Sum of Squares (SOS)

Matthew M. Peet
Arizona State University

Lecture 02: Sum of Squares (SOS)

The Dual Problem of Polynomial Programming

Polynomial Programming (NOT CONVEX): n decision variables

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$g_i(x) \geq 0$$

- f and g_i must be convex for the problem to be convex.

Optimization of Polynomials IS Convex: Lifting to a higher-dimensional space

$$\max_{g,\gamma} \quad \gamma$$
$$f(x) - \gamma = g(x) \qquad \text{for all} \qquad x \in \mathbb{R}^n$$
$$g(x) \geq 0 \qquad \text{for all} \qquad x \in \{x \in \mathbb{R}^n \;:\; h(x) \geq 0\}$$

- The decision variables are *functions* (e.g. g)
  - **Infinite Dimensional Contraints:** One constraint for *every value of* x.
- But how to parameterize functions????
- How to enforce an infinite number of constraints???
- Advantage: Problem is convex, even if f, g, h are not convex.

The Dual Problem of Polynomial Programming

Polynomial Programming (NOT CONVEX): n decision variables

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$g_i(x) \geq 0$$

- f and g_i must be convex for the problem to be convex.

Optimization of Polynomials IS Convex: Lifting to a higher-dimensional space

$$\max_{g,\gamma} \quad \gamma$$
$$f(x) - \gamma = g(x) \qquad \text{for all} \quad x \in \mathbb{R}^n$$
$$g(x) \geq 0 \qquad \text{for all} \quad x \in \{x \in \mathbb{R}^n \,:\, h(x) \geq 0\}$$

- The decision variables are *functions* (e.g. g)
  - **Infinite Dimensional Contraints:** One constraint for *every value of* x.
- But how to parameterize functions????
- How to enforce an infinite number of constraints???
- Advantage: Problem is convex, even if f, g, h are not convex.

- Hopefully you know what convexity is.
- Parameterize functions as polynomials.
- feasibility of a point x is easy to show.
- Infeasibility of a constraint requires a certificate

For Polynomial Programming

- feasibility of a point x is easy to show.
- Infeasibility of a constraint requires a certificate

For Optimization of Polynomials

- infeasibility is easy to show (give a counterexample).
- Feasibility of a function requires a certificate

Stability of Systems with Positive States: Not all states can be negative...

- Cell Populations/Concentrations
- Volume/Mass/Length

We want:

$$V(x) = x^T P x \geq 0 \qquad \text{for all} \qquad x \geq 0$$
$$\dot{V}(x) = x^T (A^T P + PA) x \leq 0 \qquad \text{for all} \qquad x \geq 0$$

Formulation:

- Matrix Copositivity (An NP-hard Problem)

Verify:

$$x^T P x \geq 0 \qquad \text{for all} \quad x \geq 0$$

Implementation: `sosdemo4p.m`

Recall: Systems with Uncertainty

$$\dot{x}(t) = A(\delta)x(t) + B_1(\delta)w(t) + B_2(\delta)u(t)$$
$$y(t) = C(\delta)x(t) + D_{12}(\delta)u(t) + D_{11}(\delta)w(t)$$

Theorem 1.

There exists an $F(\delta)$ such that $\|\underline{S}(P(\delta), K(0,0,0,F(\delta)))\|_{H_\infty} \le \gamma$ for all $\delta \in \mathbf{\Delta}$ if there exist $Y > 0$ and $Z(\delta)$ such that

$$\begin{bmatrix} YA(\delta)^T + A(\delta)Y + Z(\delta)^T B_2(\delta)^T + B_2(\delta)Z(\delta) & *^T & *^T \\ B_1(\delta)^T & -\gamma I & *^T \\ C_1(\delta)Y + D_{12}(\delta)Z(\delta) & D_{11}(\delta) & -\gamma I \end{bmatrix} < 0 \quad \text{for all} \quad \delta \in \mathbf{\Delta}$$

Then $F(\delta) = Z(\delta)Y^{-1}$.

The Structured Singular Value, μ

Definition 2.

Given system $M \in \mathcal{L}(L_2)$ and set $\mathbf{\Delta}$ as above, we define the **Structured Singular Value** of $(M, \mathbf{\Delta})$ as

$$\mu(M, \mathbf{\Delta}) = \frac{1}{\inf_{\substack{\Delta \in \mathbf{\Delta} \\ I - M\Delta \text{ is singular}}} \|\Delta\|}$$

The system

$$M = \left[\begin{array}{c|c} A_0 & M \\ \hline N & Q \end{array}\right]$$

Lower Bound for μ: $\mu \geq \gamma$ if there exists a $P(\delta)$ such that

$$P(\delta) \geq 0 \qquad \text{for all} \quad \delta \;\; \text{AND}$$

$$P(\delta)(A_0 x + Mp) + (A_0 x + Mp)^T P(\delta) < \epsilon I \;\; \text{for all} \;\; x, p, \delta \text{ such that}$$

$$(x, p, \delta) \in \left\{ x, p, \delta \,:\, p = \operatorname{diag}(\delta_i)(Nx + Qp),\; \sum_i \delta_i^2 \leq \gamma \right\}$$

Implementation (Simplified Version): `sosdemo5p.m`

Overview

In this lecture, we will show how the LMI framework can be expanded dramatically to other forms of control problems.

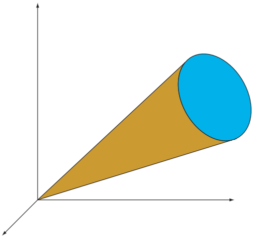
1. Positivity of Polynomials
    1.1 Sum-of-Squares
2. Positivity of Polynomials on Semialgebraic sets
    2.1 Inference and Cones
    2.2 Positivstellensatz
3. Applications
    3.1 Nonlinear Analysis
    3.2 Robust Analysis and Synthesis
    3.3 Global optimization

Is Optimization of Polynomials Tractable or Intractable?

