Compositional Verification of Large-Scale Nonlinear Systems via Sums-of-Squares Optimization

Shen Shen and Russ Tedrake

Abstract— We consider the computationally prohibitive problem of stability and invariance verification of large-scale dynamical systems. We exploit the natural interconnected structure often arising from such systems in practice (i.e., they are interconnections of low-dimensional subsystems), and propose a compositional method. We construct *independently* for each subsystem a Lyapunov-like function, and guarantee that their sum *automatically* certifies the original high-dimensional system is stable or invariant. For linear time invariant systems, our method produces block-diagonal Lyapunov matrices without structural assumptions commonly found in the literature. For polynomial system tasks, our formulation results in significantly smaller sum-of-squares programs. Demonstrated on numerical and practical examples, our algorithms can handle problems beyond the reach of direct optimizations, and are orders of magnitude faster than existing compositional methods.

I. INTRODUCTION

Linear matrix inequalities (LMIs) are ubiquitous in system analysis, largely due to its clean connection to Lyapunov theory. It is widely known that most of the common linear time invariant (LTI) systems analysis and synthesis tasks directly translate through Lyapunov argument into LMIs [7]. More recently, the development of sums-of-squares (SOS) programming makes it possible to essentially apply this technique on polynomial systems too [12], [13], expanding the applicability even further.

Practically, however, LMIs and by extension SOS do not scale very well, and the computational cost is immense for large scale systems. This computational challenge, combined with the natural decomposition structure often arising from these systems, motivates research areas such as compositional analysis [2], [20] and distributed and decentralized control [14], [4]. The common theme there is to not investigate the high dimensional system directly. Instead, the large system is first divided and studied in parts, and the implication of these individual results would then be reasoned about collectively for the original system.

In the context of stability analysis for LTI systems, the compositional idea is typically materialized as the search of block-diagonal Lyapunov matrices. Various necessary and sufficient conditions on their existence have been given in the literature, but they are either restrictive or non-constructive. The restriction is usually on the dynamics matrix A having special structure such as being Metzler matrix [11], block triangular or cyclic [3], or on the Lyapunov matrix having

particular block-diagonal patterns like strictly diagonal [6], [15], [21], [18] or being limited to a 2-by-2 partition [17]. For the more general conditions such as in [8], there is no simple recipe computing the desired Lyapunov matrix from the sufficient rank conditions given. Our work offers constructive sufficient conditions without any of these structural limitations. (It is worth mentioning that we noticed Theorem 1 in this part is very similar to an independent and recent work [22].)

For compositional analysis of the more general polynomial systems via SOS framework, we mention the work of [20], [16], [1], which are most similar to ours. In previous work, the Lyapunov value constraints are untangled but the time derivatives are not. Our algorithm completely decouples both, and the resulting additional computational saving can come in multiple orders of magnitude. We also note in particular that [1] focuses more on revealing a latent modular structure using graph partition ideas; we on the other hand assume the system has a given decomposition structure or one obvious enough by visual inspection and strive purely for efficiency. Our views are thus complementary, and combining them can solve a larger class of problem faster, as will be shown later with an example.

Next, in Section II, we formalize the problem and introduce technical background. In Section III, we focus on finding block-diagonal Lyapunov matrix for LTI system, and present two algorithms for dynamics matrices of arbitrary size, structure, and partition pattern. In Section IV, we address the extension to polynomial dynamics and formulate a much smaller SOS programming. Finally, we demonstrate on numerical and practical examples in Section V the efficiency of the proposed algorithms.

A. Notation

For a real vector $x \in \mathbb{R}^n$, the usual Euclidean 2-norm is denoted as $\|x\|$, the weighted 2-norm is denoted as $\|x\|_A^2 := x'Ax$ with $A \in \mathbb{R}^{n \times n}$, and the time derivatives are denoted as $\dot{x}$. If $x_i \in \mathbb{R}^{n_i}$, $i = 1, 2, \ldots, m$, then $(x_1, x_2, \ldots, x_m)$ denotes their column catenation. For a matrix $A \in \mathbb{R}^{m \times n}$, $\sigma_1(A)$ is its largest singular value and A' its transpose. $A \succ 0$ (resp. $A \succeq 0$) implies A is square, symmetric, and positive definite (resp. positive semidefinite). If $A_i \in \mathbb{R}^{k_i \times k_i}$, $i = 1, 2, \ldots, m$, then $\bigoplus_{i=1}^{m} A_i := A_1 \oplus A_2 \cdots \oplus A_m$ denotes the block diagonal matrix with diagonal blocks $A_1, A_2, \ldots, A_m$. I denotes the identity matrix of appropriate size. Symbol $\setminus$ denotes set complement. $\mathbb{R}[x]$ denotes the ring of scalar polynomial functions in indeterminate x with

real coefficients, and $\mathbb{R}[x]^{m \times n}$ denotes an m by n matrix whose elements are scalar polynomials in $\mathbb{R}[x]$.

II. PROBLEM STATEMENT AND BACKGROUND

A. Problem Formulation

Consider a time-invariant polynomial system described by $\dot{x} = f(x)$ where the state $x \in \mathbb{R}^n$ and the dynamics $f \in \mathbb{R}[x]^n$. We restrict ourselves to time-invariant systems and will drop all time dependencies. Let the state x be partitioned into m components: $x = (x_1, x_2, \ldots x_m)$, where $x_i \in \mathbb{R}^{n_i}$ constitutes the states of a subsystem. We assume the partition is one such that no more than two subsystems are coupled, i.e., no terms like $x_{11}x_{22}x_{32}$ (x_{11} being the first state in the first subsystem and so on) exist in f. This is not a restrictive assumption as it can always be satisfied by regrouping (e.g., one can merge x_1 and x_2 into a new sub-system should terms like $x_{11}x_{22}x_{32}$ appear).

