Analytical Validation Tools for Safety Critical Systems

Peter Seiler* and Gary Balas†

Department of Aerospace Engineering & Mechanics, University of Minnesota , Minneapolis, MN, 55455, USA

Andrew Packard‡

Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA

Ufuk Topcu§

Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, 91125, USA

The current practice to validate flight control laws relies on applying linear analysis tools to assess the closed loop stability and performance characteristics about many trim conditions. Nonlinear simulations are then used to provide further confidence in the linear analyses and also to uncover dynamic characteristics, e.g. limit cycles, which are not revealed by the linear analysis. This paper reviews analysis techniques which can be applied to nonlinear systems described by polynomial dynamic equations. The proposed approach is to reduce the analysis problems to a sum-of-squares optimization problem which can then be solved with freely available software. These techniques can fill the gap between linear analysis and nonlinear simulations and hence can be used to provide additional confidence in the flight control law performance.

I. Introduction

The current practice to validate flight control laws relies on applying linear analysis tools to assess the closed loop stability and performance characteristics about many trim conditions. Nonlinear simulations are then used to provide further confidence in the linear analyses and also to uncover dynamic characteristics, e.g. limit cycles, which are not revealed by the linear analysis. This approach is well-suited for validation of current commercial and military aircraft. However, there are drawbacks of this approach. First, the process is rather time-consuming and requires many well-trained control and simulation engineers. Second, most adaptive control laws lead to nonlinear, time-varying closed loop dynamics. Thus the current practice is not applicable to validating systems with adaptive control laws. There is a need for analytical tools to assess the performance of nonlinear feedback systems.

This paper reviews an approach to reformulate nonlinear analysis problems into a form which can be solved using freely available software. The approach is applicable to nonlinear systems described by polynomial dynamics and it relies on connections between sums-of-squares (SOS) polynomials and positive semidefinite matrices. A polynomial p is a sum of squares if it can be expressed as $p = \sum_{i=1}^{m} f_i^2$. This connection was made in the work by Parrilo[1,2] and has led to research on computational tools for estimating regions of attraction, reachability sets, input-output gains, and robustness with respect to uncertainty. The reader is referred to[15–32] and the references contained therein. There are two key ideas in this approach. First, sufficient conditions for many nonlinear analysis problems can be formulated as set containment conditions involving either a Lyapunov function or a storage function. Second, the set containment conditions can be reformulated as polynomial non-negativity conditions using a generalized version of the S-procedure.[6] This approach will be described in more detail for $L_2 - L_2$ input-output gain analysis.

We envision these techniques as filling the gap between linear analysis and nonlinear simulations. Linearized analysis is only valid over an infinitesimally small neighborhood of the equilibrium point/null input.

*Senior Research Associate: seiler@aem.umn.edu.

†Professor: balas@aem.umn.edu.

‡Professor: pack@me.berkeley.edu.

§Postdoctoral Scholar: utopcu@cds.caltech.edu.

The proposed approach provides an improvement over linearized analysis in that the results are valid over a provable region of the state/input space.[32] Moreover, the nonlinear analysis tools can complement the linear analysis tools and nonlinear simulations to provide additional confidence in the flight control law performance.

The remainder of the paper has the following outline. In the next section, we provide a brief review of background material including SOS polynomials, their connections to positive semidefinite matrices, and SOS programming problems. In Section III we derive an upper bound for the $L_2 - L_2$ input-output gain of a nonlinear system in terms of an optimization problem involving SOS constraints. We also discuss several computational approaches to solve this optimization problem. In Section IV, the approach is applied to compute bounds on the $L_2 - L_2$ gain for a simple model-reference adaptive control system. Conclusions are then given in Section V.

II. Sum of Squares Optimization

In this section we provide a brief review of computational methods for sum-of-squares polynomial optimizations. Briefly, a polynomial p is a *sum of squares* (SOS) if there exist polynomials $\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. Sum-of-squares programs are optimization problems involving sum-of-squares polynomial constraints. As discussed further in Section III, many nonlinear analysis problems can be posed within this optimization framework. The computational solutions to these problems rely on connections between semidefinite matrices and SOS polynomials.[1–3] In this section we first present notation and background material. Next we discuss the connections between semidefinite matrices and SOS polynomials. Finally we discuss software available to solve SOS optimization problems.

