Quasiconvex Sum-of-Squares Programming

Peter Seiler and Gary J. Balas

Abstract— **A sum-of-squares program is an optimization problem with polynomial sum-of-squares constraints. The constraints and the objective function are affine in the decision variables. This paper introduces a generalized sum-of-squares programming problem. This generalization allows one decision variable to enter bilinearly in the constraints. The bilinear decision variable enters the constraints in a particular structured way. The objective function is the single bilinear decision variable. It is proved that this formulation is quasiconvex and hence the global optima can be computed via bisection. Many nonlinear analysis problems can be posed within this framework and two examples are provided.**

I. Introduction

A polynomial is a sum-of-squares (SOS) if it can be expressed as a sum of squares of other polynomials. An SOS program is an optimization problem with polynomial SOS constraints [16], [17]. The constraints and objective function are affine in the decision variables. The algorithms to solve SOS programs rely on connections between SOS polynomials and positive semidefinite matrices [6], [18], [16], [12], [17]. In particular, software is available to convert the SOS programs to semidefinite programs [19], [13], [2]. Many nonlinear analysis problems, e.g. Lyapunov stability analysis, can be formulated within this optimization framework.

This paper provides a generalization to SOS programming problems. One decision variable is allowed to enter bilinearly into the SOS constraints. The bilinear decision variable enters the constraints in a particular structured way. The objective function of the generalized SOS program is the decision variable that enters bilinearly. It is proved that the generalized SOS program is quasiconvex and hence the global optima can be computed via bisection. A standard SOS feasibility problem is solved at each step of the bisection. Software has been developed to solve this particular bisection problem. The relation between SOS programs and generalized SOS programs is analogous to the relation between semidefinite programs and generalized eigenvalue problems. Algorithms for solving generalized eigenvalue optimization problems could potentially be used to solve generalized SOS programs with significantly less computation than bisection. This approach is not taken since the current theory and available software for solving large, sparse generalized eigenvalue problems is not as developed as for semidefinite programming problems.

The generalized SOS program also finds many applications in nonlinear analysis. Two examples provided in this paper are the computation of maximum decay rates and estimates of regions of attraction. In the second example an estimate of the region of attraction is formulated as a set containment condition. A sufficient condition for this set containment condition is reformulated as an SOS constraint with one bilinear variable. This leads to a generalized SOS program. Similar set containment conditions appear in many local nonlinear analysis problems, e.g in computations for reachability sets, input-output gains, and robustness with respect to uncertainty for nonlinear polynomial systems [24], [25], [28]. Thus generalized SOS programs play a role in many local nonlinear analyses.

The remainder of the paper has the following structure. The next section briefly reviews background material on semidefinite programming, generalized eigenvalue problems and sum-of-squares polynomial optimizations. The generalized SOS program is introduced in Section III. The proof of quasiconvexity is provided in the same section. Two generalized SOS examples are presented in Section IV. Finally, conclusions are given in Section V.

II. Background

This section briefly reviews the main results for semidefinite programming [3], [31], generalized eigenvalue problems [4] and sum-of-squares polynomial optimizations [6], [12], [16], [17], [18], [20]. Additional details can be found in the references.

A. Semidefinite Programming

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive semidefinite if $x^T A x \geq 0$ for all $x \in \mathbb{R}^n$. Positive semidefinite matrices are denoted by $A \succeq 0$. A semidefinite program is an optimization problem of the following form:

$$\begin{array}{ll} \min_u & c^T u \\ \text{subject to:} & A(u) := A_0 + \sum_{k=1}^r u_k A_k \succeq 0 \end{array} \qquad (1)$$

The symmetric matrices $A_0, \ldots, A_r \in \mathbb{R}^{n\times n}$ and the vector $c \in \mathbb{R}^r$ are given data. The vector $u \in \mathbb{R}^r$ is the decision variable. The constraint, $A(u) \succeq 0$, is called a linear matrix inequality (LMI) although it is technically affine in the decision variables. The single LMI constraint is without loss of generality; multiple LMI constraints can be block-diagonally concatenated for form a single larger LMI constraint. Equation 1 is referred to as the primal problem.

