1, n+2\}, 0 \text{ otherwise.}$$

Again using the definition of Q^{n+2} we see that the nodes 2, 3 and $n+2$ form a clique and hence $n+2$ is a simplicial vertex. Hence the ordering $\langle v_1, v_{n+2}, \alpha \rangle$ is a perfect elimination ordering, and by induction Q^n is chordal for all even n . The same argument can be applied to the odd n case, which completes the proof.

C. Proof of Theorem 7

Let A and P have sparsity patterns as defined in (18) and (19). Note that P can be rearranged to be block diagonal and hence have a chordal sparsity pattern. The element-wise equation for the sparsity pattern of Q is

$$Q_{ij} \neq 0 \Leftrightarrow \sum_{k=1}^n A_{ik}P_{kj} + \sum_{k=1}^n P_{ik}A_{kj} \neq 0$$

where we have used the fact that the sparsity pattern of A is symmetric. From the definitions of the sparsity pattern of A and P we conclude that the sparsity pattern of Q is given by

$$Q_{ij} = \begin{cases} 1 & \text{if } (M(i) \cap \text{Sib}(j) \neq \emptyset) \text{ or } (\text{Sib}(i) \cap M(j) \neq \emptyset) \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

Let the topological ordering $\alpha = \{1, 2, \dots, n\}$ be a candidate for a perfect elimination ordering. The subgraph induced by $\{v_i, \dots, v_n\}$ cannot include the children of v_i or the children of its siblings, since these nodes have a lower number in the topological ordering. The remaining nodes that v_i is connected to are: its siblings with a higher number in the ordering, its parent and the siblings of its parent. It remains to be shown that all of the siblings of i are adjacent to all of the siblings of the parent of node i . Let $j \in \text{Sib}(i)$ and $k \in \text{Sib}(\text{Par}(i))$, using (22) we have

$$M(j) \cap \text{Sib}(k) = \text{Par}(i) \neq \emptyset \Rightarrow Q_{jk} = 1, Q_{kj} = 1.$$

Hence the siblings of the parent of node i are adjacent to all of the siblings of i . Therefore v_i is a simplicial vertex and α is a perfect elimination ordering. We conclude that Q is a chordal graph.

REFERENCES

- [1] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.
- [2] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*. DISC course lecture notes, 2004.
- [3] S. J. Hammarling, "Numerical solution of the stable, non-negative definite Lyapunov equation," *IMA Journal of Numerical Analysis*, vol. 2, pp. 303–323, 1982.
- [4] S. Boyd and Q. Yang, "Structured and simultaneous Lyapunov functions for system stability problems," *International Journal of Control*, vol. 49, pp. 2215–2240, 1989.
- [5] M. Arcak, "Diagonal stability on cactus graphs and application to network stability analysis," *IEEE Transactions of Automatic Control*, vol. 56, pp. 2766–2777, 2011.
- [6] A. Rantzer, "Distributed control of positive systems," *50th IEEE Conference on Decision and Control and European Control Conference*, pp. 6608–6611, 2011.
- [7] 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.
- [8] J. Agler, J. W. Helton, and S. McCullough, "Positive semidefinite matrices with a given sparsity pattern," *Linear Algebra and its Applications*, vol. 107, pp. 101–149, 1988.
- [9] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion I: general framework," *Society for Industrial and Applied Mathematics*, vol. 11, pp. 647–674, 2000.
- [10] K. Nakata, K. Fujitsawa, M. Fukuda, M. Kojima, and K. Murota, "Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results," *Mathematical Programming Series B*, vol. 95, pp. 303–327, 2003.
- [11] M. S. Andersen, J. Dahl, and L. Vandenberghe, "Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones," *Mathematical Programming Computation*, vol. 2, pp. 167–201, 2010.
- [12] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita, "Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion," *Mathematical Programming Series B*, vol. 129, pp. 33–68, 2011.
- [13] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Publications, 1994.
- [14] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results," *SIAM J. Optimization*, vol. 8, pp. 746–768, 1998.
- [15] E. D. Klerk, C. Roos, and T. Terlaky, "Initialization in semidefinite programming via a self-dual skew-symmetric embedding," *Operations Research Letters*, vol. 20, pp. 213–221, 1997.
- [16] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, available at <http://sedumi.ie.lehigh.edu/>.
- [17] D. R. Fulkerson and O. A. Gross, "Incidence matrices and interval graphs," *Pacific Journal of Mathematics*, vol. 15, pp. 835–855, 1965.
- [18] J. R. S. Blair and B. Peyton, "An introduction to chordal graphs and clique trees," in *Graph Theory and Sparse Matrix Computation*, A. George, J. R. Gilbert, and J. W. Liu, Eds. New York: Springer, 1993.
- [19] N. Kakimura, "A direct proof for the matrix decomposition of chordal-structured positive semidefinite matrices," *Linear Algebra and its Applications*, vol. 433, pp. 819–823, 2010.
- [20] M. S. Andersen, A. Hansson, S. K. Pakazad, and A. Rantzer, "Distributed robust stability analysis of interconnected uncertain systems," *Proceedings of the 51st IEEE Conference on Decision and Control (CDC), Hawaii*, vol. 1, pp. 1548–1553, 2012.
- [21] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, and M. Yamashita, "SDPA-C (semidefinite programming algorithm – completion method) user's manual — version 6.10," *Research Report B-409, Dept. Math. and Comp. Sciences, Tokyo Institute of Technology*, 2004.
- [22] G. Barker, A. Berman, and R. Plemmons, "Positive diagonal solutions to the Lyapunov equations," *Linear and Multilinear Algebra*, vol. 5, pp. 249–256, 1978.