A. Background

1. *Polynomial Notation*

$\mathbb{R}[x]$ denotes the set of all polynomials in variables $\{x_1, \ldots, x_n\}$ with real coefficients. $\mathbb{N}$ 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 $\deg x^\alpha \doteq \sum_{i=1}^n \alpha_i$. In this notation a polynomial in $\mathbb{R}[x]$ is simply 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 $\mathbb{N}^n$. Using the definition of deg for a monomial, the degree of p is defined as $\deg p \doteq \max_{\alpha \in \mathcal{A}} \ [\deg x^\alpha]$.

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 as $\Sigma[x]$. We note that 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. This is related to one of the problems posed by Hilbert in 1900.[4]

2. *Semidefinite Programming*

This brief review of semidefinite programming (SDP) is based on a survey by Vandenberghe and Boyd[5] and a monograph by Boyd, et al.[6] A symmetric matrix $F \in \mathbb{R}^{n \times n}$ is positive semidefinite if $x^T F x \geq 0$ for all $x \in \mathbb{R}^n$. Positive semidefinite matrices are denoted by $F \succeq 0$. A semidefinite program is an optimization problem of the following form:

$$\begin{array}{ll} \min_\lambda & c^T \lambda \\ \text{subject to:} & F_0 + \sum_{k=1}^r \lambda_k F_k \succeq 0 \end{array} \tag{1}$$

The symmetric matrices $F_0, \ldots, F_r \in \mathbb{R}^{n \times n}$ and the vector $c \in \mathbb{R}^r$ are given data. The vector $\lambda \in \mathbb{R}^r$ is the decision variable and the constraint, $F_0 + \sum_{k=1}^r \lambda_k F_k \succeq 0$, is called a linear matrix inequality. We refer to

Equation 1 as the primal problem. The dual associated with this primal problem is:

$$\begin{array}{ll} \max_Z & -\mathbf{Tr}\,[F_0 Z] \\ \text{subject to:} & \mathbf{Tr}\,[F_k Z] = c_k \quad k = 1, \ldots, r \\ & Z \succeq 0 \end{array} \tag{2}$$

where $Z = Z^T \in \mathbb{R}^{n \times n}$ 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. While the primal and dual forms may look restrictive, these formulations are quite versatile and SDPs find applications in many problems of interest. Moreover, SDPs are convex and quality software exists to solve these problems. In particular, SeDuMi[7,8] is a freely available MATLAB toolbox that simultaneously solves the primal and/or dual forms of a semidefinite program.

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

$$\text{Find } \lambda_1, \ldots, \lambda_r \in \mathbb{R} \text{ such that } F_0 + \sum_{k=1}^{r} \lambda_k F_k \succeq 0 \tag{3}$$

B. Connections Between SOS Polynomials and Semidefinite Matrices

Theorem 1 below gives a concrete statement of the connection between sums of squares and positive semidefinite matrices. We require two facts that follow from[9] (refer to Theorem 1 and its preceding Lemma):

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 $\deg f_i \le d \;\forall i$.

Next, we define z as the column vector of all monomials in variables $\{x_1, \ldots, x_n\}$ of degree $\le d$: [a]

$$z \doteq \left[1,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_1^2,\ x_1x_2,\ \ldots,\ x_n^2,\ \ldots,\ x_n^d\right]^T \tag{4}$$

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 $\le d$, then f is a finite linear combination of monomials of degree $\le d$. Consequently, there exists $a \in \mathbb{R}^{l_z}$ such that $f = a^T z$. The proof of the following theorem, introduced as a "Gram Matrix" method by Choi, Lam, and Reznick,[10] is included for completeness. This result can be found more recently in.[11]

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 4. 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:

($\Rightarrow$) If p is a SOS, then there exists polynomials $\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. As noted 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, $p = z^T(AA^T)z = \sum_{i=1}^m f_i^2$. ■

[a] Any ordering of the monomials can be used to form z. In Equation 4, x^α precedes x^β in the definition of z if:

$$\deg x^\alpha < \deg x^\beta \quad \text{or} \quad \deg x^\alpha = \deg x^\beta \text{ and the first nonzero entry of } \alpha - \beta \text{ is } > 0$$