The dual associated with this primal problem is:

$$\begin{array}{ll} \max_Q & -\mathbf{Tr}\,[A_0 Q] \\ \text{subject to:} & \mathbf{Tr}\,[A_k Q] = c_k \quad k = 1, \ldots, r \\ & Q \succeq 0 \end{array} \qquad (2)$$

P. Seiler is with Aerospace and Engineering Mechanics Department, University of Minnesota, seiler@aem.umn.edu

G.J. Balas is with Aerospace and Engineering Mechanics Department, University of Minnesota, balas@umn.edu

where $Q = Q^T \in \mathbb{R}^{n\times n}$ is the decision variable for the dual problem. $\mathbf{Tr}[\cdot]$ denotes the trace of a matrix. This dual problem can be recast in the form of Equation 1 and thus it is also a semidefinite program. The primal and dual formulations may appear restrictive but they are quite versatile and SDPs find applications in many problems of interest [3]. Moreover, SDPs are convex optimizations and quality software exists to solve these problems. For example, the Robust Control Toolbox function `mincx` solves the primal form [1]. SeDuMi [23], [22] is a freely solver for MATLAB that simultaneously solves the primal and dual forms of a semidefinite program.

In some cases, the only goal is to find a decision variable that satisfies a linear matrix inequality constraint. These are linear matrix inequality feasibility problems. The following is an example:

$$\text{Find } u \in \mathbb{R}^r \text{ such that } A_0 + \sum_{k=1}^{r} u_k A_k \succeq 0 \tag{3}$$

B. Generalized Eigenvalue Problems

A generalized eigenvalue problem [4] is a optimization problem of the following form:

$$\begin{array}{ll} \min_u & \lambda_{max}\left(A(u), B(u)\right) \\ \text{subject to:} & B(u) \succ 0 \\ & C(u) \succeq 0 \end{array} \tag{4}$$

where $\lambda_{max}(A,B)$ is the maximum generalized eigenvalue of (A,B). The vector $u \in \mathbb{R}^r$ is the decision variable and A, B, C depend affinely on u. This optimization involves convex (LMI) constraints and a quasiconvex cost function. Thus a generalized eigenvalue problem is a quasiconvex optimization.

This optimization can be equivalently written in the following form:

$$\begin{array}{ll} \min_{t,u} & t \\ \text{subject to:} & tB(u) - A(u) \succeq 0 \\ & B(u) \succ 0 \\ & C(u) \succeq 0 \end{array} \tag{5}$$

where the scalar $t \in \mathbb{R}$ is an additional decision variable. In this form the second and third constraints in the optimization problem are LMIs and the cost is a linear function of the decision variables. However this is not an SDP because the first constraint is bilinear in the decision variables t and u.

The global minimum of a generalized eigenvalue problem can be computed via bisection on t. Each step of the bisection involves holding t fixed and solving for u that satisfies the matrix inequalities in Optimization 5. This is an LMI feasibility problem at each step of the bisection. Algorithms have also been designed specifically for solving generalized eigenvalue problems without resorting to bisection [4], [14]. For example, the Robust Control Toolbox function `gevp` [1] solves generalized eigenvalue problems using the Projective Method in [14]. The computational complexity of these algorithms is roughly the same as one LMI feasibility problem for a fixed t and hence they are substantially faster than applying bisection.

C. SOS Polynomials

$\mathbb{N}$ denotes the set of nonnegative integers, $\{0, 1, \ldots\}$, and $\mathbb{N}^n$ is the set of n-dimensional vectors with entries in $\mathbb{N}$. For $\alpha \in \mathbb{N}^n$, a monomial in variables $\{x_1, \ldots, x_n\}$ is given by $x^\alpha \doteq x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$. The degree of a monomial is defined as $\deg x^\alpha \doteq \sum_{i=1}^n \alpha_i$. A polynomial is a finite linear combination of monomials:

$$p \doteq \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha = \sum_{\alpha \in \mathcal{A}} c_\alpha x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

where $c_\alpha \in \mathbb{R}$ and $\mathcal{A}$ is a finite collection of vectors in $\mathbb{N}^n$. $\mathbb{R}[x]$ denotes the set of all polynomials in variables $\{x_1, \ldots, x_n\}$ with real coefficients. Using the definition of $\deg$ for a monomial, the degree of p is defined as $\deg p \doteq \max_{\alpha \in \mathcal{A},\ c_\alpha \neq 0} \left[\deg x^\alpha\right]$.

A polynomial p is a sum-of-squares (SOS) if there exist polynomials $\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. The set of SOS polynomials is a subset of $\mathbb{R}[x]$ and is denoted by $\Sigma[x]$. If p is a sum-of-squares then $p(x) \geq 0\ \forall x \in \mathbb{R}^n$. Thus $p \in \Sigma[x]$ is a sufficient condition for a polynomial to be globally non-negative. The converse is not true, i.e. non-negative polynomials are not necessarily SOS polynomials [21].