The Answer lies in Convex Optimization

A Generic Convex Optimization Problem:

$$\max_x \; bx$$
$$\text{subject to} \quad Ax \in C$$



The problem is *convex optimization* if

- C is a convex cone.
- b and A are affine.

Computational Tractability: Convex Optimization over C is tractable if

- The set membership test for $y \in C$ is in P (polynomial-time verifiable).
- The variable x is a finite dimensional vector (e.g. $\mathbb{R}^n$).

Optimization of Polynomials is Convex

The variables are finite-dimensional (if we bound the degree)

Convex Optimization of Functions: Variables $V \in \mathcal{C}[\mathbb{R}^n]$ and $\gamma \in \mathbb{R}$

$$\max_{V,\gamma} \gamma$$

subject to

$$V(x) - x^T x \geq 0 \qquad \forall x$$
$$\nabla V(x)^T f(x) + \gamma x^T x \leq 0 \qquad \forall x$$

V is the decision variable (infinite-dimensional)

- How to make it finite-dimensional???

The set of polynomials is an infinite-dimensional (but *Countable*) vector space.

- It is **Finite Dimensional** if we bound the degree
- All finite-dimensional vector spaces are equivalent!

But we need a way to parameterize this space...

To Begin: How do we Parameterize Polynomials???

A Parametrization consists of a **basis** and a **set of parameters** (coordinates)

- We use a **Finite Dimensional** space of polynomials of degree d or less.
  - The monomials are a simple basis for the space of polynomials

Definition 3.

Define $Z_d(x)$ to be the vector of monomial bases of degree d or less.

e.g., if $x \in \mathbb{R}^2$, then the vector of basis functions is

$$Z_2(x_1, x_2)^T = \begin{bmatrix} 1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2 \end{bmatrix}$$

and

$$Z_4(x_1)^T = \begin{bmatrix} 1 & x_1 & x_1^2 & x_2^3 & x_1^4 \end{bmatrix}$$

Linear Representation

- Any polynomial of degree d can be represented with a vector $c \in \mathbb{R}^m$

$$p(x) = c^T Z_d(x)$$

- c is the vector of *parameters* (decision variables).

$$2x_1^2 + 6x_1x_2 + 4x_2 + 1 = \begin{bmatrix} 1 & 0 & 4 & 6 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2 \end{bmatrix}^T$$

Implementation: `Zd=monomials([x1 x2],0:4)`

Optimization of Polynomials is Convex

The variables are finite-dimensional (if we bound the degree)

Convex Optimization of Functions: Variables $V \in \mathcal{C}[\mathbb{R}^n]$ and $\gamma \in \mathbb{R}$

$$\max_{V,\gamma} \gamma$$

subject to

$$V(x) - x^T x \geq 0 \qquad \forall x$$
$$\nabla V(x)^T f(x) + \gamma x^T x \leq 0 \qquad \forall x$$

Now use the polynomial parametrization $V(x) = c^T Z(x)$

- Now c is the decision variable.

Convex Optimization of Polynomials: Variables $c \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$

$$\max_{c,\gamma} \gamma$$

subject to

$$c^T Z(x) - x^T x \geq 0 \qquad \forall x$$
$$c^T \nabla Z(x) f(x) + \gamma x^T x \leq 0 \qquad \forall x$$

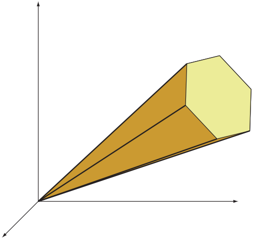
Can LMIs be used for Optimization of Polynomials???

Optimization of Polynomials is NP-Hard!!!

Problem: Use a finite number of variables:

$$\max\ b^T x$$

$$\text{subject to } A_0(y) + \sum_i^n x_i A_i(y) \succeq 0 \quad \forall y$$



The A_i are matrices of polynomials in y. e.g. Using multi-index notation,

$$A_i(y) = \sum_\alpha A_{i,\alpha}\, y^\alpha$$

The FEASIBLITY TEST is Computationally Intractable

The problem: "Is $p(x) \geq 0$ for all $x \in \mathbb{R}^n$?" (i.e. "$p \in \mathbb{R}^+[x]$?") is NP-hard.

Definition 4.

A Polynomial, f, is called Positive SemiDefinite (PSD) if

$$f(x) \geq 0 \quad \text{for all } x \in \mathbb{R}^n$$

The Primary Problem: How to enforce the constraint $f(x) \geq 0$ for all x?

Easy Proof: Certificate of Infeasibility

- A Proof that f is NOT PSD.
- i.e. To show that

$$f(x) \geq 0 \qquad \text{for all } x \in \mathbb{R}^n$$

  is FALSE, we need only find a point x with $f(x) < 0$.

Complicated Proof: It is much harder to identify a **Certificate of Feasibility**

- A Proof that f is PSD.

Global Positivity Certificates (Proofs and Counterexamples)

Question: How does one prove that $f(x)$ is positive semidefinite?

What Kind of Functions do we Know are PSD?

- Any squared function is positive.
- The sum of squared forms is PSD
- The product of squared forms is PSD
- The ratio of squared forms is PSD

So $V(x) \geq 0$ for all $x \in \mathbb{R}^n$ if

$$V(x) = \prod_k \frac{\sum_i f_{ik}(x)^2}{\sum_j h_{jk}(x)^2}$$

But is any PSD polynomial the sum, product, or ratio of squared polynomials?

- An old Question....

Definition 5.

A polynomial, $p(x) \in \mathbb{R}[x]$ is a **Sum-of-Squares (SOS)**, denoted $p \in \Sigma_s$ if there exist polynomials $g_i(x) \in \mathbb{R}[x]$ such that

$$p(x) = \sum_i^k g_i(x)^2.$$

David Hilbert created a famous list of 23 then-unsolved mathematical problems in 1900.