C. Software for SOS Optimizations

A sum-of-squares program is an optimization problem with a linear cost and SOS constraints on the decision variables:[12]

$$
\begin{aligned}
&\min_{u\in\mathbb{R}^n} c_1u_1 + \cdots + c_nu_n \\
&\text{subject to:} \\
&a_{k,0}(x) + a_{k,1}(x)u_1 + \cdots + a_{k,n}(x)u_n \in \Sigma[x] \;\; k = 1,\ldots N_s
\end{aligned} \tag{5}
$$

The polynomials $\{a_{k,j}\}$ are given as part of the optimization data and $u \in \mathbb{R}^n$ are decision variables. This formulation appears far removed from dynamical systems but we'll show in Section III that nonlinear analysis problems can be posed within this optimization framework.

Theorem 1 provides the bridge to convert an SOS program into a semidefinite-programming problem. For example, the constraint $a_{k,0}(x) + a_{k,1}(x)u_1 + \cdots + a_{k,n}(x)u_n \in \Sigma[x]$ can be equivalently written as:

$$a_{k,0}(x) + a_{k,1}(x)u_1 + \cdots + a_{k,n}(x)u_n = z^TQz \tag{6}$$

$$Q \succeq 0 \tag{7}$$

Q is a new matrix of decision variables that is introduced when we convert an SOS constraint to an LMI constraint. Equating the coefficients of z^TQz and $a_{k,0}(x)+a_{k,1}(x)u_1+\cdots+a_{k,n}(x)u_n$ imposes linear equality constraints on the decision variables u and Q. Thus, Equation 6 can be rewritten as a set of linear equality constraints on the decision variables. All SOS constraints in Equation 5 can be replaced in this fashion with linear equality constraints and LMI constraints. As a result, the SOS program in Equation 5 can be written in the SDP dual form (Equation 2).

While this may appear cumbersome, there is software available to perform the conversion. For example, SOSTOOLS[12] and Yalmip[13] are freely available MATLAB toolboxes for solving SOS optimizations. These packages allow the user to specify the polynomial constraints using a symbolic toolbox. Then they convert the SOS optimization into an SDP which is solved with SeDuMi[7,8] or another freely available SDP solver. Finally these toolboxes convert the solution of the SDP back to a polynomial solution. A drawback is that the size of the resulting SDP grows rapidly if the SOS optimization involves polynomials with many variables and/or high degree. While various techniques can be used to exploit the problem structure,[14] this computational growth is a generic trend in SOS optimizations. We have developed some methods which use simulation to ease this computational growth.[15–17]

III. Nonlinear Analysis Tools

Many nonlinear analysis problems can be formulated as sum of squares programming problems. This connection was made in the work by Parrilo[1,2] and has led to research on computational tools for estimating regions of attraction, reachability sets, input-output gains, and robustness with respect to uncertainty. The reader is referred to[15–32] and the references contained therein. The key idea is that sufficient conditions for these nonlinear analysis problems can typically be formulated as set containment conditions involving either a Lyapunov function or a storage function. The set containment conditions can be reformulated as polynomial non-negativity conditions using a generalized version of the S-procedure.[6] These problems can then be solved as SOS programs since SOS polynomials are globally non-negative. In this section we'll demonstrate this approach for an $L_2 - L_2$ input-output gain problem. More details for this input-output gain problem can be found in.[28,31,32] The formulations for the other analysis problems can be found in the references given above.

We consider nonlinear dynamical systems of the form:

$$
\begin{aligned}
\dot{x} &= f(x,u) \\
y &= h(x)
\end{aligned} \tag{8}
$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input, and $y \in \mathbb{R}^{n_y}$ is the output. We assume f is a $n_x \times 1$ polynomial function of x and u such that $f(0,0) = 0$. We also assume that h is an $n_y \times 1$ polynomial function of x such that $h(0) = 0$. We'll denote this system by $\mathcal{S}$.