Define z as the column vector of all monomials in variables $\{x_1, \ldots, x_n\}$ of degree $\leq d$: [1]

$$z \doteq \left[1,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_1^2,\ x_1 x_2,\ \ldots,\ x_n^2,\ \ldots,\ x_n^d\right]^T \tag{6}$$

There are $\binom{k+n-1}{k}$ monomials in n variables of degree k. Thus z is a column vector of length $l_z \doteq \sum_{k=0}^d \binom{k+n-1}{k} = \binom{n+d}{d}$. If f is a polynomial in n variables with degree $\leq d$ then by definition f is a finite linear combination of monomials of degree $\leq d$. Consequently, there exists $a \in \mathbb{R}^{l_z}$ such that $f = a^T z$.

Two facts that follow from Theorem 1 and its preceding Lemma in [20] are:

1) If p is a sum-of-squares then p must have even degree.
2) If p is degree $2d$ ($d \in \mathbb{N}$) and $p = \sum_{i=1}^m f_i^2$ then $\deg f_i \leq d\ \forall i$.

The following theorem, introduced as the "Gram Matrix" method by [6], connects SOS polynomials and positive semidefinite matrices. This result can be found more recently in [18].

Theorem 1: Suppose $p \in \mathbb{R}[x]$ is a polynomial of degree $2d$ and z is the $l_z \times 1$ vector of monomials defined in Equation 6. Then p is a SOS if and only if there exists a symmetric matrix $Q \in \mathbb{R}^{l_z \times l_z}$ such that $Q \succeq 0$ and $p = z^T Q z$.

Proof:

[1] Any ordering of the monomials can be used to form z. In Equation 6, x^α precedes x^β in the definition of z if:

$$\begin{array}{l} \deg x^\alpha < \deg x^\beta \\ \qquad \text{or} \\ \deg x^\alpha = \deg x^\beta \text{ and the first nonzero entry of } \alpha - \beta \text{ is } > 0 \end{array}$$

($\Rightarrow$) If p is a SOS, then there exists polynomials $\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. By fact 2 above, $\deg f_i \le d$ for all i. Thus, for each f_i there exists a vector, $a_i \in \mathbb{R}^{l_z}$, such that $f_i = a_i^T z$. Define the matrix, $A \in \mathbb{R}^{l_z \times m}$, whose i^{th} column is a_i and define $Q \doteq AA^T \succeq 0$. Then $p = z^T Q z$.

($\Leftarrow$) Assume there exists $Q = Q^T \in \mathbb{R}^{l_z \times l_z}$ such that $Q \succeq 0$ and $p = z^T Q z$. Define $m \doteq rank(Q)$. There exists a matrix $A \in \mathbb{R}^{l_z \times m}$ such that $Q = AA^T$. Let a_i denote the i^{th} column of A and define the polynomials $f_i \doteq z^T a_i$. By definition of f_i, $p = z^T(AA^T)z = \sum_{i=1}^m f_i^2$. ■

D. SOS Programs

A sum-of-squares program is an optimization problem with a linear cost and affine SOS constraints on the decision variables [19]:

$$\min_{u \in \mathbb{R}^r} c^T u \tag{7}$$
$$\text{subject to: } a_k(x,u) \in \Sigma[x], \quad k = 1, \dots N$$

$u \in \mathbb{R}^r$ are decision variables. The polynomials $\{a_k\}$ are given as part of the problem data and are affine in u, i.e. they are of the form:

$$a_k(x,u) := a_{k,0}(x) + a_{k,1}(x)u_1 + \cdots + a_{k,n}(x)u_n \tag{8}$$

Many nonlinear analysis problems can be posed within this optimization framework [16].

Theorem 1 is used to convert an SOS program into a semidefinite-programming problem. For example, the constraint $a_k(x,u) \in \Sigma[x]$ can be equivalently written as:

$$a_{k,0}(x) + a_{k,1}(x)u_1 + \cdots + a_{k,n}(x)u_n = z^T Q z \tag{9}$$
$$Q \succeq 0 \tag{10}$$

Q is a new matrix of decision variables that is introduced when converting an SOS constraint to an LMI constraint. Equating the coefficients of $z^T Q z$ and $a_k(x,u)$ imposes linear equality constraints on the decision variables u and Q. Thus, Equation 9 can be rewritten as a set of linear equality constraints on the decision variables. All SOS constraints in Equation 7 can be replaced in this fashion with linear equality constraints and LMI constraints. As a result, the SOS program in Equation 7 can be written in the SDP dual form (Equation 2).