- Only 10 have been fully resolved.
- The 17th problem has been resolved.

"Given a multivariate polynomial that takes only non-negative values over the reals, can it be represented as a sum of squares of rational functions?" -D. Hilbert, 1900

Hilbert's 17th was resolved in the affirmative by E. Artin in 1927.

- Any PSD polynomial is the sum, product and ratio of squared polynomials.
- If $p(x) \geq 0$ for all $x \in \mathbb{R}^n$, then

$$p(x) = \frac{g(x)}{h(x)}$$

  where $g, h \in \Sigma_s$.
- If p is positive definite, then we can assume $h(x) = (\sum_i x_i^2)^d$ for some d. That is,

$$(x_1^2 + \cdots + x_n^2)^d p(x) \in \Sigma_s$$

- If we can't find a SOS representation (certificate) for $p(x)$, we can try $(\sum_i x_i^2)^d p(x)$ for higher powers of d.

Of course this doesn't answer the question of how we find SOS representations.

How to use LMIs to Prove Polynomial Positivity?

Basic Idea: If there exists a Positive Matrix $P \geq 0$ such that

$$V(x) = Z_d(x)^T P Z_d(x)$$

Then $V(x)$ is positive

Why? Positive Matrices ($P \geq 0$) have square roots!

$$P = Q^T Q$$

Hence

$$\begin{aligned} V(x) &= Z_d(x)^T Q^T Q Z_d(x) = (Q Z_d(x))^T (Q Z_d(x)) \\ &= h(x)^T h(x) \geq 0 \end{aligned}$$

Conclusion:

$$V(x) \geq 0 \quad \text{for all} \quad x \in \mathbb{R}^n$$

if there exists a $P \geq 0$ such that

$$V(x) = Z_d(x)^T P Z_d(x)$$

- Such a function is called Sum-of-Squares (SOS), denoted $V \in \Sigma_s$.
- This is an LMI! Equality constraints relate the coefficients of V (decision

SOS as an LMI

Conversion between Linear and Quadratic Representation

Let

$$V(x) = c^T Z_{2d}(x)$$

V is SOS iff there exists a $P \geq 0$ such that

$$V(x) = Z_d(x)^T P Z_d(x)$$

Construct A so that

$$Z_d(x)^T P Z_d(x) = vec(P) A Z_{2d}(x)$$

$$V(x) = Z_d(x)^T P Z_d(x)$$

becomes

$$c^T Z_{2d}(x) = vec(P) A Z_{2d}(x)$$

or

$$A^T vec(P) = c$$

Quadratic Parameterization of Polynomials

Quadratic Representation

- Alternative to Linear Parametrization, a polynomial of degree d can be represented by a matrix $M \in \mathbb{S}^m$ as
$$p(x) = Z_d(x)^T M Z_d(x)$$
- However, now the problem may be under-determined
$$\begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}^T \begin{bmatrix} M_1 & M_2 & M_3 \\ M_2 & M_4 & M_5 \\ M_3 & M_5 & M_6 \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}$$
$$= M_1 x^4 + 2M_2 x^3 y + (2M_3 + M_4) x^2 y^2 + 2M_5 x y^3 + M_6 y^4$$

Thus, there are infinitely many quadratic representations of p. For the polynomial
$$f(x) = 4x^4 + 4x^3 y - 7x^2 y^2 - 2xy^3 + 10y^4,$$
we can use the alternative solution
$$4x^4 + 4x^3 y - 7x^2 y^2 - 2xy^3 + 10y^4$$
$$= M_1 x^4 + 2M_2 x^3 y + (2M_3 + M_4) x^2 y^2 + 2M_5 x y^3 + M_6 y^4$$

Polynomial Representation - Quadratic

For the polynomial

$$f(x) = 4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4,$$

we require

$$\begin{aligned}&4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4\\&\qquad = M_1x^4 + 2M_2x^3y + (2M_3 + M_4)x^2y^2 + 2M_5xy^3 + M_6y^4\end{aligned}$$

Constraint Format:

$$M_1 = 4; \quad 2M_2 = 4; \quad 2M_3 + M_4 = -7; \quad 2M_5 = -2; \quad 10 = M_6.$$

An underdetermined system of linear equations (6 variables, 5 equations).

- This yields a family of quadratic representations, parameterized by λ as

$$4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4 = \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}^T \begin{bmatrix} 4 & 2 & -\lambda \\ 2 & -7+2\lambda & -1 \\ -\lambda & -1 & 10 \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}$$

  which holds for any $\lambda \in \mathbb{R}$

Quadratic Form:

$$p(x) = Z_d(x)^T M Z_d(x)$$

Consider the case where the matrix M is positive semidefinite.

Suppose: $p(x) = Z_d(x)^T M Z_d(x)$ where $M > 0$.

- Any positive semidefinite matrix, $M \geq 0$ has a square root $M = PP^T$

Hence

$$p(x) = Z_d(x)^T M Z_d(x) = Z_d(x)^T P P^T Z_d(x).$$

Which yields

$$p(x) = \sum_i \left(\sum_j P_{i,j} Z_{d,j}(x) \right)^2$$

which makes $p \in \Sigma_s$ an SOS polynomial.

Positive Matrix Representation of SOS

Necessity

Moreover: Any SOS polynomial has a quadratic rep. with a PSD matrix.

Suppose: $p(x) = \sum_i g_i(x)^2$ is degree $2d$ (g_i are degree d).

