ECE285: Semidefinite and sum-of-squares optimization **Winter 2023**

Lecture 15: SOS applications in control and machine learning

Lecturer: Yang Zheng *Scribe: Yang Zheng*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. Any typos should be sent to* `zhengy@eng.ucsd.edu`.

Learning goals:

1. SOS hierarchies for constrained optimization
2. Lyapunov functions for nonlinear control
3. Neural network verification in machine learning

In this lecture, we continue with the previous lecture on SOS hierarchies for general constrained polynomial optimization. Then, we will introduce two SOS applications in nonlinear control and machine learning. For many other applications, you can refer to books [4,7] and surveys [2,9,12].

1 SOS hierarchies for constrained optimization

Consider a general constrained polynomial optimization problem

$$\begin{aligned} p^\star = \min_{x} \quad & p(x) \\ \text{subject to} \quad & g_i(x) \geq 0, i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $p(x), g_1(x), \ldots, g_m(x)$ are given polynomials. Similar to the previous lectures, (1) is equivalent to

$$\begin{aligned} \gamma^\star = \max_{\gamma} \quad & \gamma \\ \text{subject to} \quad & p(x) - \gamma \geq 0, \;\; \forall x \in \mathcal{K}, \end{aligned} \tag{2}$$

where $\mathcal{K} := \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, i = 1, \ldots, m\}$. Then, we can replace the nonnegativity on set $\mathcal{K}$ by an SOS condition. In particular, for any integer r, we consider the following SOS hierarchy (known as Lasserre's hierarchy [7])

$$\begin{aligned} \gamma_{2r} = \max_{\gamma} \quad & \gamma \\ \text{subject to} \quad & p(x) - \gamma = s_0(x) + s_1(x)g_1(x) + \ldots + s_m(x)g_m(x) \\ & s_0, s_1, \ldots, s_m \text{ are SOS} \\ & \deg(s_0) \leq 2r, \; \deg(s_i g_i) \leq 2r, i = 1, \ldots, m. \end{aligned} \tag{3}$$

It is not difficult to see that (3) can be formulated as a semidefinite program. Furthermore, the sequence γ_r is monotonically non-decreasing, and

$$\gamma_r \leq \gamma^\star = p^\star, \; \forall r \in \mathbb{N}.$$

Indeed, if the set $\mathcal{K}$ is compact and satisfies an extra algebraic condition — Archimedean, e.g., there is i such that $\{x \in \mathbb{R}^n \mid g_i(x) \geq 0\}$ is compact, then Putinar's Positivstellensatz ensures that $\lim_{r\to\infty} \gamma_r = p^\star$. In fact, the convergence is almost finite (generically), and this is proved in [8].

Remark 15.1. *We can use other Positivstellensatz to derive converging SOS hierarchies, in addition to* (3)*, which may require less assumption on* $\mathcal{K}$*. For instance, we can use Schmudgen Positivstellensatz, which was first used by Pablo [10]. There are also other Positivstellensatz that do not rely on SOS conditions. One such result is Polya's theorem. See a recent nice paper [1] for more constructions.* □

Example 15.1 (Example 5.3 in [7]). *Consider the following optimization problem*

$$\begin{aligned} \min_{x_1,x_2} \quad & x_1^2 x_2^2 (x_1^2 + x_2^2 - 1) \\ \textit{subject to} \quad & x_1^2 + x_2^2 \leq 4. \end{aligned} \tag{4}$$

It is clear that the constraint $x_1^2 + x_2^2 \leq 4$ satisfies the Archimedean condition. Then, the sequence γ_r from the Lasserre's SOS hierarchy in (3) *will converge to the true optimal cost value of* (4)*. In particular, the SOS hierarchy reads as*

$$\begin{aligned} \gamma_r = \max_{\gamma} \quad & \gamma \\ \textit{subject to} \quad & p(x) - \gamma = s_0(x) + s_1(x)(4 - x_1^2 - x_2^2) \\ & s_0 \in \Sigma_{2,2r}, s_1 \in \Sigma_{2,2r-2}. \end{aligned} \tag{5}$$

The following YALMIP code implements the first four levels of (5)*.*

```
x     = sdpvar(1,1);
y     = sdpvar(1,1);
gamma = sdpvar(1,1);
p     = x^2*y^2*(x^2+y^2-1);    % cost function
g     = 4 - x^2 - y^2;          % constraint
degree = [4,6,8,10];            % degree
bound  = zeros(length(degree),1);
for i = 1:length(degree)
      [s,c,v] = polynomial([x;y],degree(i));          % sos multiplier
      F       = sos(s);   % SOS constraint
      F       = [F, sos(p - gamma - s*g)];
      solvesos(F,-gamma,[],[gamma;c]);                % solve SOS program
      bound(i) = value(gamma);
end
```