With the partition and assumption above, $\dot{x} = f(x)$ can be rearranged into a component-wise expanded form:

$$\dot{x}_i = f_i(x_i) + \sum_{\substack{j=1 \\ j \neq i}}^{m} g_{ij}(x_i)h_{ij}(x_j) \tag{1}$$

where $f_i \in \mathbb{R}[x_i]^{n_i}$ describes the internal dynamics of sub-state x_i, $g_{ij} \in \mathbb{R}[x_i]^{n_i \times l_{ij}n_j}$ and $h_{ij} \in \mathbb{R}[x_j]^{l_{ij}n_j}$ captures the coupling between sub-state x_i and x_j. The newly introduced dimension l_{ij} is due to the possibility of more than one linearly independent coupling terms involving x_i and x_j, for instance, say $\dot{x}_1 = -x_1{}^3 + x_1 x_2 + 3x_1{}^2 x_2{}^2$, then $l_{12} = 2$. (For the special LTI case, $l_{ij} = 1$.)

We are interested in making claims such as asymptotic stability to the origin and invariance for the entire system states x, but ideally by examining one sub-system state x_i at a time. To this end, we associate the system with a Lyapunov-like function V such that:

$$V(x) = \sum_{i=1}^{m} V_i(x_i) \geq 0, \forall x \in \mathbb{R}^n \tag{2a}$$

$$\dot{V}(x) = \sum_{i=1}^{m} \dot{V}_i < 0, \forall x \in \mathbb{D} \tag{2b}$$

where the equality in (2a) holds only at the origin, and the region $\mathbb{D}$ in (2b) varies with the task in hand. For instance, when dealing with global stability to the origin, $\mathbb{D} = \mathbb{R}^n \backslash \{0\}$, whereas in local analysis the region is usually a sub-level set of V and part of the decision variables.

The aim of this paper is to find the set of $\{V_i\}_{i=1}^{m}$ functions *independently* so as to form as small an LMI or SOS as possible. (2a) is already in a decoupled form, and one can simply require $V_i \geq 0, \forall i$. (2b) may look decoupled too, and one may be tempted to claim that $\dot{V}_i < 0, \forall i$ is also a set of independent constraints; this is not true. Note that

$$\dot{V}_i = \frac{\partial V_i(x_i)}{\partial x_i} f_i + \sum_{\substack{j=1 \\ j \neq i}}^{m} \frac{\partial V_i(x_i)}{\partial x_i} g_{ij}(x_i)h_{ij}(x_j) \tag{3}$$

while the first term is only dependent on x_i, the second term that is the summation involves h_{ij}, a function of sub-states x_j, and the summing over all $j \neq i$ makes $\dot{V}_i$ dependent on possibly the entire states $x = (x_1, x_2, \ldots x_m)$. This is a direct consequence of sub-states coupling from the dynamics $\dot{x}_i$, in other words, the set of $\{\dot{V}_i\}_{i=1}^{m}$ are inherently entangled.

Our compositional approach thus avoids dealing with $\{\dot{V}_i\}_{i=1}^{m}$ head-on. Instead, we resort to finding an upper bound of $\dot{V}$ that is by design a sum of functions each dependent on one x_i only. We then require this upper bound to be non-positive to sufficiently imply (2b). While this detour leads to more conservative results, it allows the parallel search we desire and can bypass the computational hurdle of direct optimizations. The details of our approach are in Section III and IV.

B. Sums-of-Squares Programming

Direct application of Lyapunov theory on polynomial system requires checking non-negativity of polynomials, which is unfortunately NP-hard in general. However, the problem of checking if a polynomial is sum-of-squares (SOS), which is sufficient for non-negativity, is computationally approachable. A scalar multivariate polynomial $F(x) \in \mathbb{R}[x]$ is called SOS if it can be written as $F(x) = \sum_{i=1}^{m} f_i^2(x)$ for a set of polynomials $\{f_i\}_{i=1}^{m}$. If $\deg(F) = 2n$, this SOS condition is equivalent to $F(x) = m'(x)Qm(x)$ where $m(x)$ is a vector whose rows are monomials of degree up to n in x, and the constant matrix called the Gram matrix $Q \succeq 0$. Thus, the search of a SOS decomposition for F can be equivalently cast as an LMI on Q [12]. We will replace all sign constraints on polynomials in our optimization with SOS constraints out of computational consideration.

III. LTI SYSTEM AND BLOCK-DIAGONAL LYAPUNOV MATRIX

We first study the most fundamental LTI systems. Though the technical result in this section can be reduced from the polynomial systems' result, some more intuitive aspects of it can only be or are better appreciated in this limited setting and hence it merits the separate elaboration here.

Under the LTI assumption, (1) takes a clean form

$$\dot{x}_i = A_{ii}x_i + \sum_{\substack{j=1 \\ j \neq i}}^{m} A_{ij}x_j \tag{4}$$

where $A_{ii} \in \mathbb{R}^{n_i \times n_i}$, $A_{ii}x_i$ corresponds to the f_i term, $A_{ij} \in \mathbb{R}^{n_i \times n_j}$ corresponds to g_{ij} with dimension $l_{ij} \equiv 1$, and x_j corresponds to the h_{ij} term.