- Each $g_i(x)$ has a linear representation in the monomials.
$$g_i(x) = c_i^T Z_d(x)$$
- Hence
$$p(x) = \sum_i g_i(x)^2 = \sum_i Z_d(x) c_i c_i^T Z_d(x) = Z_d(x) \left(\sum_i c_i c_i^T \right) Z_d(x)$$
- Each matrix $c_i c_i^T \geq 0$. Hence $Q = \sum_i c_i c_i^T \geq 0$.
- We conclude that if $p \in \Sigma_s$, there is a $Q \geq 0$ with $p(x) = Z_d(x) Q Z_d(x)$.

Lemma 6.

Suppose M is polynomial of degree $2d$. $M \in \Sigma_s$ if and only if there exists some $Q \succeq 0$ such that
$$M(x) = Z_d(x)^T Q Z_d(x).$$

Sum-of-Squares

Thus we can express the search for a SOS certificate of positivity as an LMI.

Take the numerical example

$$4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4$$

The question of an SOS representation is equivalent to

$$\text{Find} \quad M = \begin{bmatrix} M_1 & M_2 & M_3 \\ M_2 & M_4 & M_5 \\ M_3 & M_5 & M_6 \end{bmatrix} \geq 0 \quad \text{such that}$$

$$M_1 = 4; \quad 2M_2 = 4; \quad 2M_3 + M_4 = -7; \quad 2M_5 = -2; \quad M_6 = 10.$$

In fact, this is feasible for

$$M = \begin{bmatrix} 4 & 2 & -6 \\ 2 & 5 & -1 \\ -6 & -1 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 2 & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & -3 \end{bmatrix}$$

Sum-of-Squares

We can use this solution to construct an SOS certificate of positivity.

$$
\begin{aligned}
4x^4+4x^3y-7x^2y^2-2xy^3+10y^4 &= \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}^T \begin{bmatrix} 4 & 2 & -6 \\ 2 & 5 & -1 \\ -6 & -1 & 10 \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix} \\
&= \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix}^T \begin{bmatrix} 0 & 2 \\ 2 & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & -3 \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix} \\
&= \begin{bmatrix} 2xy+y^2 \\ 2x^2+xy+3y^2 \end{bmatrix}^T \begin{bmatrix} 2xy+y^2 \\ 2x^2+xy+3y^2 \end{bmatrix} \\
&= (2xy+y^2)^2+(2x^2+xy+3y^2)^2
\end{aligned}
$$

Quadratic Representation: (Using Matrix $M \in \mathbb{R}^{p \times p}$):

$$p(x) = Z_d(x)^T M Z_d(x)$$

Linear Representation: (Using Vector $c \in \mathbb{R}^q$)

$$q(x) = c^T Z_{2d}(x)$$

To constrain $p(x) = q(x)$, we write $[Z_d]_i = x^{\alpha_i}$, $[Z_{2d}]_j = x^{\beta_j}$ and reformulate

$$p(x) = Z_d(x)^T M Z_d(x) = \sum_{i,j} M_{i,j} x^{\alpha_i + \alpha_j} = \mathsf{vec}(M)^T A Z_{2d}(x)$$

where $A \in \mathbb{R}^{p^2 \times q}$ is defined as

$$A_{i,j} = \begin{cases} 1 & \text{if } \ \alpha_{\mathsf{mod}(i,p)} + \alpha_{\lfloor i \rfloor_p + 1} = \beta_j \\ 0 & \text{otherwise} \end{cases}$$

This then implies that

$$Z_d(x)^T M Z_d(x) = \mathsf{vec}(M)^T A Z_{2d}(x)$$

Hence if we constrain $c = \mathsf{vec}(M)^T A$, this is equivalent to $p(x) = q(x)$

Summarizing, e.g., for Lyapunov stability, we have variables $M > 0, Q > 0$ with the constraint

$$-\mathsf{vec}(M)^T A = \mathsf{vec}(Q)^T AB$$

Feasibility implies stability since

$$\begin{aligned} V(x) &= Z(x)^T Q Z(x) \geq 0 \\ \dot{V}(x) &= \mathsf{vec}(Q)^T A \nabla Z_{2d}(x) \\ &= \mathsf{vec}(Q)^T AB Z_{2d}(x) \\ &= -\mathsf{vec}(M)^T A Z_{2d}(x) \\ &= -Z(x)^T M Z(x) \geq 0 \end{aligned}$$

YALMIP has SOS functionality

Link: YALMIP SOS Manual

To test whether

$$4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4$$

is a positive polynomial, we use:

```
> sdpvar x y
> p = 4*x^4 + 4*x^3*y - 7*x^2*y^2 - 2*x*y^3 + 10*y^4;
> F=[];
> F=[F;sos(p)];
> solvesos(F);
```

To retrieve the SOS decomposition, we use

```
> sdisplay(p)
> ans =
>     '1.7960*x^2 - 3.0699*y^2 + 0.6468*x*y'
>     '-0.6961*x^2 - 0.7208*y^2 - 1.4882*x*y'
>     '0.5383*x^2 + 0.2377*y^2 - 0.3669*x*y'
```

In this class, we will use instead SOSTOOLS
Link: SOSTOOLS Website

To test whether

$$4x^4 + 4x^3y - 7x^2y^2 - 2xy^3 + 10y^4$$

is a positive polynomial, we use:

```
> pvar x y
> p = 4*x^4 + 4*x^3*y - 7*x^2*y^2 - 2*x*y^3 + 10*y^4;
> prog=sosprogram([x y]);
> prog=sosineq(prog,p);
> prog=sossolve(prog);
```

SOS Programming:

Numerical Example

This also works for matrix-valued polynomials.

$$M(y,z) = \begin{bmatrix} (y^2+1)z^2 & yz \\ yz & y^4+y^2-2y+1 \end{bmatrix}$$

$$\begin{aligned}
\begin{bmatrix} (y^2+1)z^2 & yz \\ yz & y^4+y^2-2y+1 \end{bmatrix} &= \begin{bmatrix} z & 0 \\ yz & 0 \\ 0 & 1 \\ 0 & y \\ 0 & y^2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & -1 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ yz & 0 \\ 0 & 1 \\ 0 & y \\ 0 & y^2 \end{bmatrix} \\
&= \begin{bmatrix} z & 0 \\ yz & 0 \\ 0 & 1 \\ 0 & y \\ 0 & y^2 \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ yz & 0 \\ 0 & 1 \\ 0 & y \\ 0 & y^2 \end{bmatrix} \\
&= \begin{bmatrix} yz & 1-y \\ z & y^2 \end{bmatrix}^T \begin{bmatrix} yz & 1-y \\ z & y^2 \end{bmatrix} \in \Sigma_s
\end{aligned}$$

SOS Programming:

Numerical Example

This also works for matrix-valued polynomials.

