Some Controls Applications of Sum of Squares Programming[1]

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun and Andrew Packard
Department of Mechanical Engineering, University of California, Berkeley
{zachary, rfeeley, weehong, kpsun, pack}@jagger.me.berkeley.edu

***Abstract*— We consider nonlinear systems with polynomial vector fields and present two algorithms based on sum of squares programming, that may answer system theoretic questions. The first algorithm provides a bound for the reachable set of a system driven by a unit-energy disturbance, while the second synthesizes a polynomial state feedback controller to enlarge the provable region of attraction. We also outline a variant of the second algorithm for handling systems with input saturation. Both algorithms are demonstrated using two-state nonlinear systems.**

I. Introduction

Recent developments in sum of squares (SOS) programming [1], [2] have greatly extended the class of problems that can be solved with convex optimization. These results provide a general methodology to find formulations or relaxations, solvable by semidefinite programming, which address seemingly intractable nonconvex problems. Many of the problems that are amenable to SOS programming relate to polynomial optimization or algebraic geometry and reach back to the original work on global lower bounds for polynomials. This work is collected and expanded upon in [3].

First we define the basic tools needed to state the main theorem, the Positivstellensatz, which leads to the development of our results. We use this methodology to pose two system theoretic problems for nonlinear systems with polynomial vector fields. The problems considered are 1) bounding the reachable set subject to unit energy disturbance, 2) expanding a region of attraction with state feedback, and 2') problem 2, but for systems with input saturation. Finally, we present two proof of concept numerical examples to demonstrate the two algorithms for these problems.

II. Preliminaries

A. Polynomial Definitions

Definition 1 (Monomials): A **Monomial** m_α in n variables is a function defined as $m_\alpha(x) = x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ for $\alpha \in \mathbb{Z}_+^n$. The degree of a monomial is defined, $\deg m_\alpha := \sum_{i=1}^n \alpha_i$.

Definition 2 (Polynomials): A **Polynomial** f in n variables is a finite linear combination of monomials,

$$f := \sum_\alpha c_\alpha m_\alpha = \sum_\alpha c_\alpha x^\alpha$$

with $c_\alpha \in \mathbb{R}$. Define $\mathcal{R}_n$ to be the set of all polynomials in n variables. The degree of f is defined as $\deg f := \max_\alpha \deg m_\alpha$ (provided the associated c_α is non-zero).

Additionally we define Σ_n to be the set of sum of squares (SOS) polynomials in n variables.

$$\Sigma_n := \{p \in \mathcal{R}_n \,|\, p = \sum_{i=1}^t f_i^2 \quad f_i \in \mathcal{R}_n,\, i = 1, \ldots, t\}$$

Obviously if $p \in \Sigma_n$, then $p(x) \geq 0 \;\forall x \in \mathbb{R}^n$.

It is interesting to note that there are polynomials that are positive semidefinite (PSD) that are not sum of squares. In general, there are only three combinations of number of variables and degree such that the set of SOS polynomials is equivalent to the set of positive semidefinite ones, namely, $n = 2$; $d = 2$; and $n = 3$ with $d = 4$. This result dates to Hilbert and is related to his 17th problem.

B. Positivstellensatz

In the section we define concepts to state a central theorem from real algebraic geometry, the Positivstellensatz, which we will hereafter refer to as the P-satz. This is a powerful theorem which generalizes many known results. For example, applying the P-satz, it is possible to derive the $\mathcal{S}$-procedure by carefully picking the free parameters, as will be shown in §II-D.

Definition 3: Given $\{g_1, \ldots, g_t\} \in \mathcal{R}_n$, the **Multiplicative Monoid** generated by g_j's is the set of all finite products of g_j's, including 1 (i.e. the empty product). It is denoted as $\mathcal{M}(g_1, \ldots, g_t)$.

Definition 4: Given $\{f_1, \ldots, f_r\} \in \mathcal{R}_n$, the **Cone** generated by f_i's is

$$\mathcal{P}(f_1, \ldots, f_r) := \left\{ s_0 + \textstyle\sum_{i=1}^l s_i b_i \,|\, l \in \mathbb{Z}_+, s_i \in \Sigma_n, b_i \in \mathcal{M}(f_1, \ldots, f_r) \right\}.$$

Note that if $s \in \Sigma_n$ and $f \in \mathcal{R}_n$, then $f^2 s \in \Sigma_n$ as well. This allows us to express a cone of $\{f_1, \ldots, f_r\}$ as a sum of 2^r terms. An example: $\mathcal{P}(f_1, f_2) = \{s_0 + s_1 f_1 + s_2 f_2 + s_3 f_1 f_2 \,|\, s_0, \ldots, s_3 \in \Sigma_n\}$

Definition 5: Given $\{h_1, \ldots, h_u\} \in \mathcal{R}_n$, the **Ideal** generated by h_k's is

$$\mathcal{I}(h_1, \ldots, h_u) := \left\{ \sum h_k p_k \,|\, p_k \in \mathcal{R}_n \right\}.$$

With these definitions we can state the following theorem taken from [4, Theorem 4.2.2]

[1]The authors would like to thank the following for providing support for this project: DARPA's Software Enabled Control Program under USAF contract #F33615-99-C-1497, the NSF under contract #CTS-0113985, and DSO National Laboratories-Singapore.