The optimal γ from these SOS programs are

$$\gamma_3 = -0.0417, \quad \gamma_4 = -0.0370, \quad \gamma_5 = -0.0370, \gamma_6 = -0.0370.$$

In this case, the optimal value stabilizes at the second level. We can expect that the optimal cost to (4) *should be $p^* = -0.0417$. Indeed, there are sufficient conditions (mainly based on the flat extension theory from the moment side [5]) to certify when the hierarchy stabilizes at the optimal solutions and further extract global minimizers (see [7, Section 5.3.1]).* □

2 Lyapunov functions

We have seen that quadratic Lyapunov functions are very useful for analysis and control of linear time-invariant systems, which leads to various SDP formulations. SOS optimization allows us to deal with many analysis and synthesis problems of nonlinear systems. In this section, we consider the basic problem of establishing stability of nonlinear systems.

Consider a dynamical system

$$\dot{x}(t) = f(x),$$

where $f(x)$ is a vector of polynomials. Assume that the origin $x = 0 \in \mathbb{R}^n$ is an equilibrium of the system, i.e., $f(0) = 0$. We aim to check whether all the trajectories $x(t)$ converges to 0 at $t \to \infty$. One method is to find a Lyapunov function that can be viewed as a positive energy function decreasing along any trajectories. The nonlinear system is globally asymptotically stable, if we can find a Lyapunov function $V(x) : \mathbb{R}^n \to \mathbb{R}$ that satisfies

$$\begin{aligned} V(x) > 0, \quad & x \in \mathbb{R}^n \setminus \{0\}, \\ \dot{V}(x) = \langle \nabla V(x), f(x) \rangle < 0, \quad & x \in \mathbb{R}^n \setminus \{0\}. \end{aligned}$$

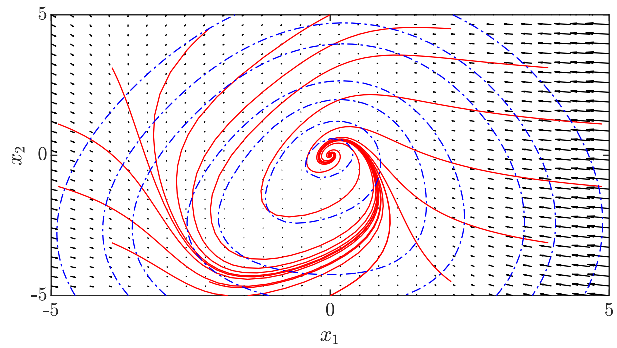

Figure 1: Trajectories of nonlinear dynamical system (7) and the level sets of a Lyapunov function found via SOS techniques.

We will consider candidate Lyapunov functions to be polynomials (quadratics and beyond). Since nonnegativity over polynomials is computationally hard, we will impose that the candidate Lyapunov function $V(x)$ and its derivative $\dot{V}(x)$ satisfy stronger SOS conditions[1]

$$\begin{aligned} V(x) \quad &\text{is SOS}, \\ -\langle \nabla V(x), f(x) \rangle \quad &\text{is SOS}. \end{aligned} \tag{6}$$

Upon fixing the degree of $V(x)$, searching for a polynomial Lyapunov function satisfying (6) is amount to solving an SDP.

Example 15.2 ([4, Chapter 3.6.2])**.** *Consider the following nonlinear system*

$$\begin{aligned} \dot{x} &= -y - \frac{3}{2}x^2 - \frac{1}{2}x^3, \\ \dot{y} &= 3x - y, \end{aligned} \tag{7}$$

which corresponds to the Moore-Greitzer model of a jet engine with stabilizing feedback operating in the no-stall model. Using following YALMIP code:

```
% define variables
x = sdpvar(1); y = sdpvar(1);

% Constructing the vector field dx/dt = f
f = [-y - 1.5*x^2 - 0.5*x^3;
     3*x-y];

% Step 1: Construct a Lyapunov candidate
degree         = 4;
[V,Vc,mbasis] = polynomial([x;y],degree,1);

% Step 2: SOS constraint
Constraint = [sos(V - (x^2+y^2))];
epsilon    =1e-6;
Vdot       = -(jacobian(V,x)*f(1)+jacobian(V,y)*f(2)) - epsilon*(x^2+y^2);
Constraint = [Constraint, sos(Vdot)];

% Step 3: solve the sos program
solvesos(Constraint,[],[],[Vc]);

Vcoef = value(Vc);   % the coefficient of the Lyapunov function
```

we can easily find a Lyapunov function that is a polynomial of degree 4*. The trajectories of the nonlinear system, and the level sets of the Lyapunov function is shown in Figure 1.* □

[1]The strict positivity requirement can be handled by including a strictly positive term, e.g., $\epsilon\|x\|_2^2$.