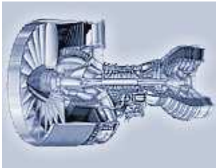
$$M(y,z) = \begin{bmatrix}(y^2+1)z^2 & yz \\ yz & y^4+y^2-2y+1\end{bmatrix}$$

SOSTOOLS Code: Matrix Positivity

```
> pvar x y
> M = [(y^2 + 1) * z^2  y * z; y * z  y^4 + y^2 - 2 * y + 1];
> prog=sosprogram([y z]);
> prog=sosmatrixineq(prog,M);
> prog=sossolve(prog);
```

An Example of Global Stability Analysis

A controlled model of a jet engine (Derived from Moore-Greitzer).



$$\dot{x} = -y - \frac{3}{2}x^2 - \frac{1}{2}x^3$$
$$\dot{y} = 3x - y$$

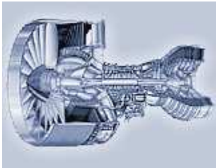
SOSTOOLS Code: Global Stability

```
> pvar x y
> f = [-y - 1.5 * x^2 - .5 * x^3; 3 * x - y];
> prog=sosprogram([x y]);
> Z=monomials([x,y],0:2);
> [prog,V]=sossosvar(prog,Z);
> V = V + .0001 * (x^4 + y^4);
> prog=soseq(prog,subs(V,[x; y],[0; 0]));
> nablaV=[diff(V,x);diff(V,y)];
> prog=sosineq(prog,-nablaV'*f);
> prog=sossolve(prog);
> Vn=sosgetsol(prog,V)
```

Finds a Lyapunov Function of degree 4.

An Example of Global Stability Analysis

A controlled model of a jet engine (Derived from Moore-Greitzer).



$$\dot{x} = -y - \frac{3}{2}x^2 - \frac{1}{2}x^3$$
$$\dot{y} = 3x - y$$

YALMIP Code: Global Stability

```
> sdpvar x y
> f = [-y - 1.5 * x^2 - .5 * x^3; 3 * x - y];
> [V,Vc]=polynomial([x y],4);
> F=[Vc(1)==0];
> F = [F; sos(V - .00001 * (x^2 + y^2))];
> nablaV=jacobian(V,[x y]);
> F=[F;sos(-nablaV*f)];
> solvesos(F,[],[],[Vc])
```

Finds a Lyapunov Function of degree 4.

- Going forward, we will use mostly SOSTOOLS

SOSOPT and DelayTOOLS

There is a third relatively new Parser called SOSOPT

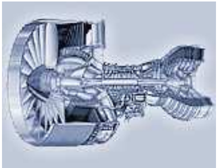
Link: SOSOPT Website

And I can plug my own mini-toolbox version of SOSTOOLS:

Link: DelayTOOLS Website

- However, I don't expect you to need this toolbox for this Lecture.

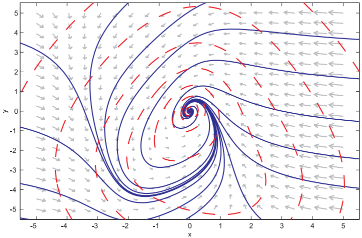
A controlled model of a jet engine (Derived from Moore-Greitzer).



$$\dot{x} = -y - \frac{3}{2}x^2 - \frac{1}{2}x^3$$
$$\dot{y} = 3x - y$$

This is feasible with

$$V(x) = 4.5819x^2 - 1.5786xy + 1.7834y^2 - 0.12739x^3 + 2.5189x^2y - 0.34069xy^2$$
$$+ 0.61188y^3 + 0.47537x^4 - 0.052424x^3y + 0.44289x^2y^2 + 0.090723y^4$$



Summary of the SOS Conditions

Proposition 1.

Suppose: $p(x) = Z_d(x)^T Q Z_d(x)$ *for some* $Q > 0$*. Then* $p(x) \geq 0$ *for all* $x \in \mathbb{R}^n$

Refinement 1: Suppose $Z_d(x)^T P Z_d(x) p(x) = Z_d(x)^T Q Z_d(x)$ for some $Q, P > 0$. Then $p(x) \geq 0$ for all $x \in \mathbb{R}^n$.

Refinement 2: Suppose $(\sum_i x_i^2)^q p(x) = Z_d(x)^T Q Z_d(x)$ for some $P > 0$, $q \in \mathbb{N}$. Then $p(x) \geq 0$ for all $x \in \mathbb{R}^n$.

Ignore these Refinements

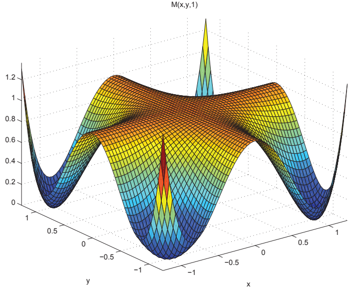
- SOS by itself is sufficient. The refinements are Necessary and Sufficient.
- Almost never necessary in practice...