Define the L_2 norm of a signal as $\|u\|_2 := \left[\int_0^\infty u^T(t)u(t)dt\right]^{0.5}$. u is called an L_2 signal if this integral is finite and we'll assume all inputs to $\mathcal{S}$ are L_2 signals. The L_2-L_2 input-output gain of the system is defined as $\|\mathcal{S}\| := \sup_{\|u\|_2 \neq 0} \frac{\|y\|_2}{\|u\|_2}$. We can also define a "local" input-output gain of the system as $\|\mathcal{S}\|_R := \sup_{0<\|u\|_2 \leq R} \frac{\|y\|_2}{\|u\|_2}$. For a linear system, the magnitude of the output scales proportionally with the magnitude of the input and so the ratio $\frac{\|y\|_2}{\|u\|_2}$ does not depend on $\|u\|_2$. Thus $\|\mathcal{S}\|_R = \|\mathcal{S}\|$ for all $R > 0$. For a nonlinear system, the local gain depends on the magnitude of the input and hence $\|\mathcal{S}\|_R$ and $\|\mathcal{S}\|$ need not be equal. The class of possible inputs increases with increasing values of R and so we can conclude that $\|\mathcal{S}\|_R$ is a monotonically increasing function of R and $\|\mathcal{S}\|_R \leq \|\mathcal{S}\|$ for all $R > 0$.

Lemma 1 provides a sufficient condition for the local L_2-L_2 input-output gain to be less than γ. This specific lemma can be found in[28, 31, 32] but similar results are given in textbooks.[33, 34]

Lemma 1 *If there exists a $\gamma > 0$ and a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ such that:*

- $V(0) = 0$ *and* $V(x) \geq 0 \;\; \forall x \in \mathbb{R}^{n_x}$
- $\{(x,u) \in \mathbb{R}^{n_x+n_u} \; : \; V(x) \leq R^2\} \subseteq \{(x,u) \in \mathbb{R}^{n_x+n_u} \; : \; \frac{\partial V}{\partial x} f(x,u) \leq u^T u - \gamma^{-2} y^T y\}$

then $x(0) = 0$ and $\|u\|_2 \leq R$ implies $\|y\|_2 \leq \gamma \|u\|_2$.

Proof:

We provide a sketch of the proof. Assume that $\frac{\partial V}{\partial x} f(x,u) \leq u^T u - \gamma^{-2} y^T y$ holds along the trajectories of the system $\mathcal{S}$ from time 0 to T. Integrating then yields:

$$V(x(T)) - V(x(0)) \leq \int_0^T \left(u^T u - \gamma^{-2} y^T y\right) dt \tag{9}$$

If $x(0) = 0$ and $\|u\|_2 \leq R$ then this implies that $V(x(T)) \leq \|u\|_2^2 \leq R^2$. Thus the state trajectories satisfy $V(x(T)) \leq R^2$ for all time T and it is valid to assume $\frac{\partial V}{\partial x} f(x,u) \leq u^T u - \gamma^{-2} y^T y$ holds along the system trajectories. Moreover, Equation 9 implies that $\int_0^T \left(y^T y\right) dt \leq \gamma^2 \int_0^T \left(u^T u\right) dt$ since $V(x) \geq 0 \; \forall x$. Letting $T \to \infty$, we conclude that $\|y\|_2 \leq \gamma \|u\|_2$. ■

Lemma 1 provides a sufficient condition to prove $\|S\|_R \leq \gamma$ in terms of a storage function, V. This lemma involves one non-negativity condition on the storage function and a set containment condition. The next Lemma provides a sufficient condition for set containment in terms of function non-negativity constraints. This lemma is a generalization of the S-procedure which has been frequently applied in control theory.[6] The function s appearing in the Lemma is called a multiplier.

Lemma 2 *Define two sets $A := \{x \in R^n \; : \; f_A(x) \geq 0\}$ and $B := \{x \in R^n \; : \; f_B(x) \geq 0\}$. If there exists a function $s(x) \geq 0 \; \forall x$ such that $f_B(x) - f_A(x)s(x) \geq 0 \; \forall x$ then $A \subseteq B$.*

Proof:

Assume there exists a function $s(x) \geq 0 \; \forall x$ such that $f_B(x) - f_A(x)s(x) \geq 0 \; \forall x$. Take any $x \in A$. Then $f_B(x) \geq f_A(x)s(x) \geq 0$. Thus x is also in B. ■

We can now formulate a sum-of-squares program which provides an upper bound on the local L_2-L_2 gain:

$$\gamma^* := \min_{V,s,\gamma} \gamma \tag{10}$$

subject to:

$$s(x,u) \in \Sigma[x,u], V(x) \in \Sigma[x], V(0) = 0 \tag{11}$$

$$u^T u - \gamma^{-2} h(x)^T h(x) - \frac{\partial V(x)}{\partial x} f(x,u) - s(x,u)\left(R^2 - V(x)\right) \in \Sigma[x,u] \tag{12}$$