Theorem 1 (Positivstellensatz): Given polynomials $\{f_1,\ldots,f_r\}$, $\{g_1,\ldots,g_t\}$, and $\{h_1,\ldots,h_u\}$ in $\mathcal{R}_n$, the following are equivalent:

1) The set
$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} f_1(x) \geq 0, \ldots, f_r(x) \geq 0 \\ g_1(x) \neq 0, \ldots, g_t(x) \neq 0 \\ h_1(x) = 0, \ldots, h_u(x) = 0 \end{array} \right\}$$
is empty.
2) There exist polynomials $f \in \mathcal{P}(f_1,\ldots,f_r)$, $g \in \mathcal{M}(g_1,\ldots,g_t)$, $h \in \mathcal{I}(h_1,\ldots,h_u)$ such that
$$f + g^2 + h = 0.$$

When there are only inequality constraints, and they describe a compact region, this theorem can be improved to reduce the number of free parameters [5], and with slightly stronger assumptions [6]. These results have been used to improve bounds on nonconvex polynomial optimization [2] and [7] highlighted a software package to do so.

C. SOS Programming

Sum of squares polynomials play an important role in the P-satz. Using a "Gram matrix" approach, Choi et al. [8] showed that $p \in \Sigma_n$ iff $\exists\ Q \succeq 0$ such that $p(x) = z^*(x)Qz(x)$, with $z(x)$ a vector of suitable monomials. Powers and Wörmann [9] proposed an algorithm to check if any $Q \succeq 0$ exists for a given $p \in \mathcal{R}_n$. Parrilo [1] showed that their algorithm is an LMI, and proved the following extension.

Theorem 2 (Parrilo): Given a finite set $\{p_i\}_{i=0}^m \in \mathcal{R}_n$, the existence of $\{a_i\}_{i=1}^m \in \mathbb{R}$ such that
$$p_0 + \sum_{i=1}^{m} a_i p_i \in \Sigma_n$$
is an LMI feasibility problem.

This theorem is useful since it allows one to answer questions like the following SOS programming example.

Example 1: Given $p_0, p_1 \in \mathcal{R}_n$, does there exist a $k \in \mathcal{R}_n$, of a given degree, such that
$$p_0 + kp_1 \in \Sigma_n. \tag{1}$$
To answer this question, write k as a linear combination of its monomials $\{m_j\}$, $k = \sum_{j=1}^s a_j m_j$. Rewrite (1) using this decomposition
$$p_0 + kp_1 = p_0 + \sum_{j=1}^{s} a_j(m_j p_1)$$
which since $(m_j p_1) \in \mathcal{R}_n$ is a feasibility problem that can be checked by Theorem 2.

A software package, SOSTOOLS, [10], [11], exists to aid in solving the LMIs that result from Theorem 2. This package as well as [7] use Sturm's SeDuMi semidefinite programming solver [12].

D. $\mathcal{S}$-Procedure

What does the $\mathcal{S}$-procedure look like in the P-satz formalism? Given symmetric $n \times n$ matrices $\{A_k\}_{k=0}^m$, the $\mathcal{S}$-procedure states: if there exist nonnegative scalars $\{\lambda_k\}_{k=1}^m$ such that $A_0 - \sum_{k=1}^m \lambda_k A_k \succeq 0$, then
$$\bigcap_{k=1}^{m} \left\{ x \in \mathbb{R}^n | x^T A_k x \geq 0 \right\} \subseteq \left\{ x \in \mathbb{R}^n | x^T A_0 x \geq 0 \right\}.$$
Written in P-satz form, the question becomes "is
$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} x^T A_1 x \geq 0, \ldots, x^T A_m x \geq 0, \\ -x^T A_0 x \geq 0,\, x^T A_0 x \neq 0 \end{array} \right\}$$
empty?" Certainly, if the λ_k exist, define $0 \preceq Q := A_0 - \sum_{k=1}^m \lambda_k A_k$. Further define SOS functions $s_0(x) := x^T Q x, s_{01} := \lambda_1, \ldots, s_{0m} := \lambda_m$. Note that
$$\begin{array}{rcl} f & := & (-x^T A_0 x)s_0 + \sum_{k=1}^m (-x^T A_0 x)(x^T A_k x)s_{0k} \\ & \in & \mathcal{P}\left(x^T A_1 x, \ldots, x^T A_m x, -x^T A_0 x\right) \end{array}$$
and that $g := x^T A_0 x \in \mathcal{M}(x^T A_0 x)$. Substitution yields $f + g^2 = 0$ as desired. We will use this insight to make specific selections in the P-satz formulation of the state-feedback problem considered in Section IV. For the special case of $m = 1$, the converse of the $\mathcal{S}$-Procedure is also true [13, §2.6.3].