Remark 15.2. *The SOS conditions (6) are sufficient for global asymptotic stability. Similar SOS conditions can be derived for many other analysis tasks, e.g, local asymptotic stability, and region of attraction etc. See e.g., [2, 9]* □

3 Neural network verification

Neural networks are one of the fundamental building blocks of modern machine-learning methods. For safety-critical applications, it is essential to ensure that they are provably robust to input perturbations.

Given a neural network $f(x_0) : \mathbb{R}^d \to \mathbb{R}^m$, a nominal input $\bar{x} \in \mathbb{R}^d$, a linear function $\phi : \mathbb{R}^m \to \mathbb{R}$ on the network's output, and a perturbation radius $\epsilon \in \mathbb{R}$, the network verification problem asks to either verify that

$$\phi(f(x_0)) > 0 \quad \forall x_0 : \|x_0 - \bar{x}\|_\infty \leq \epsilon, \tag{8}$$

or to identify at least one counterexample to this relation.

Consider an L-layer feedforward neural network where

$$\begin{aligned} f(x_0) &= W_L x_L + b_L, \\ x_{i+1} &= \mathrm{ReLU}(W_i x_i + b_i), \quad i = 0, \ldots, L-1, \end{aligned}$$

where $W_i \in \mathbb{R}^{n_{i+1} \times n_i}$ and $b_i \in \mathbb{R}^{n_{i+1}}$ are the network weights and biases, respectively, and the so-called Rectified Linear Unit (ReLU) activation function $\mathrm{ReLU} : \mathbb{R}^k \to \mathbb{R}^k$ is the element-wise positive part of its argument, $\mathrm{ReLU}(z) = [\max(z_i, 0)]_{i=1}^k$. Condition (8) can be decided by solving the optimization problem

$$\begin{aligned} \gamma^\star := \min_{x_0,\ldots,x_L} \quad & c^\mathsf{T} x_L + c_0 \\ \text{subject to} \quad & x_{i+1} = \mathrm{ReLU}(W_i x_i + b_i),\ i \in [L], && \text{(9a)} \\ & \|x_0 - \bar{x}\|_\infty \leq \epsilon, && \text{(9b)} \end{aligned}$$

where $[L] := \{0, 1, \ldots, L-1\}$ and c, c_0 are problem data related to the linear function $\phi(\cdot)$. If $\gamma^\star > 0$, then (8) holds, otherwise counterexamples can be found.

Since the action of the ReLU function can be described by quadratic constraints,

$$y = \mathrm{ReLU}(z) \quad \Longleftrightarrow \quad y \geq z,\ y \geq 0,\ y(y - z) = 0,$$

problem (9) can be reformulated into a polynomial optimization problem with variable $x = \left[x_0^\mathsf{T}, x_1^\mathsf{T}, \ldots, x_L^\mathsf{T}\right]^\mathsf{T}$ (which is indeed a QCQP [11]), and subsequently relaxed into an SDP in previous lectures or solved using the SOS hierarchy (3) above.

The interested reader can refer to [3, 6, 11] for more detailas and numerical experiments.

Notes

Further reading for this lecture can refer to [4, Chapter 3] and [7, Chapter 3].

References

[1] Amir Ali Ahmadi and Georgina Hall. On the construction of converging hierarchies for polynomial optimization based on certificates of global positivity. *Mathematics of Operations Research*, 44(4):1192–1207, 2019.

[2] James Anderson and Antonis Papachristodoulou. Advances in computational lyapunov analysis using sum-of-squares programming. *Discrete & Continuous Dynamical Systems-B*, 20(8):2361, 2015.

[3] Ben Batten, Panagiotis Kouvaros, Alessio Lomuscio, and Yang Zheng. Efficient neural network verification via layer-based semidefinite relaxations and linear cuts. In *International Joint Conference on Artificial Intelligence (IJCAI21)*, pages 2184–2190, 2021.

[4] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

[5] Raúl E Curto and Lawrence A Fialkow. *Solution of the truncated complex moment problem for flat data*, volume 568. American Mathematical Soc., 1996.

[6] Mahyar Fazlyab, Manfred Morari, and George J Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 2020.

[7] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

[8] Jiawang Nie. Optimality conditions and finite convergence of lasserre's hierarchy. *Mathematical programming*, 146(1):97–121, 2014.

[9] Antonis Papachristodoulou and Stephen Prajna. A tutorial on sum of squares techniques for systems analysis. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2686–2700. IEEE, 2005.

[10] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[11] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018.

[12] Yang Zheng, Giovanni Fantuzzi, and Antonis Papachristodoulou. Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization. *Annual Reviews in Control*, 2021.