For LTI systems, it only makes sense to consider global asymptotic stability (to the origin) as all convergences in LTI systems are in the global sense. A quadratic parameterization of Lyapunov function $V = x'Px$ such that $P \succ 0$ and $AP + PA' \prec 0$ is both necessary and sufficient for this task. Naturally then, when considering Lyapunov functions for the subsystems, we use this quadratic parameterization as well

and let $V_i = x_i' P_i x_i$. This is equivalent to imposing a block-diagonal structure constraint on $P \succ 0$ as $P = \bigoplus_{i=1}^m P_i \succ 0$. Substituting the parameterization into condition (2) yields:

find $\{P_i\}_{i=1}^m$ (5a)

s.t. $P_i \succ 0, \forall i$ (5b)

$$\begin{bmatrix} A_{11}P_1 + P_1 A_{11}' & \cdots & A_{1n}P_n + P_1 A_{n1}' \\ A_{21}P_1 + P_2 A_{12}' & \cdots & A_{2n}P_n + P_2 A_{n2}' \\ \vdots & \ddots & \vdots \\ A_{n1}P_1 + P_n A_{1n}' & \cdots & A_{nn}P_n + P_n A_{nn}' \end{bmatrix} \prec 0,$$

(5c)

Note that (5c) gives a more tangible sense of the latent intertwined nature of (2b). The clean block-diagonal structure in P is deeply buried here as the left hand side is a full matrix with $A_{ij}P_j + P_i A_{ji}'$ at all the off-diagonal spots.

Our goal in this section is to find the set of $\{P_i\}_{i=1}^m$ independently for each i. We start with an LMI-based algorithm that is still somewhat coupled, and then gradually get to the truly decoupled algorithm which is based on Riccati equations.

A. Sparse LMIs

Theorem 1: For an n-dimensional LTI system written in the form (4), if the optimization problem

find $\{P_i\}_{i=1}^m, \{M_{ij}\}_{i,j=1, i \neq j}^m$ (6a)

s.t. $P_i \succ 0, \forall i$ (6b)

$M_{ij} \succ 0, \forall i, j, i \neq j$ (6c)

$$A_{ii}P_i + P_i A_{ii}' + \sum_{\substack{j=1 \\ j \neq i}}^m A_{ij}M_{ij}A_{ij}' + P_i M_{ji}^{-1} P_i \prec 0, \forall i$$

(6d)

is feasible, then the set of $\{P_i\}_{i=1}^m$ satisfies problem (5) with $\mathbb{D} = \mathbb{R}^n \backslash \{0\}$, and the original system is strictly asymptotically stable.

Proof: Two proofs, one from the primal perspective and the other the dual, are included in the appendix. Theorem 1 can also be reduced from Theorem 2 (in Subsection IV-A), whose proof is in fact less involved. The appended proofs, however, offer a control and optimization connection that the simple proof lacks. ∎

Remark 1: Theorem 1 can be viewed as a generalization of the sufficient direction of Lyapunov inequality for LTI systems. Particularly, if the state is not partitioned, P has no structural constraint, the summation in (6d) disappears, and the condition reduces to the ordinary Lyapunov inequality. Furthermore, Theorem 1 gives an explicit procedure to construct a Lyapunov matrix with user specified structures including the extreme case of pure diagonal structure.

Remark 2: Theorem 1 also closely resembles the small-gain theorem (which in fact is the inspiration for our results). Notice that setting $m = 2$ (two subsystems scenario) reduces Theorem 1 to the matrix version of the bounded real lemma, which proves the product of the two subsystems' ℓ_2-gains less than or equal to one and implies stability. The intuition

behind the connection is this: think of diagonal blocks A_{11} and A_{22} as describing two disconnected "nominal" plants, and the off-diagonal blocks are pumping feedback disturbance from one nominal system to the other. The compound system admitting a block-diagonal Lyapunov matrix indicates it is stable whether or not the disturbance blocks are present, and this is exactly what small-gain theorem implies. The feedback disturbance interpretation obviously carries over to more than two interconnected systems even though there is no extension of small-gain theorem in those settings. (The structured disturbance setup arising from μ-synthesis is not such a generalization. While it does admit multi-dimensional disturbances, all the disturbances are to the central nominal plant but not to one another.)

Remark 3: One might wonder what is the virtue of studying the bare bone Lyapunov inequality; after all, checking stability can easily be done through an eigenvalue computation and that is uniformly faster than LMIs. We believe the value lies in that LMIs is more general and clean than eigen-based methods. For instance, LMI formulation leads to extensions such as robustness analysis via common Lyapunov functions which eigen-based method fails to handle; or to a straightforward formulation of ℓ_2-gain bound, for which the eigen-based method leads to very messy computation. Therefore, $AP + PA'$, the most basic building block appearing in almost every control LMI, deserves a close examination.

Remark 4: Computationally, (6d) with the nonlinear term $P_i M_{ji}^{-1} P_i$ can be equivalently turned into an LMI via Schur complement. Hence, Theorem 1 requires solving m coupled LMIs of the original problem size, but all of them enjoy strong sparsity as all non-zero terms only appear, in the block sense, at one row, one corresponding column, and the main diagonal. For instance, when $m = 8$, Figure. 1 shows the sparsity pattern of three out of the eight LMIs. The sparsity in practice might already be a worthy trade off, and we test the claim in Section V. Further, if we fix the set of M_{ij} rather than searching for them, constraints (6) become decoupled low-dimensional LMIs. Better yet, they can be solved by an even faster Riccati equation based method below.
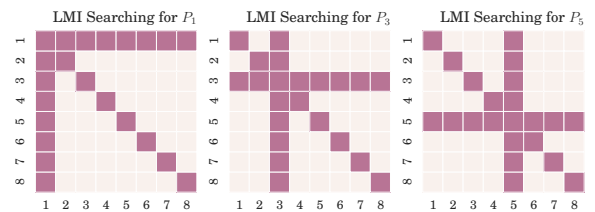

Fig. 1.   Sparsity Pattern of Three LMIs for $m = 8$ Example

B. Riccati equations

If M_{ij} are fixed, the set of decoupled low-dimensional LMIs can be solved by a method based on Riccati equations, which has near-analytical solutions and by implication far better scalability and numerical stability than LMIs.

Specifically, if we replace the inequality with equality, then (6) are precisely Riccati equations with unknown P_i. The feasibility of LMIs like (6) is equivalent to the feasibility of the associated Riccati equations. That is, the (unique) positive definite solution to the Riccati equation lives on the boundary of the feasible set of the LMI [7]. So by nudging the right hand size in the constraint slightly in the positive direction, e.g., replacing zero with ϵI for some small $\epsilon > 0$, we get a solution strictly in the interior and one precise to the original LMI. We note that when the Riccati equations return a feasible solution, it is much faster than solving the sparse LMIs (6), and even more significantly so than the original LMI (5).