III. Bounding the Reachable Set Subject to Unit Energy Disturbances

Using the tools of SOS programming and the P-satz, we can, after some simplifications, cast some control problems for systems with polynomial vector fields as tractable optimization problems. In this and the following section, we discuss two system theoretic problems these techniques are applicable to.

Given a system of the form
$$\dot{x} = f(x) + g_w(x)w \tag{2}$$
with $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}$, and f, g_w n-vectors with elements of $\mathcal{R}_n$ such that $f(0) = 0$, compute a bound on the set of points $x(T)$ that are reachable from $x(0) = 0$ under (2) provided the disturbance satisfies $\int_0^T w^2(t)\,dt \leq 1$, $T \geq 0$.

A similar problem is considered in [14], where real quantifier elimination is used to calculate the exact reachable set for a larger class of dynamical systems. Our approach only considers convex relaxations of the exact problem, and as such requires less computation. A comparison of SOS programming and computational algebra is given in [15] for the case of polynomial minimization.

Following the Lyapunov-like argument in [13, §6.1.1], if we have a polynomial V such that
$$V(x) > 0 \text{ for all } x \in \mathbb{R}^n \backslash \{0\} \text{ with } V(0) = 0, \text{ and} \tag{3}$$
$$\frac{\partial V}{\partial x}(f(x) + g_w(x)w) \leq w^2 \text{ for all } x \in \mathbb{R}^n, w \in \mathbb{R}, \tag{4}$$

then $\{x|V(x) \leq 1\}$ contains the set of points $x(T)$ that are reachable from $x(0) = 0$ for any w such that $\int_0^T w(t)^2\,dt \leq 1$, $T \geq 0$. We can see this by integrating the inequality in (4) from 0 to T, yielding

$$V(x(T)) - V(x(0)) = \int_0^T w(t)^2\,dt \leq 1.$$

Recalling $V(x(0)) = 0$, $x(T) \in \{x|V(x) \leq 1\}$. Furthermore, $x(\tau) \in \{x|V(x) \leq 1\}$ for all $\tau \in [0,T]$, allowing us to relax the inequality in (4) to

$$\frac{\partial V}{\partial x}(f(x) + g_w(x)w) \leq w^2 \quad \forall x \in \{x|V(x) \leq 1\}, \forall w \in \mathbb{R}.$$

To bound the reachable set, we require a V satisfying these conditions. Additionally, to achieve a useful bound, the level set $\{x|V(x) \leq 1\}$ should be as small as possible. This is accomplished by requiring $\{x|V(x) \leq 1\}$ to be contained in a variable sized region $P_\beta := \{x \in \mathbb{R}^n|p(x) \leq \beta\}$, for some positive definite p, and minimizing β under the constraint that we can find a V satisfying (3) and (4). Restricting V to be a polynomial with no constant term, so that $V(0) = 0$, we formulate the problem in the following way, leading to application of the P-satz.

$$\min_{V \in \mathcal{R}_n} \beta$$

such that

$$\{x \in \mathbb{R}^n|V(x) \leq 0, l_1(x) \neq 0\} \text{ is empty} \tag{5}$$

$$\{x \in \mathbb{R}^n|V(x) \leq 1, p(x) \geq \beta, p(x) \neq \beta\} \text{ is empty} \tag{6}$$

$$\left\{ \begin{array}{c} x \in \mathbb{R}^n \\ w \in \mathbb{R} \end{array} \;\middle|\; \begin{array}{c} V(x) \leq 1, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \geq w^2, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \neq w^2 \end{array} \right\} \text{ is empty} \tag{7}$$

where l_1 is some positive definite and SOS polynomial that replaces x in the non-polynomial constraint $x \neq 0$. The constraints (5) and (7) make V and $\dot{V}$ behave properly, while (6) allows that $\{x|V(x) \leq 1\} \subseteq P_\beta$.