There is software available to perform the conversion from SOS programs to SDPs. SOSOPT [2], SOSTOOLS [19], and Yalmip [13] are freely available MATLAB toolboxes for solving SOS optimizations. These packages allow the user to specify polynomial constraints using a symbolic or polynomial toolbox. The toolboxes convert the SOS optimization into an SDP which is solved with a freely available SDP solver. Finally these toolboxes convert the SDP solution back to a polynomial solution. A drawback is that the size of the resulting SDP grows rapidly in both the number of variables and degrees of the polynomials in the SOS optimization. While various techniques can be used to exploit the problem structure [9], this computational growth is a generic trend in SOS optimizations. This roughly limits SOS methods to nonlinear analysis problems with at most 8-10 states and polynomial models with degree of at most 3-5.

III. Quasiconvex SOS Programs

Generalized eigenvalue problems extend SDPs by allowing one decision variable to enter bilinearly in the matrix constraints. Similarly, SOS programs can be extended to allow the one decision variable to enter bilinearly in the SOS constraints. A generalized SOS program is an optimization of the form:

$$\begin{array}{ll} \min_{t,u} \ t & \\ \text{subject to:} & t b_k(x,u) - a_k(x,u) \in \Sigma[x], \quad k = 1, \dots N \\ & b_k(x,u) \in \Sigma[x], \quad k = 1, \dots N \\ & c_k(x,u) \in \Sigma[x], \quad k = 1, \dots M \end{array} \tag{11}$$

$t \in \mathbb{R}$ and $u \in \mathbb{R}^r$ are decision variables. The polynomials $\{a_k\}$, $\{b_k\}$, and $\{c_k\}$ are given data and are affine in u. The optimization cost is linear in the decision variables and the constraints $b_k(x,u) \in \Sigma[x]$ and $c_k(x,u) \in \Sigma[x]$ are standard SOS constraints. This is not an SOS program because the constraints $t b_k(x,u) - a_k(x,u) \in \Sigma[x]$ are bilinear in the decision variables t and u. However, the generalized SOS program is quasiconvex. The proof of this statement follows from the next Lemma and Theorem.

Lemma 1: If c_1, $c_2 \in \mathbb{R}$ are non-negative and p_1, $p_2 \in \Sigma[x]$ then $c_1 p_1 + c_2 p_2 \in \Sigma[x]$.

Proof: Since $p_1, p_2 \in \Sigma[x]$, there exists polynomials $\{f_i\}_{i=1}^{m_1}$ and $\{g_i\}_{i=1}^{m_2}$ such that $p_1 = \sum_{i=1}^{m_1} f_i^2$ and $p_2 = \sum_{i=1}^{m_2} g_i^2$. Define $h_i = \sqrt{c_1} f_i$ for $i = 1, \dots, m_1$ and $h_{m_1+i} = \sqrt{c_2} g_i$ for $i = 1, \dots, m_2$. Then $c_1 p_1 + c_2 p_2 = \sum_{i=1}^{m_1+m_2} h_i^2$. By definition, $c_1 p_1 + c_2 p_2 \in \Sigma[x]$. ■

Theorem 2: Optimization 11 is quasiconvex.

Proof: For notational simplicity consider the case where $N = 1$ and $M = 1$. The extension to multiple SOS and generalized SOS constraints is straightforward. The generalized SOS program can be equivalently written:

$$\begin{array}{ll} \min_u \ f(u) & \\ \text{subject to:} & c(x,u) \in \Sigma[x] \end{array} \tag{12}$$

where the function $f : \mathbb{R}^r \to \mathbb{R}$ is defined as:

$$f(u) := \begin{cases} +\infty & b(x,u) \notin \Sigma[x] \\ \min_t t & b(x,u) \in \Sigma[x] \\ \text{subject to:} & \\ t b(x,u) - a(x,u) \in \Sigma[x] & \end{cases} \tag{13}$$

The constraint $c(x,u) \in \Sigma[x]$ in Optimization 12 is a convex constraint on u. Hence the proof is completed by showing that the cost function f is quasiconvex.

Quasiconvexity of f follows by proving that all sublevel sets of f are convex. Consider the sublevel set $S_\gamma := \{u \in \mathbb{R}^r : f(u) \le \gamma\}$. If $\gamma = +\infty$ or $S_\gamma = \emptyset$ then S_γ is trivially convex so assume $\gamma < +\infty$ and $S_\gamma \ne \emptyset$. Take any $u_1, u_2 \in S_\gamma$. By the definition of f, there exists $t_i \le \gamma$ such that $t_i b(x,u_i) - a(x,u_i) \in \Sigma[x]$ and $b(x,u_i) \in \Sigma[x]$ for $i = 1, 2$.