The choice of the set of positive definite M_{ij} scaling matrices can be arbitrary but would largely affect the feasibility. Identity scaling is one obviously valid choice, and it is very likely to succeed in cases such as when the off-diagonal blocks are very close to zeros. In general though, identity scaling is not guaranteed to always work, it is then desirable to have some other heuristics at our disposal. Inspired by the small-gain theorem connection in Remark 2, we propose another educated guess that we call σ_1-scaling. The procedure is to (1) initialize a set of scalars $\gamma_{ij} = \sigma_1(A_{ii}^{-1}A_{ij})$, (2) keep γ_{ij} as is if $\gamma_{ij}\gamma_{ji} \leq 1$, otherwise, say $\gamma_{ij} > \gamma_{ji}^{-1}$, then keep only γ_{ji} and shrink γ_{ij} down to γ_{ji}^{-1}, and (3) set $M_{ij} = \gamma_{ij}I$. The justification of the heuristic is that σ_1 operator of a dynamics matrix loosely reflects the input-output signal magnification by the system, and $\sigma_1(A_{ii}^{-1}A_{ij})$ can therefore serve as a barometer of the relative energy exchange between an internal A_{ii} subsystem and the coupling A_{ij} term from system j. Of course, there is no guarantee on the performance of σ_1-scaling either. Empirically though, they succeed roughly 7 times out of 10. Plus, the time it takes to test these scalings is negligible compared with solving any LMIs, so it is well worth a try.

IV. POLYNOMIAL SYSTEMS

Extending the LTI decoupling idea to polynomial systems is conceptually straightforward: we again want to upper bound $\dot{V}$. Technically, a few nice properties from the LTI case would vanish. We will address these issues when they appear, and for now start with the result for global asymptotic stability.

A. Global Asymptotic Stability

Theorem 2: For a polynomial system described in the expanded form (1), if the optimization:

$$\text{find} \quad \{V_i\}_{i=1}^m, \{M_{ij}\}_{i,j=1,i\neq j}^m \tag{7a}$$

$$\text{s.t.} \quad M_{ij} \succ 0, \forall i, j, i \neq j \tag{7b}$$

$$V_i - \epsilon\|x_i\| \quad \text{is SOS}, \forall i \tag{7c}$$

$$-\frac{\partial V_i}{\partial x_i}f_i - \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^m \|\frac{\partial V_i}{\partial x_i}g_{ij}\|_{M_{ij}}^2 - \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^m \|h_{ji}\|_{M_{ji}^{-1}}^2$$
$$- \epsilon\|x_i\| \quad \text{is SOS}, \forall i \tag{7d}$$

is feasible for some small $\epsilon > 0$, then the set of polynomial functions $\{V_i(x_i)\}_{i=1}^m$ satisfies (2) with $\mathbb{D} = \mathbb{R}^n\backslash\{0\}$ and the system is globally asymptotically stable to the origin.

Proof: (7c) obviously implies (2a). Then by introducing invertible matrices $m_{ij} \in \mathbb{R}^{l_{ij}nj \times l_{ij}nj}$ and let $M_{ij} = m_{ij}m_{ij}'$, we can have $\dot{V}$ upper bounded as:

$$\dot{V} = \sum_{i=1}^m \dot{V}_i \tag{8a}$$

$$= \sum_{i=1}^m \left(\frac{\partial V_i}{\partial x_i}f_i + \sum_{\substack{j=1 \\ j\neq i}}^m \frac{\partial V_i}{\partial x_i}g_{ij}h_{ij} \right) \tag{8b}$$

$$= \sum_{i=1}^m \frac{\partial V_i}{\partial x_i}f_i + \sum_{i=1}^m \sum_{\substack{j=1 \\ j\neq i}}^m \left(\frac{\partial V_i}{\partial x_i}g_{ij}m_{ij}m_{ij}^{-1}h_{ij} \right) \tag{8c}$$

$$\leq \sum_{i=1}^m \frac{\partial V_i}{\partial x_i}f_i + \frac{1}{2}\sum_{i=1}^m\sum_{\substack{j=1 \\ j\neq i}}^m \left(\|g_{ij}\frac{\partial V_i}{\partial x_i}\|_{M_{ij}}^2 + \|h_{ij}\|_{M_{ij}^{-1}}^2 \right) \tag{8d}$$

$$= \sum_{i=1}^m \left(\frac{\partial V_i}{\partial x_i}f_i + \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i}}^m \left(\|\frac{\partial V_i}{\partial x_i}g_{ij}\|_{M_{ij}}^2 + \|h_{ji}\|_{M_{ji}^{-1}}^2 \right) \right) \tag{8e}$$

(8d) is due to the elementary inequality of arithmetic and geometric means (AM-GM inequality), and the exchange of summation index at (8e) is due to the symmetry between i and j. (7d) implies the negation of (8e) is SOS, which directly leads to that the negation of $\dot{V}$ is SOS, and sufficient to imply (2b). ∎

Remark 5: Our method can be extended to handle coupling terms such as $x_ix_jx_k$ that involves more than two sub-states. The key step is to use the generalized version of AM-GM inequality with $n > 2$ variables at (8d).