Invoking the P-satz, constraints (5)–(7) are equivalent to the constraints in the following minimization.

$$\min \beta \quad \text{over} \quad \begin{array}{c} V \in \mathcal{R}_n \quad s_1, \ldots, s_6 \in \Sigma_n \\ s_7, \ldots, s_{10} \in \Sigma_{n+1} \quad k_1, k_2, k_3 \in \mathbb{Z}_+ \end{array}$$

such that

$$s_1 - Vs_2 + l_1^{2k_1} = 0 \tag{8}$$

$$s_3 + (1-V)s_4 + (p-\beta)s_5 + (1-V)(p-\beta)s_6 + (p-\beta)^{2k_2} = 0 \tag{9}$$

$$s_7 + \Big(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^2\Big)s_8 + (1-V)s_9 + (1-V)\Big(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^2\Big)s_{10} + \Big(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^2\Big)^{2k_3} = 0 \tag{10}$$

Conditions (8)–(10) cannot be directly checked using SOS programming methods. Therefore we specify convenient values for some of the s_i's and k_j's. We also restrict the degree of V and the remaining s_i's. Consequently (8)–(10) become only sufficient for (5)–(7).

Picking any of the $k_j = 0$, can prevent feasibility. Thus we set all of the k_j's equal to the next smallest value, 1. If we pick $s_2 = l_1$ and $s_1 = \hat{s}_1 l_1$, then (8) looks like the form used to show positive definiteness of a Lyapunov function in [16]. Additionally, if we pick $s_7 = s_9 = 0$, and realize that $\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^2$ is not the zero polynomial, we can write (10) in the form of a "generalized" $\mathcal{S}$-procedure. These choices leave the following problem:

$$\min \beta \quad \text{over } V \in \mathcal{R}_n \quad s_4, s_5, s_6 \in \Sigma_n \quad s_{10} \in \Sigma_{n+1}$$

such that

$$V - l_1 \in \Sigma_n \tag{11}$$

$$-\Big((1-V)s_4 + (p-\beta)s_5 + (1-V)(p-\beta)s_6 + (p-\beta)^2\Big) \in \Sigma_n \tag{12}$$

$$-\left((1-V)s_{10} + \Big(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^2\Big)\right) \in \Sigma_{n+1} \tag{13}$$

where (11) ensures the positive definiteness of V, (12) establishes $\{x|V(x) \leq 1\} \subseteq P_\beta$, and (13) constrains $\dot{V} \leq w^2$ on $\{x|V(x) \leq 1\}$.

We minimize β using an iterative algorithm and check constraints (11)–(13) using SOS programming. Before presenting the algorithm, two issues deserve mention. First, to use SOS programming, we must specify the maximum degree of V and the SOS polynomials s_i. To ensure (11)–(13) might be satisfied, the degree of the polynomials must satisfy

$$\begin{gathered} \deg V = \deg l_1 \\ \max\{\deg(Vs_4); \deg(Vps_6)\} \geq \max\{\deg(ps_5); 2\deg p\} \\ \deg s_{10} \geq \max\{\deg f; \deg g + 1\} - 1. \end{gathered} \tag{14}$$

These constraints are a consequence of the nature of polynomials; e.g. a SOS polynomial of degree 2 cannot be greater than a SOS polynomial of degree 4 for all x.

The second issue is that the algorithm does not reliably find a feasible point $\{V, s_4, s_5, s_6, s_{10}, \beta\}$. Rather it can only improve upon one, by driving β smaller. As written, the user must supply an initial V_0 that is a component of some feasible point, though the other components can be determined with SDPs. *Given* a V_0 satisfying (11), an SDP can determine the existence of s_4, s_5, and s_6 to satisfy (12). Likewise, a separate SDP can determine the existence of s_{10} satisfying (13). Note that a "poor" choice of initial V_0 may render (12) and/or (13) unsatisfiable for any choice of $\beta, s_4, s_5, s_6, s_{10}$, although for a different V_0, (12) and (13) may be satisfied. Heuristics (based on linearizations) to find suitable initial V_0's are possible.

Iterative Bounding Procedure

Setup: Specify the maximum degree that will be considered for both V and the s_i's, observing the constraints in (14). Set $l_1 = \epsilon \sum x_i^m$ for some small $\epsilon > 0$ where m is the maximum degree of V. Each step of the iteration, which is indexed by i, consists of three substeps, the first two subject to constraints (11)–(13). To begin the iteration, choose a V_0, initialize $V^{(i=0)} = V_0$ and the iteration index $i = 1$, and proceed to step 1.

1) Minimize β over s_4, s_5, s_6, and s_{10}, with $V = V^{(i-1)}$ to obtain $s_4^{(i)}$, $s_6^{(i)}$, and $s_{10}^{(i)}$.
2) Minimize β over s_5 and V, with $s_4 = s_4^{(i)}$, $s_6 = s_6^{(i)}$, and $s_{10} = s_{10}^{(i)}$ to obtain $V^{(i)}$ and $\beta^{(i)}$.
3) If $\beta^{(i)} - \beta^{(i-1)}$ is less than a specified tolerance, conclude the iteration, otherwise increment i and return to substep 1.