Unfortunately, a Sum-of-Squares representation is not necessary for positivity.

- Artin included ratios of squares.

Counterexample: The Motzkin Polynomial

$$M(x,y) = x^2y^4 + x^4y^2 + 1 - 3x^2y^2$$



However, $(x^2+y^2+1)M(x,y)$ is a Sum-of-Squares.

$$\begin{aligned}(x^2+y^2+1)M(x,y) &= (x^2y-y)^2 + (xy^3-x)^2 + (x^2y^2-1)^2 \\ &\quad + \frac{1}{4}(xy^3-x^3y)^2 + \frac{3}{4}(xy^3+x^3y-2xy)^2\end{aligned}$$

Problems with SOS

The problem is that most nonlinear stability problems are **local**.

- Global stability requires a unique equilibrium.
- Very few nonlinear systems are globally stable.

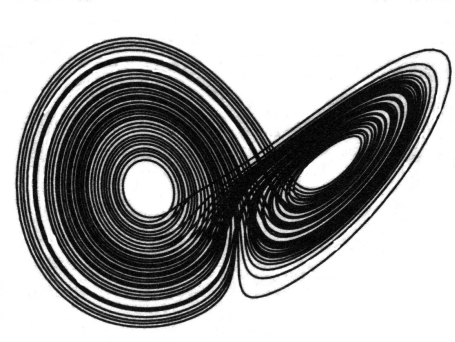


Figure: The Lorentz Attractor

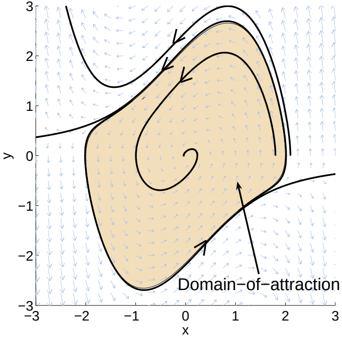


Figure: The van der Pol oscillator in reverse

Local Positivity

A more interesting question is the question of local positivity.

Question: Is $y(x) \geq 0$ for $x \in X$, where $X \subset \mathbb{R}^n$.

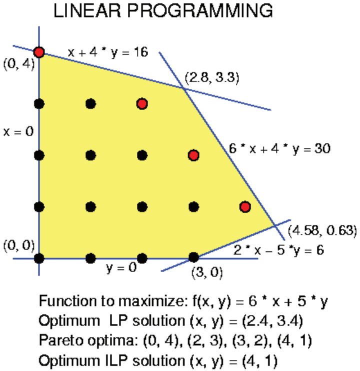
Examples:

- Matrix Copositivity:

$$y^T M y \geq 0 \qquad \text{for all } y \geq 0$$

- Integer Programming (Upper bounds)

$$\min \gamma$$
$$\gamma \geq f_i(y)$$
$$\text{for all } y \in \{-1, 1\}^n \text{ and } i = 1, \cdots, k$$



- Local Lyapunov Stability

$$V(x) \geq \|x\|^2 \qquad \text{for all } \|x\| \leq 1$$
$$\nabla V(x)^T f(x) \leq 0 \qquad \text{for all } \|x\| \leq 1$$

All these sets are **Semialgebraic**.

How are these sets represented???

Definition 7.

A set $X \subset \mathbb{R}^n$ is **Semialgebraic** if it can be represented using polynomial equality and inequality constraints.

$$X := \left\{ x : \begin{array}{ll} p_i(x) \geq 0 & i = 1, \ldots, k \\ q_j(x) = 0 & j = 1, \ldots, m \end{array} \right\}$$

If there are only equality constraints, the set is **Algebraic**.

Note: A semialgebraic set can also include $\neq$ and $<$.

Discrete Values

$$\{-1, 1\}^n = \{y \in \mathbb{R}^n \, : \, y_i^2 - 1 = 0\}$$

The Ball of Radius 1

$$\{x \, : \, \|x\| \leq 1\} = \{x \, : \, 1 - x^T x \geq 0\}$$

The *representation* of a set is **NOT UNIQUE**.

- Some representations are better than others...

Other Interesting Sets

Poisson's Equation (Courtesy of James Forbes)

Consider the dynamics of the rotation matrix on SO(3)

- Gives the orientation in the Body-fixed frame for a body rotating with angular velocity ω.

$$\dot{C} = -\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} C$$

where $C = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \\ C_7 & C_8 & C_9 \end{bmatrix} \in \mathbb{R}^{3\times 3}$ which satisfies $C^T C = I$ and $\det C = 1$.

Define

$$S := \{ \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \\ C_7 & C_8 & C_9 \end{bmatrix} \; : \; det(C) = 1, \; C^T C = I\}$$

So we would like a Lyapunov function $V(C)$ which satisfies

$$\nabla V(C)^T f(C) \le 0 \qquad \text{for all } C \text{ such that } C \in S$$

Recall the SOS Conditions

Proposition 2.

Suppose: $p(x) = Z_d(x)^T Q Z_d(x)$ *for some* $Q > 0$. *Then* $p(x) \geq 0$ *for all* $x \in \mathbb{R}^n$

Corollary 8 (S-Procedure).

$z^TFz \ge 0$ for all $z \in S := \{x \in \mathbb{R}^n \; : \; x^TGx \ge 0\}$ if there exists a scalar $\tau \ge 0$ such that $F - \tau G \succeq 0$.

This works because

- $\tau \ge 0$ and $z^TGz \ge 0$ for all $z \in S$
- Hence $\tau z^TGz \ge 0$ for all $z \in S$

If $F \ge \tau G$, then

$$\begin{aligned} z^TFz &\ge \tau z^TGz \qquad \text{for all } \; z \in \mathbb{R}^n \\ &\ge 0 \qquad \text{for all } \; z \in S \end{aligned}$$

Now Consider *Polynomials*

Proposition 3.