Remark 6: Similar to the LTI case, the scaling matrices M_{ij} brings coupling across the constraints. Eliminating these constants as decision variables could again untangle the entire set of constraints. However, we do not believe a trivial extension of σ_1-scaling developed for the LTI case would be as convincing a heuristic for hand-picking these constants in the polynomial settings. This is mainly due to the lack of a notion of 'coupling strength' in the polynomial sense. Specifically, for LTI systems, the coupling can only enter as $A_{ij}x_ix_j$, so at least intuitively, for a 'normalized' A_{ii} the strength of the coupling is quantified by A_{ij}. Polynomial systems with the additional freedom of degrees, however, can have coupling terms like $4x_ix_j$ and $x_ix_j^2$. It is then hard to argue, even hand-wavingly, if the coefficients play a more important role or if the degrees do. Therefore, we settle with just fixing all the M_{ij} to identity.

Remark 7: Even with $M_{ij} = I$, the term $\|\frac{\partial V_i}{\partial x_i}g_{ij}\|^2$ in (7d) is still not directly valid for a SOS program because of the quadratic dependency on V_i. We develop below what can be considered the generalization of Schur complement

in the polynomial settings to legalize the constraint.

Lemma 3: Given a scalar SOS polynomial $q(x) \in \mathbb{R}[x]$ of degree $2d_q$ and a vector of generic polynomials $s(x) \in \mathbb{R}[x]^n$ of maximum degree d_s, let y be a vector of indeterminates whose elements are independent of x, then $q(x) - s'(x)s(x) \in \mathbb{R}[x]$ is SOS if and only if $q(x) + 2y's(x) + y'y \in \mathbb{R}[x,y]$ is SOS.

Proof: Define $m(x)$ and $n(x,y)$ respectively as the standard monomial basis of x and (x,y) up to degree $d = \max(d_q, d_s)$. It is always possible to rewrite $s(x) = C'm(x)$ for some coefficients C, and $q(x) = m'(x)[Q + L(\alpha_1)]m(x)$, where Q is a constant symmetric matrix such that $q(x) = m'(x)Qm(x)$, $L(\alpha_1)$ is a parameterization of the linear subspace $\mathcal{L} := \{L = L' : m'(x)L(\alpha)m(x) = 0\}$, and $Q + L(\alpha_1) \succeq 0$. Denote $q(x) - s'(x)s(x)$ as Π_1 and plug in these parameterizations, $\Pi_1 = m'(x)[Q + L(\alpha_1) - C'C]m(x)$.

If Π_1 is SOS, then there exists an $L(\alpha_2) \in \mathcal{L}$ (possibly different from $L(\alpha_1)$) such that $Q + L(\alpha_1) - C'C + L(\alpha_2) \succeq 0$. This implies via Schur complement that $V := \begin{bmatrix} Q+L(\alpha_1)+L(\alpha_2) & C' \\ C & I \end{bmatrix} \succeq 0$. Denote $q(x) + 2y's(x) + y'y$ as Π_2 and notice that it is precisely $(m(x),y)'V(m(x),y)$, and therefore Π_2 is SOS.

If Π_2 is SOS, then there exists a β_1 such that $\Pi_2 = n'(x,y)[T + M(\beta_1)]n(x,y)$ where T is a constant symmetric matrix such that $n'(x,y)Tn(x,y) = \Pi_2$, $M(\beta_1)$ is a parameterization of the linear subspace $\mathcal{M} := \{M = M' : n'(x,y)M(\beta)n(x,y) = 0\}$, and $T + M(\beta_1) \succeq 0$. Since the elements of $m(x)$ and y form a strict subset of those in $n(x,y)$, the ordering $n(x,y) = (m(x),y,k(x,y))$ where k encapsulates the x,y cross term monomials is possible. Accordingly, $T + M(\beta_1)$ can be partitioned as $\begin{bmatrix} T_{11}+M_{11} & T_{12}+M_{12} & T_{13}+M_{13} \\ * & T_{22}+M_{22} & T_{23}+M_{23} \\ * & * & T_{33}+M_{33} \end{bmatrix}$ (β_1 from now on dropped for concision). Then, since Π_2 has no cross terms of second or higher order in y, it must be that $k'(x,y)[T_{33} + M_{33}]k'(x,y) = 0$, and since y and x are independent, $T_{33} + M_{33} = 0$. Similar arguments imply that $T_{23} + M_{23} = T'_{32} + M'_{32} = 0$, and $T_{22} + M_{22} = I$. Once these four blocks are fixed, by the equality constraint in the Schur complement of $T_{11} + M_{11}$, it must be the case that $T_{13} + M_{13} = T'_{31} + M'_{31} = 0$. In other words, Π_2 in fact admits a more compact expansion $\Pi_2 = (m(x),y)'\begin{bmatrix} T_{11}+M_{11} & T_{12}+M_{12} \\ * & I \end{bmatrix}(m(x),y)$ where by matching the terms and invoking the independence of x and y, $m'(x)[T_{11} + M_{11}]m(x) = q(x)$, $m'(x)[T_{12} + M_{12}] = s(x)$, and the gram matrix is positive semidefinite. The Schur complement of the I block therefore gives an explicit SOS parameterization of Π_1 in the $m(x)$ basis. ∎

Lemma 3 trivially extends to '$q(x) - \sum_{j=1}^{m}\|s_j(x)\|^2$ is SOS', again via Schur complement of the Gram matrix. The extended condition can be mapped to (7d), with $-\frac{\partial V_i}{\partial x_i}f_i - \frac{1}{2}\sum_{\substack{j=1 \\ j \neq i}}^{m}\|h_{ji}\|^2$ as $q(x)$, and $\frac{\partial V_i g_{ij}}{\sqrt{2}\partial x_i}$ as $s_j(x)$. Now the constraint (7d) is linear in V_i and can be readily handled by a SOS program.