In (12), β is multiplied by polynomials we are searching over. Therefore we minimize β in substeps 1 and 2 using a line search.

If we restrict ourselves to linear dynamics, $\dot{x} = Ax + B_w w$, and quadratic Lyapunov functions, $V(x) = x^* P x$, then (11) becomes $P \succ 0$, and with $s_{10} = 0$, (13) becomes

$$\begin{bmatrix} A^*P + PA & PB_w \\ B_w^* P & -I \end{bmatrix} \preceq 0.$$

Thus (11) and (13) generalize the LMI in [13, §6.1.1].

A. Reachable set example

Consider the following nonlinear system

$$\begin{aligned} \dot{x}_1 &= -x_1 + x_2 - x_1 x_2^2 \\ \dot{x}_2 &= -x_2 - x_1^2 x_2 + w \end{aligned} \tag{15}$$

with $x(t) \in \mathbb{R}^2$ and $w(t) \in \mathbb{R}$. Given $p(x) = 8x_1 - 8x_1x_2 + 4x_2^2$, we would like to determine the smallest level set $P_\beta := \{x \in \mathbb{R}^2 \,|\, p(x) \leq \beta\}$ that contains all possible system trajectories for $t \leq T$ starting from $x(0) = 0$ with $\int_0^T w^2\, dt \leq R$, where R is a given constant. Employing the algorithm in §III, we fix $s_5 = 1$ and $s_6 = 0$ to eliminate the need for a line search in each substep. We initialize the algorithm with $V_0(x) = x_1^2 + x_2^2$. Figure 1 shows the modified algorithm's progress in reducing β versus iteration number as well as the trade off between R and β. After 10 iterations with $R = 1$, β is reduced to 1.08, which is a large improvement over the first iteration bound of $\beta = 10.79$. For increasing values of disturbance energy R, the size of the reachable set increases, which is expected.

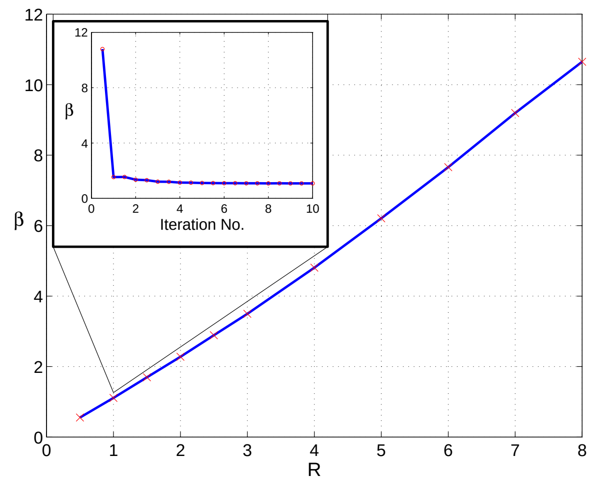

Fig. 1. Insert: The algorithm's progress for R=1, Main: Trade off between R and β.

IV. Expanding a Region of Attraction with State Feedback

Given a system of the form

$$\dot{x} = f(x) + g(x)u \tag{16}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, and f, g n-vectors of elements of $\mathcal{R}_n$ such that $f(0) = 0$, we would like to synthesize a state feedback controller $u = K(x)$ with $K \in \mathcal{R}_n$ that enlarges the set of points that we can show are attracted to the fixed point at the origin.

We define a variable sized region as $P_\beta := \{x \in \mathbb{R}^n | p(x) \leq \beta\}$, for some given positive definite p. We then expand the provable region of attraction by maximizing β while requiring that all of the points in P_β converge to the origin under the controller K. Using a Lyapunov argument, every point in P_β will converge asymptotically to the origin if there exists $K, V \in \mathcal{R}_n$ such that the following hold:

$$V(x) > 0 \text{ for all } x \in \mathbb{R}^n \backslash \{0\} \text{ and } V(0) = 0, \tag{17}$$

$$\{x \in \mathbb{R}^n | p(x) \leq \beta\} \subseteq \{x \in \mathbb{R}^n | V(x) \leq 1\}, \tag{18}$$

$$\begin{aligned} &\{x \in \mathbb{R}^n | V(x) \leq 1\} \backslash \{0\} \subseteq \\ &\{x \in \mathbb{R}^n | \frac{\partial V}{\partial x}(f(x) + g(x)K(x)) < 0\}. \end{aligned} \tag{19}$$

These conditions show that V is positive definite, P_β is contained in a level set of V, and $\frac{dV}{dt}$ is strictly negative on all the points contained in the level set aside from $x = 0$.