For any $\alpha \in [0,1]$ define the convex combination $u_\alpha := \alpha u_1 + (1-\alpha)u_2$. It is now shown that $f(u_\alpha) \le \gamma$. Define $t_0 = \max(t_1, t_2) \le \gamma$. For $i = 1, 2$,

$$\begin{aligned} &t_0 b(x,u_i) - a(x,u_i) \\ &\qquad = (t_0 - t_i) b(x,u_i) + (t_i b(x,u_i) - a(x,u_i)) \end{aligned}$$

It follows from Lemma 1 that $t_0 b(x,u_i) - a(x,u_i) \in \Sigma[x]$ for $i = 1, 2$.

Since a and b are affine in u,

$$\begin{aligned} &t_0 b(x,u_\alpha) - a(x,u_\alpha) \\ &= \alpha(t_0 b(x,u_1) - a(x,u_1)) + (1-\alpha)(t_0 b(x,u_2) - a(x,u_2)) \end{aligned}$$

Thus $t_0 b(x,u_\alpha) - a(x,u_\alpha) \in \Sigma[x]$ by another application of Lemma 1. $b(x,u_\alpha) \in \Sigma[x]$ follows similarly from Lemma 1. This implies that the constraints in Optimization 13 are feasible at $(t,u) = (t_0, u_\alpha)$ and hence $f(u_\alpha) \le t_0 \le \gamma$. Therefore $u_\alpha \in S_\gamma$ and S_γ is convex. f is quasiconvex since this holds for any γ. ■

A consequence of Theorem 2 is that the global minimum of a generalized SOS program can be computed via bisection on t. Each step of the bisection involves holding t fixed and solving for u that satisfies the SOS constraints in Optimization 11. This can be converted to an LMI feasibility problem at each step of the bisection. The SOSOPT software [2] contains a function `gsosopt` to solve generalized SOS programs using bisection. For large problems the conversion from polynomial SOS constraints to LMI constraints (using the Gram matrix method) can be computationally demanding. `gsosopt` performs the SOS to LMI constraint conversion once and avoids repeating this computation during the bisection iteration.

The Gram matrix method can also be used to convert a generalized SOS program to a generalized eigenvalue problem. One minor technical issue is that quasiconvex SOS programs, as formulated in Equation 11, lead to generalized eigenvalue problems (Equation 4) with $B(u) \succeq 0$, i.e the constraint is only semidefinite. A proper generalized eigenvalue problem with $B(u) \succ 0$ arises if the constraints in Equation 11) are modified to $b_k(x,u) - l_k(x) \in \Sigma[x]$ where $l_k(x)$ are strictly positive definite functions with small coefficients. Algorithms specifically designed for solving generalized eigenvalue problems [4], [14] can then be applied. Unfortunately the theory and available software for generalized eigenvalue problems are not as well-developed as for SDPs. The Robust Control Toolbox function `gevp` [1] is not suitable for moderate to large generalized SOS programs because it does not exploit the sparsity that arises in the matrix data. It will be the subject of future work to investigate algorithms for directly solving generalized eigenvalue problems that arise from generalized SOS programs. For example, the algorithms in [15] and [8] could be applied to solve quasiconvex SOS problems.

IV. Examples

This section presents two generalized SOS programs that arise in nonlinear analysis. All computations were performed on a 2.16GHz Intel processor.

A. Maximum Decay Rate

Consider the following third-order nonlinear system:

$$\dot{x} = f(x) := A_1 z_1(x) + A_2 z_2(x) + A_3 z_3(x) \tag{14}$$

where:

$$z_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad z_2 = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}, \quad z_3 = \begin{bmatrix} x_1^3 \\ x_1^2 x_2 \\ x_1 x_2^2 \\ x_2^3 \end{bmatrix}$$

and

$$A_1 = \begin{bmatrix} -4 & 5 \\ -1 & -2 \end{bmatrix}, \quad A_2 = \frac{1}{4}\begin{bmatrix} 3 & 6 & 3 \\ 1 & 2 & 1 \end{bmatrix},$$
$$A_3 = \frac{1}{8}\begin{bmatrix} -1 & 0 & -9 & 6 \\ 0 & -3 & 6 & -7 \end{bmatrix}$$

If there exists a $V := x^T P x$ such that $V \ge x_1^2 + x_2^2$ and $\dot{V} \le -2rV$ then $x = 0$ is a globally exponentially stable and all trajectories satisfy $\|x(t)\| \le \sqrt{\kappa(P)} e^{-rt} \|x(0)\|$ where $\kappa(P)$ is the condition number of P (Section 5.1.3 of [3]). V is a Lyapunov function that proves the decay rate of the system is at least r.