V. Examples

The examples are run on a MacBook Pro with 2.9GHz i7 processor and 16GB memory. The LMI problem specifications are parsed via CVX [9], SOS problems are parsed via SPOTLESS [19], and both are then solved via MOSEK [10]. The source code is available online.[1]

A. Randomly Generated LTI Systems

We randomly generate 1000 candidate A matrices of various sizes that admit block-diagonal Lyapunov matrices of various block sizes. We facilitate the sampling process by biasing A towards negative block-diagonal dominance (to make it more likely an eligible candidate), and then pass this sample A into the full LMI (5) to check if the LMI (a necessary and sufficient condition) produces a block-diagonal Lyapunov matrix, if not, the sample is rejected. Table I records the average run time comparison of this full LMI (5) and our proposed sparse LMIs (6) and Riccati equations algorithms. It also records the success rate of identity scaling and σ_1-scalings for hand-picking the M_{ij} term in the Riccati equations.

TABLE I
RUN-TIME AND SUCCESS RATES FOR LTI SYSTEMS

Size of A		50	100	100	1000
Block Size		25	10	25	25
Run-time (sec)	Full LMI	.2	1.03	2.26	N/A
	Sparse LMIs	.18	.83	1.76	N/A
	Riccati eqns	1.44e-2	1.1e-3	1.89e-2	.5
Success Rate (Riccati eqns)	Identity	70%	56%	68%	N/A
	σ_1-scaling	78%	69%	65%	N/A

The first two N/As in the last column indicate the LMI-based methods run into memory issues and the solver is forced to stop. Riccati-equations is still able to solve some of these sampled problems, but without the baseline LMI feasibility, its success rate is not available either, hence the last two N/As. We note that when the Riccati equations are feasible, they are the fastest method to check stability of the generated A matrices. The computational saving becomes more significant as the dimension goes up thanks to scalable algorithms solving Riccati equations. We also note that there is no clear winner between identity scaling and σ_1 scaling when it comes to producing feasible Riccati equations. In practice, one may want to try both as there is very little added real time computational cost of doing so.

B. Lotka-Volterra System

We take from [1] the Lotka-Volterra example and verify its stability. It is a 16-dimensional polynomial system[2], and is thus beyond the reach of direct SOS optimization. In that paper, the authors handle the task by first developing a graph partition based algorithm, which finds a 3-way partition scheme for this example, and then a non-sparse or

[1]codes available at https://github.com/shensquared/ComposableVerification
[2]we omit explicitly listing here the dynamics for saving space and refer the reader to the source code for details

sparse SOS programs for computing a Lyapunov function for each subsystem and a composite Lyapunov function for the original system.

Their graph partition algorithm is of particular interest to us, since our compositional algorithm requires a partition but lacks the ability to search for one. Reusing their resulting 3-way partition scheme, we are also able to find a composite Lyapunov function. However, our underlying SOS formulations and hence run-time are significantly different.

TABLE II
RUN-TIME COMPARISON FOR LOTKA-VOLTERRA SYSTEM

	Non-sparse Algorithm in [1]	Sparse Algorithm in [1]	Proposed
V_1	0.25s	0.38s	0.1023s
V_2	0.25s	0.37s	0.0587s
V_3	0.44s	0.59s	0.0568s
V	1415.23s	688.54s	0.2178s

For both non-sparse and sparse algorithms in [1], after the individual $\{V_i\}_{i=1}^3$ are found by low-dimensional SOS, it relies on an additional *search* of $\alpha_i > 0$ such that $\sum_i \alpha_i V_i$ satisfies the derivative condition in x. This has to be done with yet another SOS program of considerable size since the indeterminate is the high dimensional x. Also, there is no guarantee such an $\alpha_i > 0$ always exists; it heavily depends on how compatible $\{V_i\}_{i=1}^3$ are (in our test, the $\{V_i\}_{i=1}^3$ we found do not produce feasible α_i, so we copy the run-time reported in the paper for comparison).

Our sufficient condition (7), in contrast, guarantees that $\sum_i V_i$ would *automatically* satisfy the derivative condition. This is done by imposing more restrictive conditions on $\{V_i\}_{i=1}^3$ at the their independent construction stages. In other words, once we get these ingredients, no extra work is necessary and the time spent searching for V is just the sum of time spent on each $\{V_i\}_{i=1}^3$ in much lower dimensions. Consequentially, as shown in Table II, our final run-time is 3-4 orders of magnitude faster in finding the composite V.

This example showcases the combined power of graph partition-like algorithms such as [1] and the technique proposed in this paper: the former as a prepossessing step can extend the use case, and the latter can facilitate a much faster optimization program.

VI. CONCLUSIONS

In this paper, general and constructive compositional algorithms are proposed for the computationally prohibitive problem of stability and invariance verification of large-scale polynomial systems. The key idea is to break the large system into several sub-systems, construct *independently* for each subsystem a Lyapunov-like function, and guarantee that their sum *automatically* certifies the original high-dimensional system is stable or invariant. The proposed algorithms can handle problems beyond the reach of direct optimizations, and are orders of magnitude faster than existing compositional methods.

We currently are exploring extensions of our work to compositional safety verification of multi-agent networks by leveraging the barrier certificate idea [13]. We believe the connection is immediate both technically, as barrier certificates are natural extensions of Lyapunov function, and practically, as the multi-agent network has, by definition, a compositional structure.

APPENDIX

A. Primal Proof of Theorem 1

Let us denote the left hand side of (5c) as U. Let U_k be the k-th leading principal sub-matrix of U in the blocks sense (e.g., U_1 equals $A_{11}P_1 + P_1 A'_{11}$ instead of the first scalar element in U), and let $\tilde{U}_k$ be the last column-blocks of U_k with its last block element deleted, i.e.,

$$\tilde{U}_k := \begin{bmatrix} A_{1k}P_k + P_1 A'_{k1} \\ A_{2k}P_k + P_2 A'_{k2} \\ \vdots \\ A_{(k-1)k}P_k + P_{k-1}A'_{k(k-1)} \end{bmatrix}$$

Also, define a sequence of matrices:

$$N_k := \bigoplus_{i=1}^k \left(\sum_{j=k+1}^m A_{ij} M_{ij} A'_{ij} + P_i M_{ji}^{-1} P_i{}' \right)$$

for $k = 1, 2, \ldots, m-1$, and $N_k = 0$ for $k = m$. Let $\bar{N}_k$ be the largest principal minor of N_k in the block sense. It's obvious then that by construction $N_k \succeq 0, \forall k$.