The condition that $V(0) = 0$ is satisfied by setting the constant term to zero. Enlarging the region of attraction subject to the preceding requirements can be cast into the following form which is amenable to the P-satz.

$$\max_{K, V \in \mathcal{R}_n} \beta$$

such that

$$\{x \in \mathbb{R}^n | V(x) \leq 0, l_1(x) \neq 0\} \text{ is empty} \tag{20}$$

$$\{x \in \mathbb{R}^n | p(x) \leq \beta, V(x) \geq 1, V(x) \neq 1\} \text{ is empty} \tag{21}$$

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} V(x) \leq 1, \; l_2(x) \neq 0, \\ \frac{\partial V}{\partial x}\Big(f(x) + g(x)K(x)\Big) \geq 0 \end{array} \right\} \text{ is empty} \tag{22}$$

where l_1, l_2 are fixed positive definite and SOS polynomials which replace the non-polynomial constraints $x \neq 0$ in (17) and (19).

Applying the P-satz, the region maximization problem with constraints (20)–(22) is equivalent to

$$\max \beta \quad \text{over} \quad \begin{matrix} K, V \in \mathcal{R}_n \quad k_1, k_2, k_3 \in \mathbb{Z}_+ \\ s_1, \ldots, s_{10} \in \Sigma_n \end{matrix}$$

such that

$$s_1 - V s_2 + l_1^{2k_1} = 0 \tag{23}$$

$$s_3 + (\beta - p)s_4 + (V-1)s_5 + (\beta - p)(V-1)s_6 + (V-1)^{2k_2} = 0 \tag{24}$$

$$s_7 + (1-V)s_8 + (\frac{\partial V}{\partial x}(f+gK))s_9 + (1-V)(\frac{\partial V}{\partial x}(f+gK))s_{10} + l_2^{2k_3} = 0. \tag{25}$$

We cannot check (23)–(25) using SOS programming methods, so we will have to pick values for some of the s_i's and k_j's. We set $k_1 = k_2 = k_3 = 1$ and pick $s_2 = l_1$ and $s_1 = \hat{s}_1 l_1$ to simplify (23). Equation (24) has a $(V-1)^{2k_2}$ term which we can not directly optimize over using SOS programming, so we cast this constraint as an $\mathcal{S}$-procedure (see §II-D). This is done by setting $s_3 = s_4 = 0$, $k_2 = 1$, and factoring out a $(V-1)$ term. To simplify (25) we set $s_{10} = 0$ and factor out l_2, leaving the sufficient conditions below,

$$\max \beta \quad \text{over } K, V \in \mathcal{R}_n \quad s_6, s_8, s_9 \in \Sigma_n$$

such that

$$V - l_1 \in \Sigma_n \tag{26}$$

$$-\Big((\beta - p)s_6 + (V-1)\Big) \in \Sigma_n \tag{27}$$

$$-\left((1-V)s_8 + \frac{\partial V}{\partial x}(f+gK)s_9 + l_2\right) \in \Sigma_n. \tag{28}$$

We employ an iterative algorithm to solve this maximization. A slight modification to (28) is needed because for a given Lyapunov candidate function V, searching over K does not affect β at all. An intermediate variable, α, is introduced to (28) so that we maximize the level set of $\{x \,|\, V(x) \le \alpha\}$ that is contractively invariant under K and use α to scale V and l_2.

If p is quadratic, and the linearization is controllable from u, then producing a feasible starting point for the algorithm is easy. Let P be the positive definite solution of the algebraic Riccati equation (ARE) $A^*P + PA - PBR^{-1}B^*P + Q = 0$. In this ARE, A, B are the state and input matrices of the linearized system and R, Q are suitable positive definite matrices. Using $V_0(x) := x^*Px$, for sufficiently small β there exists $K, s_6, s_8, s_9, l_1, l_2$ such that (26)–(28) are satisfied. For reasons highlighted in §III, the maximum degree of V, K, l_1, l_2, and the s_i's must satisfy the following constraints:

$$\begin{gathered} \deg V = \deg l_1, \\ \deg(ps_6) \ge \deg V, \\ \deg s_8 \ge \max\{\deg(fs_9); \deg(gKs_9)\} - 1, \\ \deg(Vs_8) = \deg l_2. \end{gathered} \tag{29}$$

Control Design Algorithm

Setup: Specify the maximum degree that will be considered for both V and the s_i's. Set $l_1 = \epsilon \sum x_i^m$ for some small $\epsilon > 0$, where m is the maximum degree of V. Each step of the iteration, indexed by i, consists of three substeps, two of which also involve iterations. These inner iterations will be indexed by j. To begin the iteration, choose a V_0 and initialize $V^{(i=0)} = V_0$ and $s_9^{(i=0)} = 1$. Also, set $l_2^{(i=0)} = \epsilon \sum x_i^q$, where q is the maximum degree of (Vs_8). Now set the outer iteration index $i = 1$ and proceed to step 1.