The largest bound on the decay rate for the polynomial system can be computed by a generalized SOS program:

$$\begin{aligned} &\min_{t,u} \; t \\ &\text{subject to:} \quad tV(x,u) - \nabla V(x,u) f(x) \in \Sigma[x] \\ &\qquad\qquad\qquad V(x,u) - x^T x \in \Sigma[x] \end{aligned} \tag{15}$$

where $t := -2r$ and the remaining decision variables are the entries of the symmetric matrix P, i.e. $u := [P_{1,1}, \; P_{2,1}, \; P_{2,2}]^T$. This problem took 1.35sec to solve using `gsosopt` [2]. The largest bound on the decay rate is $r = 1.93$ and the Lyapunov function is:

$$V(x) = 2.12 x_1^2 - 4.01 x_1 x_2 + 25.48 x_2^2 \tag{16}$$

Figure 1 shows the phase plane for this nonlinear system and the level sets for this Lyapunov function. Bisecting with `sosopt` and SOSTOOLs took 1.75sec and 2.73sec, respectively. Bisecting with these functions is slower because they perform the conversion from SOS to LMI constraints at each step of the bisection. As noted previously `gsosopt` only performs this conversion once prior to the bisection. The difference in computation times is relatively small on this example but it can be significant on larger problems.

For comparison, the decay rate of the linearization A_1 can be computed by a generalized eigenvalue problem:

$$\begin{aligned} &\min_{t,P} \; t \\ &\text{subject to:} \quad tP - \left(A_1^T P + P A_1\right) \succeq 0 \\ &\qquad\qquad\qquad P \succeq I \end{aligned} \tag{17}$$

where $t := -2r$ and the remaining decision variables are the entries of P. This problem was solved using `gevp`. The largest bound on the decay rate is $\alpha = 3.00$ and the Lyapunov function is:

$$V_{LIN}(x) = 1.95 x_1^2 - 3.89 x_1 x_2 + 9.73 x_2^2 \tag{18}$$

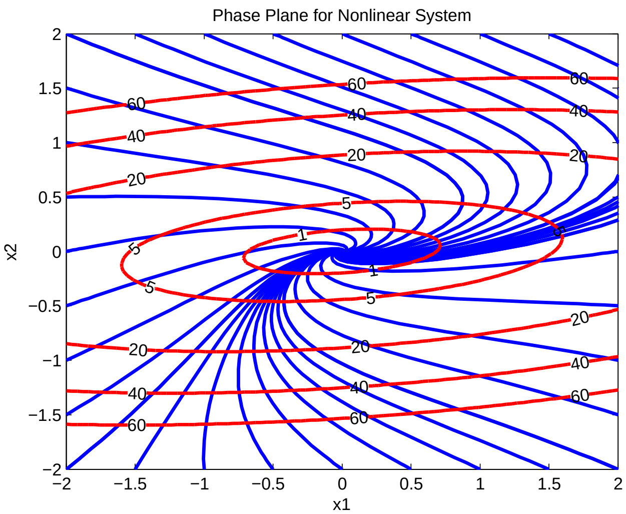

Fig. 1. Nonlinear system trajectories and Lyapunov function level sets.

The decay rate for a stable linear system is equal to the magnitude of the maximum real part of the eigenvalues of the state matrix. The eigenvalues of A_1 are at $-3 \pm 2i$ and thus the decay rate is 3.00. This is in agreement with the decay rate computed by `gevp`. For linear systems, the decay rate computed by Optimization 17 is exact [3]. For nonlinear systems, the decay rate computed by Optimization 15 need not be exact. Thus $\alpha = 1.93$ is only a lower bound on the maximal decay rate for the polynomial system (Equation 14). Increasing the degree of the polynomial Lyapunov function in the generalized SOS program may improve the bound on the decay rate at the expense of additional computation.

B. Region of Attraction Estimation

Consider the Van der Pol oscillator:

$$\dot{x} = f(x) := \begin{bmatrix} -x_2 \\ x_1 + (x_1^2 - 1)x_2 \end{bmatrix} \tag{19}$$

This system has an equilibrium point at $x = 0$. The equilibrium point is locally asymptotically stable but not globally asymptotically stable. The region of attraction $\mathcal{R}$ is the set of initial conditions whose trajectories converge back to $x = 0$. Lyapunov theory can be used to formulate a generalized SOS program to estimate a subset of $\mathcal{R}$. Standard local Lyapunov theorems are applied in this example [33], [11].