Suppose $\tau(x)$ is SOS ($\ge 0 \; \forall x$). If $f(x) - \tau(x)g(x)$ is SOS ($\ge 0 \; \forall x$), then

$$f(x) \ge 0 \qquad \textit{for all} \quad x \in S := \{x \; : \; g(x) \ge 0\}$$

Summary of SOS Positivity on a set

The Main Idea

Proposition 4.

Suppose $s_i(x)$ are SOS and t_i are polynomials (not necessarily positive). If

$$f(x) = s_0(x) + \sum_i s_i(x) g_i(x) + \sum_j t_j(x) h_j(x)$$

then $f(x) \geq 0$ *for all* $x \in S := \{x \,:\, g_i(x) \geq 0,\ h_i(x) = 0\}$

This works because

- $s_i(x) \geq 0$ for all $z \in S$
- $g_i(x) \geq 0$ for all $z \in S$
- $h_i(x) = 0$ for all $z \in S$

Question: Is it Necessary and Sufficient???
Answer: Yes, but only if we represent S in the *right way*.

- The Dark Art of the Positivstellensatz!

Consider how to represent a semialgebraic set:
Example: A representation of the interval $S = [a, b]$.

- A first order representation:

$$\{x \in \mathbb{R} : x - a \geq 0,\, b - x \geq 0\}$$

- A quadratic representation:

$$\{x \in \mathbb{R} : (x - a)(b - x) \geq 0\}$$

- We can add arbitrary polynomials which are PSD on X to the representation.

$$\{x \in \mathbb{R} : (x - a)(b - x) \geq 0,\, x - a \geq 0\}$$
$$\{x \in \mathbb{R} : (x^2 + 1)(x - a)(b - x) \geq 0\}$$
$$\{x \in \mathbb{R} : (x-a)(b-x) \geq 0,\, (x^2+1)(x-a)(b-x) \geq 0,\, (x - a)(b - x) \geq 0\}$$

There are infinite ways to represent the same set

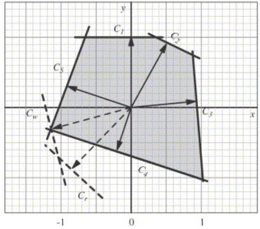
- Some Work well and others Don't!

Why are all these representations valid?

- We are adding redundant constraints to the set.
- $x - a \geq 0$ and $b - x \geq 0$ for $x \in [a, b]$ implies

$$(x - a)(b - x) \geq 0.$$

- $x^2 + 1$ is SOS, so is obviously positive on $x \in [a, b]$.



How are we creating these redundant constraints?

- **Logical Inference**
- Using existing polynomials which are positive on X to create new ones.

Note: If $f(x) \geq 0$ for $x \in S$

- So f is positive on S if and only if it is a valid constraint...

Big Question:

- Can ANY polynomial which is positive on $[a, b]$ be constructed this way?

The Cone of Inference

Definition 9.

Given a semialgebraic set S, a function f is called a **valid inequality** on S if

$$f(x) \geq 0 \qquad \text{for all } x \in S$$

Question: How to construct valid inequalities?

- Closed under addition: If f_1 and f_2 are valid, then $h(x) = f_1(x) + f_2(x)$ is valid
- Closed under multiplication: If f_1 and f_2 are valid, then $h(x) = f_1(x)f_2(x)$ is valid
- Contains all Squares: $h(x) = g(x)^2$ is valid for ANY polynomial g.

A set of inferences constructed in such a manner is called a cone.

The Cone of Inference

Definition 10.

The set of polynomials $C \subset \mathbb{R}[x]$ is called a **Cone** if

- $f_1 \in C$ and $f_2 \in C$ implies $f_1 + f_2 \in C$.
- $f_1 \in C$ and $f_2 \in C$ implies $f_1 f_2 \in C$.
- $\Sigma_s \subset C$.

Note: this is **NOT** the same definition as in optimization.

The Cone of Inference

The set of inferences is a cone

Definition 11.

For any set, S, the cone $C(S)$ is the set of polynomials PSD on S

$$C(S) := \{f \in \mathbb{R}[x] \,:\, f(x) \geq 0 \text{ for all } x \in S\}$$

The big question: how to test $f \in C(S)$???

Corollary 12.

$f(x) \geq 0$ for all $x \in S$ if and only if $f \in C(S)$

The Monoid

Suppose S is a semialgebraic set and define its *monoid*.

Definition 13.

For given polynomials $\{f_i\} \subset \mathbb{R}[x]$, we define $\texttt{monoid}(\{f_i\})$ as the set of all products of the f_i

$$\texttt{monoid}(\{f_i\}) := \{h \in \mathbb{R}[x] : h(x) = \prod f_1^{a_1}(x) f_2^{a_k}(x) \cdots f_k^{a_2}(x),\, a \in \mathbb{N}^k\}$$

- $1 \in \texttt{monoid}(\{f_i\})$
- $\texttt{monoid}(\{f_i\})$ is a subset of the cone defined by the f_i.
- The monoid does not include arbitrary sums of squares

The Cone of Inference

If we combine $\texttt{monoid}(\{f_i\})$ with Σ_s, we get $\texttt{cone}(\{f_i\})$.

Definition 14.

For given polynomials $\{f_i\} \subset \mathbb{R}[x]$, we define $\texttt{cone}(\{f_i\})$ as

$$\texttt{cone}(\{f_i\}) := \{h \in \mathbb{R}[x] : h = \sum s_i g_i,\ g_i \in \texttt{monoid}(\{f_i\}),\ s_i \in \Sigma_s\}$$

If

$$S := \{x \in \mathbb{R}^n \,:\, f_i(x) \geq 0,\ i = 1 \cdots, k\}$$

$\texttt{cone}(\{f_i\}) \subset C(S)$ is an approximation to $C(S)$.