The constraint in Equation 12, if satisfied, ensures that $\{(x,u) \in \mathbb{R}^{n_x+n_u} \; : \; V(x) \leq R^2\} \subseteq \{(x,u) \in \mathbb{R}^{n_x+n_u} \; : \; \frac{\partial V}{\partial x} f(x,u) \leq u^T u - \gamma^{-2} y^T y\}$. Since SOS polynomials are non-negative everywhere this follows by applying the generalized S-procedure in Lemma 2. We can then apply Lemma 1 to conclude that $\|S\|_R \leq \gamma$ for any γ for which the constraints are valid. γ^* is the smallest upper bound on $\|S\|_R$ which can be found with this sufficient condition.

This optimization problem involves SOS constraints on $s(x,u)$ and $V(x)$ (Equation 11). The coefficients of the polynomials $s(x,u)$ and $V(x)$ are decision variables which the optimization can choose to try to minimize

γ. The constraint in Equation 12 is simply an SOS constraint on a polynomial of x and u. Unfortunately this constraint is bilinear in the decision variables since it involves a term of the form $s(x, u) \cdot V(x)$. This makes the computational problem substantially more difficult. This can be solved directly using bilinear matrix inequality solvers.[16] Alternatively one can note that if either s or V is held fixed then it is possible to express this optimization problem in the form of a standard SOS programming problem (Equation 5) and it can be solved using SOS programming software.[12, 13] Thus another solution method is to iterate back and forth solving for either V or s while the other variable is held fixed. This iteration can be initialized with V fixed at the quadratic storage function obtained from linear analysis.[21] A final method is to use simulation data to construct a candidate V and then perform this iteration.[16]

The computational tools for other nonlinear problems (estimating regions of attraction, reachability sets, input-output gains with other signal norms, and robustness with respect to uncertainty) all essentially follow the same steps as used for computing a bound on the local L_2-L_2 gain. Specifically, a Lyapunov or storage function type theorem is used to derive a sufficient condition for the nonlinear system to have a particular performance/stability property. Lyapunov and storage functions are naturally restricted to be positive definite and this can be enforced using SOS constraints. Additional conditions can typically be formulated as set containment conditions. These set containment conditions can then be converted into function non-negativity constraints using the generalized S-procedure. Since SOS polynomials are non-negative everywhere, the non-negativity constraints can be relaxed and written as SOS constraints. In many cases this sequence of constraint reformulations leads to either a linear or bilinear SOS programming problem which yields a bound on a particular systems property (e.g. inner approximations to regions of attraction or upper bounds on system gains). Bilinear problems can be solved using one of the methods described above. Simulations or gradient searches can be used to compute dual bounds (e.g. outer approximations to regions of attraction or lower bounds on system gains). For example, lower bounds on the local gain can be computed using a power method derived for a finite horizon optimal control problem.[35] This approach provides an improvement over linearized analysis in that the results are valid over a provable region of the state/input space rather than for an infinitesimally small neighborhood of the equilibrium point/null input. Further details on this statement can be found in.[32]

IV. Adaptive Control Example

To demonstrate the computational tools we consider the following single state system:

$$\begin{aligned} \dot{x} &= -x + w + u \\ y &= -1.8x + w + u \end{aligned}$$

$x \in \mathbb{R}$ is the plant state, $u \in \mathbb{R}$ is the control input, $y \in \mathbb{R}$ is the output, and $w \in \mathbb{R}$ is a disturbance which impacts both the state and the output. The following model-reference adaptive controller will be used for reference tracking:

$$\begin{aligned} \dot{x}_m &= -x_m + r \\ \dot{z}_x &= -x^2 + x x_m \\ \dot{z}_r &= -xr + x_m r \\ u &= (1 + z_r) r + z_x x \end{aligned}$$

x_m is the reference model state, r is the reference signal, and z_x and z_r are feedback gains which are tuned by the adaptation. Figure 1 shows lower and upper bounds on $\|S\|_R$. The upper bounds (blue curves) are computed as described in the previous section with deg(V)=2 and deg(V)=4. Simulation data is used to construct a candidate storage function V and then an iteration is performed successively holding V or s fixed.[16, 31] The green curves are a slightly better upper bound using the refinement technique described in.[28, 29] The refinement only yields a minor improvement on the upper bounds. The lower bound (red curve) is computed using the power method technique described in.[35] There is a significant gap between the lower and upper bounds which indicates that improvement in one or both bounds is possible. Increasing the degree of the storage function should improve the upper bound at the cost of additional computation.