The linearization of Equation 19 around $x = 0$ is

$$A = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \tag{20}$$

The eigenvalues of A are $-0.5 \pm 0.866j$. The Lyapunov equation $A^T P + PA = -I$ is satisfied by

$$P = \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1.0 \end{bmatrix} > 0 \tag{21}$$

Thus $V := x^T P x$ is a Lyapunov function that proves $x = 0$ is a locally asymptotically stable equilibrium point. For $\gamma > 0$, denote the sublevel set of V by $\Omega_\gamma := \{x \in \mathbb{R}^n : V(x) \leq \gamma\}$. If

$$\Omega_\gamma \subset \{x \in \mathbb{R}^n \;:\; \nabla V(x) f(x) < 0\} \tag{22}$$

then for all $x_0 \in \Omega_\gamma$, the solution of Equation (19) starting from $x(0) = x_0$ satisfies $x(t) \to 0$ as $t \to \infty$. Thus $\Omega_\gamma \subset \mathcal{R}$ if the set containment condition Equation 22 is satisfied.

The set containment condition in Equation 22 can be converted to an algebraic inequality constraint. Define $l(x) = 10^{-6} x^T x$. The set containment is satisfied if there exists a polynomial s such that

$$s(x) \geq 0 \quad \forall x \tag{23}$$

$$(V(x) - \gamma)s(x) \geq \nabla V(x) f(x) + l(x) \quad \forall x \tag{24}$$

It is straightforward to verify that these algebraic conditions imply the set containment. If $x \in \Omega_\gamma$ then the left side of Equation 24 is non-positive. This implies that $\nabla V(x) f(x) \leq -l(x) < 0$ and hence the set containment holds. This is a generalization of the S-procedure [3].

A generalized SOS program can be used to compute the largest inner estimate of the region of attraction provable with the given $V(x)$:

$$\begin{aligned} &\min_{t,u} \; t \\ &\text{subject to:} \\ &s(x,u) \in \Sigma[x] \\ &ts(x,u) + (V(x)s(x,u) - \nabla V(x,u) f(x) - l(x)) \in \Sigma[x] \end{aligned} \tag{25}$$

$t := -\gamma$ and the remaining decision variables are the coefficients of s. s was chosen to be of the form $s(x,u) = z^T U z$ where

$$z := \begin{bmatrix} x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix}^T \tag{26}$$

U is a 5×5 symmetric matrix and the decision variables $u \in \mathbb{R}^{15}$ consist of the independent entries of U. This problem took 1.63sec to solve using `gsosopt`. $\gamma = 2.30$ is the largest level set of V contained within the region of attraction. The optimal function s is

$$\begin{aligned} s(x) = {} & 0.232x_1^4 + 0.0536x_1^3 x_2 - 0.0611x_1^2 x_2^2 - 0.0954x_1 x_2^3 \\ & + 0.177x_2^4 + 0.268x_1^2 - 0.103x_1 x_2 + 0.239x_2^2 \end{aligned} \tag{27}$$

Figure 2 is a phase plane plot that shows the region of attraction estimate $\Omega_{\gamma=2.30}$ (dashed ellipse). The figure also shows the limit cycle (solid curve) that forms the exact boundary of the region of attraction. The region of attraction for the Van der Pol oscillator consists of all points in the interior of this limit cycle. There are more sophisticated algorithms for estimating the region of attraction estimate [24], [25], [30], [29], [28], [5], [7], [10], [26], [27], [32], [16] and these algorithms achieve better inner estimates on the ROA than those shown in Figure 2. Set containment conditions appear in most of these advanced algorithms. For example the V and s steps in the V-s iteration [24], [29], [28] involve set containments that are reformulated as quasiconvex SOS optimizations. Thus the quasiconvex code reported in this paper can be used to improve the computational efficiency of more advanced algorithms.

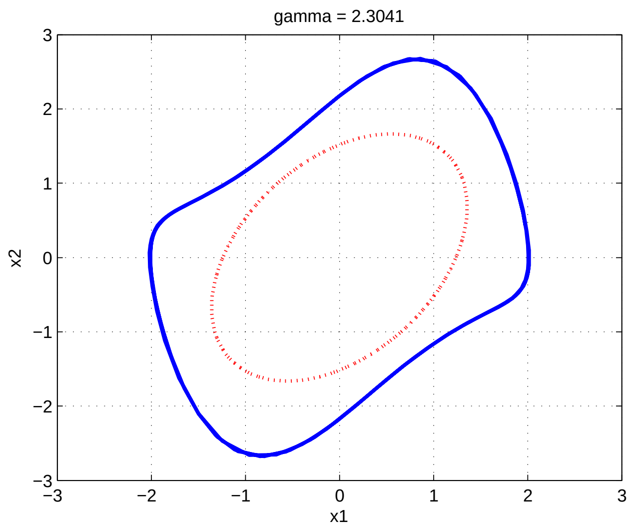

Fig. 2. Van der Pol limit cycle and region of attraction estimate.