We will use induction to show that $U_k + N_k \prec 0, \forall k$, so that in the terminal case $k = n$, we would arrive at the desired Lyapunov inequality $U = U_n + N_n \prec 0$. For $k = 1$, $U_1 + N_1 \prec 0$ is trivially guaranteed by taking $i = 1$ in (6d). Suppose $U_k + N_k \prec 0$ for a particular $k \leq n-1$, let us now show that $U_{k+1} + N_{k+1} \prec 0$.

First, notice that for $k \leq n-1$, the sequence of N_k satisfies this recursive update: $N_k = n_k + \bar{N}_{k+1}$ where

$$n_k := \bigoplus_{i=1}^k \left(A_{i(k+1)} M_{i(k+1)} A'_{i(k+1)} + P_i M_{(k+1)i}^{-1} P_i{}' \right)$$

Notice also that $n_k = L_k S_{k+1} L'_k$ where

$$L_k := \left[\bigoplus_{i=1}^k P_i \quad , \quad \bigoplus_{i=1}^k A_{i(k+1)} \right]$$

$$S_{k+1} := \left[\bigoplus_{i=1}^k M_{(k+1)i}^{-1} \oplus \bigoplus_{i=1}^k M_{i(k+1)} \right]$$

From the assumption that $U_k + N_k \prec 0$, we have $U_k + N_k = U_k + L_k S_{k+1} L'_k + \bar{N}_{k+1} \prec 0$. Rearrange the terms:

$$- (U_k + \bar{N}_{k+1}) \succ L_k S_{k+1} L'_k \qquad (9)$$

Next, let D_{k+1} be the left hand side of constraint (6d) at $i = k + 1$ with the summation truncated to be only over indicies $k + 2$ to n (as opposed to be over all indices other than $k + 1$), i.e.,

$$D_{k+1} := A_{(k+1)(k+1)}P_{k+1} + P_{k+1}A'_{(k+1)(k+1)}$$
$$+ \sum_{j=k+2}^m A_{(k+1)j} M_{(k+1)j} A'_{(k+1)j}$$
$$+ \sum_{j=k+2}^m P_{k+1} M_{(k+1)j} P_{k+1}$$

and let

$$T_{k+1} := [A_{(k+1)1}, A_{(k+1)2}, \ldots, A_{(k+1)k}, \underbrace{P_{k+1}, \ldots, P_{k+1}}_{\text{repeat k times}}]$$

Then by Schur complement, constraint (6d) with $i = k+1$ is equivalent to:

$$\begin{bmatrix} D_{k+1} & T_{k+1} \\ T'_{k+1} & -S_{k+1} \end{bmatrix} \prec 0 \tag{10}$$

Use Schur complement on (10) again, this time from the opposite direction, it is also equivalent to:

$$S_{k+1} \succ -T'_{k+1} D_{k+1}^{-1} T_{k+1} \tag{11}$$

Because $P_i \succ 0, \forall i$, L_k has full row-rank, then pre- and post-multiplying (11) with L_k and L'_k preserves the positive definite order:

$$L_k S_{k+1} L'_k \succ -L_k T'_{k+1} D_{k+1}^{-1} T_{k+1} L'_k \tag{12a}$$
$$= -\tilde{U}_{k+1} D_{k+1}^{-1} \tilde{U}'_{k+1} \tag{12b}$$

The last equality is due to

$$L_k T'_{k+1} = \begin{bmatrix} A_{1(k+1)} P_{k+1} + P_1 A'_{(k+1)1} \\ A_{2(k+1)} P_{k+1} + P_2 A'_{(k+1)2} \\ \vdots \\ A_{k(k+1)} P_{k+1} + P_k A'_{(k+1)k} \end{bmatrix} = \tilde{U}_{k+1}$$

Finally, combining (9) and (12), we have

$$-(U_k + \bar{N}_{k+1}) \succ -\tilde{U}_{k+1} D_{k+1}^{-1} \tilde{U}'_{k+1} \tag{13}$$

By again taking Schur complement, this is equivalent to:

$$\begin{bmatrix} U_k + \bar{N}_{k+1} & \tilde{U}_{k+1} \\ \tilde{U}'_{k+1} & D_{k+1} \end{bmatrix} = U_{k+1} + N_{k+1} \prec 0 \tag{14}$$

which completes the induction. ∎

B. Dual Proof of Theorem 1.

We'll make use of Theorem of Alternatives [5] below.

Theorem 4: Let $\mathcal{V}$ (resp. $\mathcal{S}$) be a finite-dimensional vector space with inner product $\langle \cdot, \cdot \rangle_\mathcal{V}$ (resp. $\langle \cdot, \cdot \rangle_\mathcal{S}$). Let $\mathcal{A} : \mathcal{V} \to \mathcal{S}$ be a linear mapping, and $\mathcal{A}^{adj} : \mathcal{S} \to \mathcal{V}$ be the adjoint mapping such that $\forall x \in \mathcal{V}$ and $\forall Z \in \mathcal{S}$, $\langle \mathcal{A}(x), Z \rangle_\mathcal{S} = \langle x, \mathcal{A}^{adj}(Z) \rangle_\mathcal{V}$, and let $A_0 \in \mathcal{S}$. Then exactly one of the two statements is true:

1) There exists an $x \in \mathcal{V}$ with $\mathcal{A}(x) + A_0 > 0$.
2) There exists a $Z \in \mathcal{S}$ with $Z \gneq 0$, $\mathcal{A}^{adj}(Z) = 0$, and $\langle A_0, Z \rangle_\mathcal{S} \leq 0$.