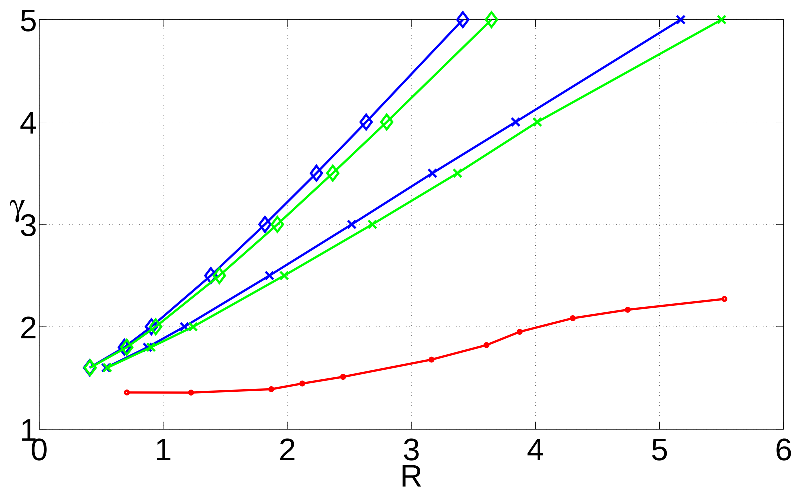

Figure 1. Upper bounds on $\|S\|_R$ for deg(V) = 2 (with $\diamond$) and deg(V) = 4 (with $\times$) before the refinement (blue curves) and after the refinement (green curves) along with the lower bounds (red curve).

V. Conclusion

In this paper we described the connections between sums-of-squares optimizations and analysis problems for nonlinear polynomial systems. In particular, a SOS optimization was derived to compute upper bounds on the L_2 to L_2 gain of nonlinear polynomial system. Many other nonlinear analysis problems can be formulated within this optimization framework. The approach was applied to compute the disturbance to output gain for a simple model-reference adaptive control system. We view these nonlinear analysis tools as filling the gap between linear analyses, which are valid only for infinitesimally small neighborhoods about an equilibrium, and nonlinear simulations. These tools can be used to provide additional confidence when validating the performance of a flight control law. Significant work remains to be done to reduce the computational cost and enable these techniques to be applied to moderate-sized systems (systems with more than ≈ 8 states).

Acknowledgments

This research was partially supported under a NASA Langley NRA NNH077ZEA001N entitled "Analytical Validation Tools for Safety Critical Systems." The technical contract monitor was Dr. Christine Belcastro.

References

[1] Parrilo, P., *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, 2000.

[2] Parrilo, P., "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming Ser. B*, 2003.

[3] Lasserre, J., "Global Optimization with Polynomials and the Problem of Moments," *SIAM Journal on Optimization*, Vol. 11, No. 3, 2001, pp. 796–817.

[4] Reznick, B., "Some concrete aspects of Hilberts 17th problem," *Contemporary Mathematics*, Vol. 253, No. 251-272, 2000.

[5] Vandenberghe, L. and Boyd, S., "Semidefinite Programming," *SIAM Review*, Vol. 38, No. 1, 1996, pp. 49–95.

[6] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Vol. 15 of *Studies in Applied Mathematics*, SIAM, 1994.

[7] Sturm, J., "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, 1999, pp. 625–653.

[8] Sturm, J., "SeDuMi version 1.05," http://fewcal.kub.nl/sturm/software/sedumi.html, 2001.

[9] Reznick, B., "Extremal PSD forms with few terms," *Duke Mathematical Journal*, Vol. 45, No. 2, 1978, pp. 363–374.

[10] Choi, M., Lam, T., and Reznick, B., "Sums of Squares of Real Polynomials," *Proceedings of Symposia in Pure Mathematics*, Vol. 58, No. 2, 1995, pp. 103–126.