V. Conclusions

This paper generalizes sum-of-squares programming by allowing one decision variable to enter bilinearly into the constraints. It is proved that this formulation is quasiconvex and the global optima can be computed via bisection. Many nonlinear analysis problems can be posed within this framework. Software has been developed to quickly solve this particular bisection problem. Algorithms for solving large, sparse generalized eigenvalue optimizations could be applied to solve these generalized SOS programs with significantly less computation than bisection. This will be the subject of future research.

VI. Acknowledgments

This research was partially supported under the NASA Langley NRA contract NNH077ZEA001N entitled "Analytical Validation Tools for Safety Critical Systems" and the NASA Langley NNX08AC65A contract entitled 'Fault Diagnosis, Prognosis and Reliable Flight Envelope Assessment." The technical contract monitors are Dr. Christine Belcastro and Dr. Suresh Joshi, respectively.

References

[1] G. Balas, R. Chiang, A. Packard, and M. Safonov. *Robust Control Toolbox*. The MathWorks, Inc., 2009.
[2] G.J. Balas, A. Packard, P. Seiler, and U. Topcu. Robustness analysis of nonlinear systems. http://www.aem.umn.edu/∼AerospaceControl/, 2009.
[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, 1994.
[4] S. Boyd and L. El Ghaoui. Method of centers for minimizing generalized eigenvalues. *Linear Algebra and Its Applications*, 188:63–111, 1993.
[5] H.-D. Chiang and J.S. Thorp. Stability regions of nonlinear dynamical systems: A constructive methodology. *IEEE Transactions on Automatic Control*, 34(12):1229–1241, 1989.
[6] M.D. Choi, T.Y. Lam, and B. Reznick. Sums of squares of real polynomials. *Proc. of Symposia in Pure Mathematics*, 58(2):103–126, 1995.
[7] E.J. Davison and E.M. Kurak. A computational method for determining quadratic Lyapunov functions for nonlinear systems. *Automatica*, 7:627–636, 1971.
[8] M. Fukuda and M. Kojima. Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem. *Computational Optimization and Applications*, 19(1):79–105, 2001.
[9] K. Gatermann and P. Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192:95–128, 2004.
[10] R. Genesio, M. Tartaglia, and A. Vicino. On the estimation of asymptotic stability regions: State of the art and new proposals. *IEEE Transactions on Automatic Control*, 30(8):747–755, 1985.
[11] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.
[12] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
[13] J. Lofberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.
[14] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, 1994.
[15] D. Noll, M. Torki, and P. Apkarian. Partially augmented Lagrangian method for matrix inequality constraints. *SIAM J. Optim.*, 15(1):161–184, 2004.
[16] P. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.
[17] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming Ser. B*, 96(2):293–320, 2003.
[18] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Algebra*, 127:99–104, 1998.
[19] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.
[20] B. Reznick. Extremal PSD forms with few terms. *Duke Mathematical Journal*, 45(2):363–374, 1978.
[21] B. Reznick. Some concrete aspects of Hilberts 17th problem. *Contemporary Mathematics*, 253(251-272), 2000.
[22] J. Sturm. SeDuMi version 1.05. http://fewcal.kub.nl/sturm/software/sedumi.html, 2001.
[23] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, pages 625–653, 1999.
[24] W. Tan. *Nonlinear Control Analysis and Synthesis using Sum-of-Squares Programming*. PhD thesis, University of California, Berkeley, 2006.
[25] W. Tan, U. Topcu, P. Seiler, G. Balas, and A. Packard. Simulation-aided reachability and local gain analysis for nonlinear dynamical systems. In *Proc. of the IEEE Conference on Decision and Control*, pages 4097–4102, 2008.
[26] B. Tibken. Estimation of the domain of attraction for polynomial systems via LMIs. In *Proc. of the IEEE Conference on Decision and Control*, pages 3860–3864, 2000.
[27] B. Tibken and Y. Fan. Computing the domain of attraction for polynomial systems via BMI optimization methods. In *Proc. of the American Control Conference*, pages 117–122, 2006.
[28] U. Topcu. *Quantitative Local Analysis of Nonlinear Systems*. PhD thesis, University of California, Berkeley, 2008.
[29] U. Topcu, A. Packard, and P. Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.
[30] U. Topcu, A. Packard, P. Seiler, and T. Wheeler. Stability region analysis using simulations and sum-of-squares programming. In *Proc. of the American Control Conference*, pages 6009–6014, 2007.
[31] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.
[32] A. Vannelli and M. Vidyasagar. Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1):69–80, 1985.
[33] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, 2nd edition, 1993.