1) **Controller Synthesis:**
Set $V = V^{(i-1)}$, $s_9^{(j=0)} = s_9^{(i-1)}$, and the inner iteration index $j = 1$. In substeps 1a and 1b, the constraint
$$-\left((\alpha - V)s_8 + \frac{\partial V}{\partial x}(f+gK)s_9 + l_2\right) \in \Sigma_n$$
must be satisfied.
   (a) Maximize α over s_8, K, with $s_9 = s_9^{(j-1)}$ to obtain $K^{(j)}$.
   (b) Maximize α over s_8, s_9, with $K = K^{(j)}$ to obtain $s_9^{(j)}$ and $\alpha^{(j)}$.
   (c) If $\alpha^{(j)} - \alpha^{(j-1)}$ is less than a specified tolerance, set $s_8^{(i)} = s_8^{(j)}$, $s_9^{(i)} = s_9^{(j)}$, $l_2^{(i)} = l_2^{(i-1)}/\alpha^{(j)}$, and $\alpha^{(i)} = \alpha^{(j)}$ and continue to step 2. Otherwise increment j and return to 1a.
2) **Lyapunov Function Synthesis:**
Set $V^{(j=0)} = V^{(i-1)}/\alpha^{(i)}$, $s_8 = s_8^{(i)}$, $s_9 = s_9^{(i)}$, and $l_2 = l_2^{(i)}$, and the inner iteration index $j = 1$.
   (a) Maximize β over s_6, with $V = V^{(j-1)}$ subject to (27) to obtain $s_6^{(j)}$.
   (b) Maximize β over V, with $s_6 = s_6^{(j)}$, subject to (26)–(28) to obtain $V^{(j)}$ and $\beta^{(j)}$.
   (c) If $\beta^{(j)} - \beta^{(j-1)}$ is less than a specified tolerance, set $V^{(i)} = V^{(j)}$ and $\beta^{(i)} = \beta^{(j)}$ and continue to step 3. Otherwise increment j and return to 2a.
3) If $\beta^{(i)} - \beta^{(i-1)}$ is less than a specified tolerance conclude the iterations, otherwise return to step 1.

As in §III, we use a line search to maximize α and β in the steps above.

A. Expanding the region of attraction for systems with input saturation

Given a system of the form

$$\dot{x} = f(x) + g(x)\,\text{sat}(u) \tag{30}$$

where

$$\text{sat}(u) := \begin{cases} u & \text{if } |u| \le 1 \\ 1 & \text{if } u > 1 \\ -1 & \text{if } u < -1 \end{cases}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, and f, g n-vectors of elements of $\mathcal{R}_n$ such that $f(0) = 0$, we would like to synthesize a state feedback controller $u = K(x)$ with $K \in \mathcal{R}_n$ to enlarge the set of points which are attracted to the origin.

Again, we define the region to expand as $P_\beta := \{x \in \mathbb{R}^n | p(x) \le \beta\}$, for some given positive definite p. We want to design state feedback controller $K(x)$ to maximize β such that the P_β is a domain of attraction and $|u| \le 1$. This is accomplished by appending two conditions to (17)–(19):

$$\{x \in \mathbb{R}^n | V(x) \le 1\} \subseteq \{x \in \mathbb{R}^n | K(x) \le 1\}, \qquad (31)$$

$$\{x \in \mathbb{R}^n | V(x) \le 1\} \subseteq \{x \in \mathbb{R}^n | K(x) \ge -1\}. \qquad (32)$$

These two equations ensure that $|u| = |K(x)| \le 1$ for all x in the contractively invariant set $\{x \in \mathbb{R}^n | V(x) \le 1\}$, so the control action will not hit saturation.

Following the procedure in §IV, we will obtain constraints (26)–(28). Additionally due to the saturation, we have

$$\Big((1 - K) - (1 - V)s_{10}\Big) \in \Sigma_n, \qquad (33)$$

$$\Big((1 + K) - (1 - V)s_{11}\Big) \in \Sigma_n. \qquad (34)$$

The control design algorithm for this problem is similar to that proposed in §IV, with the inclusions of the two additional constraints (33) and (34).