[11]Powers, V. and Wörmann, T., "An Algorithm for sums of squares of real polynomials," *Journal of Pure and Applied Algebra*, Vol. 127, 1998, pp. 99–104.

[12]Prajna, S., Papachristodoulou, A., Seiler, P., and Parrilo, P. A., *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.

[13]Lofberg, J., "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[14]Gatermann, K. and Parrilo, P., "Symmetry groups, semidefinite programs, and sums of squares," *Journal of Pure and Applied Algebra*, Vol. 192, 2004, pp. 95–128.

[15]Topcu, U., Packard, A., Seiler, P., and Wheeler, T., "Stability region analysis using simulations and sum-of-squares programming," *Proceedings of the American Control Conference*, 2007, pp. 6009–6014.

[16]Tan, W., Topcu, U., Seiler, P., Balas, G., and Packard, A., "Simulation-aided Reachability and Local Gain Analysis for Nonlinear Dynamical Systems," *Proceedings of the IEEE Conference on Decision and Control*, 2008, pp. 4097–4102.

[17]Topcu, U., Packard, A., and Seiler, P., "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, Vol. 44, No. 10, 2008, pp. 2669–2675.

[18]Tibken, B., "Estimation of the domain of attraction for polynomial systems via LMIs," *Proceedings of the IEEE Conference on Decision and Control*, 2000, pp. 3860–3864.

[19]Hachicho, O. and Tibken, B., "Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments," *Proceedings of the IEEE Conference on Decision and Control*, 2002, pp. 3150–3155.

[20]Jarvis-Wloszek, Z., *Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems using Sum-of-Squares Optimization*, Ph.D. thesis, University of California, Berkeley, 2003.

[21]Jarvis-Wloszek, Z., Feeley, R., Tan, W., Sun, K., and Packard, A., "Some Controls Applications of Sum of Squares Programming," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 5, 2003, pp. 4676–4681.

[22]Tan, W. and Packard, A., "Searching for control Lyapunov functions using sums of squares programming," *42nd Annual Allerton Conference on Communications, Control and Computing*, 2004, pp. 210–219.

[23]Chesi, G., "On the estimation of the domain of attraction for uncertain polynomial systems via LMIs," *Proceedings of the IEEE Conference on Decision and Control*, 2004, pp. 881–886.

[24]Papachristodoulou, A., *Scalable analysis of nonlinear systems using convex optimization*, Ph.D. thesis, California Institute of Technology, 2005.

[25]Jarvis-Wloszek, Z., Feeley, R., Tan, W., Sun, K., and Packard, A., *Positive Polynomials in Control*, Vol. 312 of *Lecture Notes in Control and Information Sciences*, chap. Controls Applications of Sum of Squares Programming, Springer-Verlag, 2005, pp. 3–22.

[26]Chesi, G., Garulli, A., Tesi, A., and Vicino, A., "LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems," *International Journal of Robust and Nonlinear Control*, Vol. 15, 2005, pp. 35–49.

[27]Prajna, S., *Optimization-Based Methods for Nonlinear and Hybrid Systems Verification*, Ph.D. thesis, California Institute of Technology, 2005.

[28]Tan, W., Packard, A., and Wheeler, T., "Local gain analysis of nonlinear systems," *Proceedings of the American Control Conference*, pp. 92–96.

[29]Tan, W., *Nonlinear Control Analysis and Synthesis using Sum-of-Squares Programming*, Ph.D. thesis, University of California, Berkeley, 2006.

[30]Tibken, B. and Fan, Y., "Computing the domain of attraction for polynomial systems via BMI optimization methods," *Proceedings of the American Control Conference*, 2006, pp. 117–122.

[31]Topcu, U., *Quantitative Local Analysis of Nonlinear Systems*, Ph.D. thesis, University of California, Berkeley, 2008.

[32]Topcu, U. and Packard, A., "Linearized analysis versus optimization-based nonlinear analysis for nonlinear systems," *submitted to the 2009 American Control Conference*, 2009.

[33]Vidyasagar, M., *Nonlinear Systems Analysis*, Prentice Hall, 2nd ed., 1993.

[34]Khalil, H., *Nonlinear Systems*, Prentice Hall, 3rd ed., 2002.

[35]J. Tierno, R. Murray, J. D. I. G., "Numerically efficient robustness analysis of trajectory tracking for nonlinear systems," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, 1997, pp. 640–647.