The theorem can be intuitively thought of as a generalization of Farkas' Lemma to non-polyhedral convex cones. A complete proof can be found in [5]. Tailored to our need, $\mathcal{V}$ and $\mathcal{S}$ are taken as the cone of positive semidefinite matrices equipped with $\langle A, B \rangle = trace(AB)$, $>$ (resp. $\geq$) therefore means $\succ$ (resp. $\succeq$), and $Z \gneq 0$ means $Z \succeq 0, Z \neq 0$.

Now, let us first match (5) to the first statement to get the linear mapping $\mathcal{A}$ and the adjoint. Then (5) is infeasible if

and only if there exists a Z with the same size and partition of A (as well as P) such that:

$$Z \gneq 0 \tag{15a}$$
$$Z_{ii} A_{ii} + A'_{ii} Z_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^{m} A'_{ij} Z_{ij} + Z'_{ij} A_{ij} \gneq 0, \forall i \tag{15b}$$

Finding the alternative of (6) is less straightforward. It turns out it is easier to work with an equivalent form of (6):

$$P_i > 0, M_{ij} > 0, \forall i, j, i \neq j \tag{16a}$$
$$P_i A_{ii} + A'_{ii} P_i + \sum_{\substack{j=1 \\ j \neq i}}^{m} P_i A_{ij} M_{ij} A'_{ij} P_i + M_{ji}^{-1} < 0, \forall i \tag{16b}$$

We then match (16) also to the first statement. It is infeasible if and only if there exists a set of $\{T_i\}_{i=1}^m$ where each T_i is with the same size and partition of A (as well as P) such that:

$$T_i \gneq 0, \forall i \tag{17a}$$
$$A_{ii} T_{iii} + T_{iii} A'_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^{m} T_{iij} A'_{ij} + A_{ij} T'_{iij} \gneq 0, \forall i \tag{17b}$$
$$T_{iii} \leq T_{j_{ii}}, \forall i, j, i \neq j \tag{17c}$$

It is clear that the feasibility of (15) implies that of (17) because one can simply let $T_i = Z^{-1}, \forall i$ (the reverse implication does not necessarily hold; it is possible the set $\{T_i\}_{i=1}^m$ can not be 'squashed' into a single Z^{-1}). Therefore, the infeasibility of (5) implies the infeasibility of (16) which is equivalent to (6). Flipping both sides of the last statement, (6) is feasible implies (5) is feasible. ∎

REFERENCES

[1] J. Anderson and A. Papachristodoulou, "A Decomposition Technique for Nonlinear Dynamical System Analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1516–1521, 2012.
[2] M. Araki, "Stability of Large-Scale Nonlinear Systems-Quadratic-Order Theory of Composite-System Method Using M-Matrices," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 129–142, 1978.
[3] M. Arcak and E. D. Sontag, "Diagonal stability of a class of cyclic systems and its connection with the secant criterion," *Automatica*, vol. 42, no. 9, pp. 1531–1537, 2006.
[4] L. Bakule, "Decentralized control: An overview," *Annual Reviews in Control*, vol. 32, no. 1, pp. 87–98, 2008.
[5] V. Balakrishnan and L. Vandenberghe, "Semidefinite programming duality and linear time-invariant systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 30–41, 2003.
[6] G. P. Barker, A. Berman, and R. J. Plemmons, "Positive diagonal solutions to the Lyapunov equations," *Linear and Multilinear Algebra*, vol. 5, no. 4, pp. 249–256, 1978.
[7] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, 1994, vol. 15.

[8] D. Carlson, D. Hershkowitz, and D. Shasha, "Block diagonal semistability factors and Lyapunov semistability of block triangular matrices," *Linear Algebra and Its Applications*, vol. 172, no. C, pp. 1–25, 1992.

[9] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[10] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 8.*, 2017.

[11] K. S. Narendra and R. Shorten, "Hurwitz stability of Metzler matrices," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1484–1487, 2010.

[12] P. A. Parrilo, "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization," PhD Thesis, California Institute of Technology, 2000.

[13] S. Prajna, A. Jadbabaie, G. J. Pappas, and S. Member, "A Framework for Worst-Case and Stochastic Safety Verification Using Barrier Certificates," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[14] A. Rantzer, "Dynamic dual decomposition for distributed control," *Proceedings of the American Control Conference*, pp. 884–888, 2009.

[15] R. N. Shorten and K. S. Narendra, "On a theorem of Redheffer concerning diagonal stability," *Linear Algebra and Its Applications*, vol. 431, pp. 2317–2329, 2009.

[16] C. Sloth, R. Wisniewski, and G. J. Pappas, "On the existence of compositional barrier certificates," *Proceedings of the IEEE Conference on Decision and Control*, pp. 4580–4585, 2012.

[17] A. Sootla and J. Anderson, "On existence of solutions to structured Lyapunov inequalities," *Proceedings of the American Control Conference*, vol. 2016-July, no. 0, pp. 7013–7018, 2016.

[18] T. Tanaka and C. Langbort, "The Bounded Real Lemma for Internally Positive Systems and H-Infinity Structured Static State Feedback," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2218–2223, 2011.

[19] M. M. Tobenkin, F. Permenter, and A. Megretski, "Spotless polynomial and conic optimization," https://github.com/spot-toolbox/spotless, 2015.

[20] U. Topcu, A. Packard, and R. Murray, "Compositional stability analysis based on dual decomposition," *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, pp. 1175–1180, 2009.

[21] J. C. Willems, "Lyapunov functions for diagonally dominant systems," *Automatica*, vol. 12, no. 5, pp. 519–523, 1976.

[22] Y. Zheng, M. Kamgarpour, A. Sootla, and A. Papachristodoulou, "Convex Design of Structured Controllers using Block-Diagonal Lyapunov Functions Convex Design of Structured Controllers using Block-Diagonal Lyapunov Functions," 2017. [Online]. Available: https://arxiv.org/abs/1709.00695