B. State feedback example

Consider the following nonlinear system: $\dot{x}_1 = u$, $\dot{x}_2 = -x_1 + \frac{1}{6}x_1^3 - u$, with $x(t) \in \mathbb{R}^2$ and $u(t) \in \mathbb{R}$. We are interested in enlarging the domain of attraction described by the level set $P_\beta := \{x \in \mathbb{R}^2 \,|\, p(x) \le \beta\}$, where $p(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_1x_2 + \frac{1}{12}x_2^2$, through state feedback. Using the algorithm in §IV, we start with $V_0(x) = \frac{1}{4.6}(1.5x_1^2 + x_1x_2 + x_2^2)$. We set the maximum degrees of V, K, s_6, s_8 and s_9 to 2, 2, 2, 2, and 0 respectively.

Figure 2 shows the resulting domain of attraction using our algorithm with the largest β achieved equaling 16.1. This was obtained after 14 iterations. The resulting controller K is shown in the title of figure 2 as $x1'$ ($x1' = \dot{x}_1 = u$). Surprisingly, for higher orders of V and K, we have obtained smaller regions of attraction. This is likely due to the nonconvexity of the overall control design algorithm. Although each substep is optimal (i.e., convex), our iterative approach of breaking the algorithm into substeps is not.

V. Conclusions

Our expansion of existing SOS programming results to system theoretic questions about nonlinear systems with polynomial vector fields appears promising. As limited as these two algorithms are, the underlying technique provides opportunities to extend standard LMI analysis of linear systems to more general polynomial vector fields. A drawback of the approach is that implementation of each algorithm requires a feasible starting point. This may be produced by trial and error, or using established nonlinear design techniques.

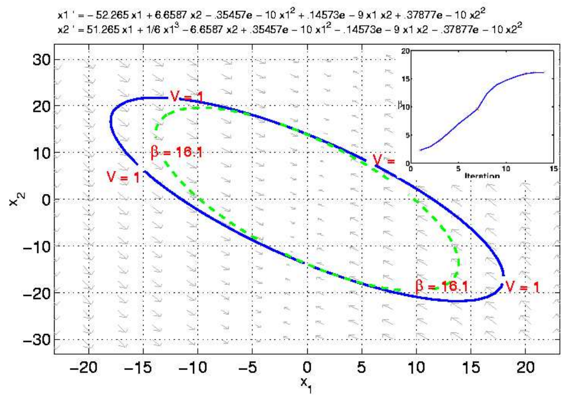

Fig. 2. Closed loop system's region of attraction. Insert: β vs. iteration no.

VI. References

[1] P. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, , no. 96:2, pp. 293–320, 2003.

[2] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optimiz.*, vol. 11(3), pp. 796–817, 2000.

[3] N. Shor, *Nondiferentiable Optimization and Polynomial Problems*, Kluwer Academic Pub, 1998.

[4] J. Bochnak, M. Coste, and M-F. Roy, *Géométrie algébrique réelle*, Springer, Berlin, 1986.

[5] K. Schmüdgen, "The k-moment problem for compact semialgebraic sets," *Mathematische Annalen*, vol. 289, pp. 203–206, 1991.

[6] M. Putinar, "Positive polynomials on compact semialgebraic sets," *Indiana University Mathematical Journal*, vol. 42, pp. 969–984, 1993.

[7] D. Henrion and J. Lasserre, "Gloptipoly: Global optimization over polynomials with matlab and sedumi," in *Proc. of the Conf. on Decision and Control*, 2002, pp. 747–752.

[8] M. Choi, T. Lam, and B. Reznick, "Sums of squares of real polynomials," in *Proc. Sympos. Pure Math.*, 1995, vol. 58(2), pp. 103–126.

[9] V. Powers and T. Wörmann, "An algorithm for sums of squares of real polynomials," *J. Pure Appl. Algebra*, vol. 127, pp. 99–104, 1998.

[10] S. Prajna, A. Papachristodoulou, and P. Parrilo, "Introducing sostools: A general purpose sum of squares programming solver," in *Proc. of the Conf. on Decision and Control*, 2002, pp. 741–746.

[11] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2002.

[12] J. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999, Available at `http ://fewcal.cub.nl/sturm/software/sedumi.html`.

[13] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[14] H. Anai and V. Weispfenning, "Reach set computations using real quantifier elimination," in *Hybrid Systems: Comp and Ctrl*, M. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. 2001, number 2034 in Lecture Notes in Comp Sci, pp. 63–76, Springer.

[15] P. Parrilo and B. Sturmfels, "Minimizing polynomial functions," in *Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science, held at DIMACS, Rutgers University, March 12-16*, 2001, available at `http://control.ee.ethz.ch/~parrilo`.

[16] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition," in *Proc. of the Conf. on Decision and Control*, 2002, pp. 3482–3487.