- The key is that it is possible to test whether $f \in \texttt{cone}(\{f_i\}) \subset C(S)$!!!
  - Sort of... (need a degree bound)
  - Use e.g. SOSTOOLS

More on Inference

Corollary 15.

$h \in \textit{cone}(\{f_i\}) \subset C(S)$ if and only if there exist $s_i, r_{ij}, \cdots \in \Sigma_s$ such that

$$h(x) = s_0 + \sum_i s_i f_i + \sum_{i \neq j} r_{ij} f_i f_j + \sum_{i \neq j \neq k} r_{ijk} f_i f_j f_k + \cdots$$

Note we must include all possible combinations of the f_i

- A finite number of variables s_i, r_{ij}.
- $s_i, r_{ij} \in \Sigma_s$ is an SDP constraint.

This gives a sufficient condition for $h(x) \geq 0$ for all $x \in S$.

- Can be tested using, e.g. SOSTOOLS

Numerical Example

Example: Show that $h(x) = 5x - 9x^2 + 5x^3 - x^4$ is PSD on the interval

$$[0, 1] = \{x \in \mathbb{R}^n \,:\, f_1(x) = x(1 - x) \geq 0\},$$

A single inequality $f_1(x) = x(1 - x)$. The cone `cone(f1)` only has 2 terms

$$s_0(x) + x(1 - x)s_1(x)$$

We find $f \in$ `cone(f1)` using $s_0(x) = 0$, $s_1(x) = (2 - x)^2 + 1$ so that

$$h(x) = 5x - 9x^2 + 5x^3 - x^4 = 0 + ((2 - x)^2 + 1)x(1 - x)$$

Which is a certificate of non-negativity of h on $S = [0, 1]$

Note: the original representation of S matters:

- If we had used $S = \{x \in \mathbb{R} \,:\, x \geq 0,\, 1 - x \geq 0\}$, then we would have had 4 SOS variables

$$h(x) = s_0(x) + xs_1(x) + (1 - x)s_2(x) + x(1 - x)s_3(x)$$

The complexity can be *decreased* through judicious choice of representation.

Stengle's Positivstellensatz

We have two big questions

- How close an approximation is $\texttt{cone}(\{f_i\}) \subset C(S)$ to $C(S)$?
  - Cannot always be exact since not every positive polynomial is SOS.
- Can we reduce the complexity?

Both these questions are answered by *Positivstellensatz* Results. Recall

$$S := \{x \in \mathbb{R}^n \,:\, f_i(x) \geq 0,\, i = 1 \cdots, k\}$$

Theorem 16 (Stengle's Positivstellensatz).

$S = \emptyset$ if and only if $-1 \in \texttt{cone}(\{f_i\})$. That is, $S = \emptyset$ if and only if there exist $s_i, r_{ij}, \cdots \in \Sigma_s$ such that

$$-1 = s_0 + \sum_i s_i f_i + \sum_{i \neq j} r_{ij} f_i f_j + \sum_{i \neq j \neq k} r_{ijk} f_i f_j f_k + \cdots$$

Note that this is not exactly what we were asking.

- We would prefer to know whether $h \in \texttt{cone}(\{f_i\})$
- Difference is important for reasons of convexity.

Problem: We want to know whether $f(x) > 0$ for all $x \in \{x : g_i(x) \geq 0\}$.

Corollary 17 (Stengle's Positivstellensatz).

$f(x) > 0$ for all $x \in \{x : g_i(x) \geq 0\}$ if and only if there exist $s_i, q_{ij}, r_{ij}, \dots \in \Sigma_s$ such that

$$f\left(s_{-1} + \sum_i q_i g_i + \sum_{i \neq j} q_{ij} g_i g_j + \sum_{i \neq j \neq k} q_{ijk} g_i g_j g_k + \cdots\right) = 1 + s_0 + \sum_i s_i g_i + \sum_{i \neq j} r_{ij} g_i g_j + \sum_{i \neq j \neq k} r_{ijk} g_i g_j g_k + \cdots$$

We have to include all possible combinations of the g_i!!!!

- But assumes **Nothing** about the g_i
- The worst-case scenario
- Also bilinear in s_i and f (Can't search for both)

We can do better if we choose our g_i more carefully!

Stengle's Weak Positivstellensatz

Non-Negativity: Considers whether $f(x) \geq 0$ for all $x \in \{x : g_i(x) \geq 0\}$.

Corollary 18 (Stengle's Positivstellensatz).

$f(x) \geq 0$ for all $x \in \{x : g_i(x) \geq 0\}$ if and only if there exist $s_i, q_{ij}, r_{ij}, \dots \in \Sigma_s$ and $q \in \mathbb{N}$ such that

$$f\left(s_{-1} + \sum_i q_i g_i + \sum_{i \neq j} q_{ij} g_i g_j + \sum_{i \neq j \neq k} q_{ijk} g_i g_j g_k + \cdots\right)$$
$$= f^{2q} + s_0 + \sum_i s_i g_i + \sum_{i \neq j} r_{ij} g_i g_j + \sum_{i \neq j \neq k} r_{ijk} g_i g_j g_k + \cdots$$

Lyapunov Functions are **NOT** strictly positive!

- The only P-Satz to deal with functions not *Strictly* Positive.

Schmudgen's Positivstellensatz

If the set S is closed, bounded, then the problem can be simplified.

Theorem 19 (Schmüdgen's Positivstellesatz).

Suppose that $S = \{x : g_i(x) \geq 0,\, h_i(x) = 0\}$ *is compact. If* $f(x) > 0$ *for all* $x \in S$*, then there exist* $s_i, r_{ij}, \dots \in \Sigma_s$ *and* $t_i \in \mathbb{R}[x]$ *such that*

$$f = 1 + \sum_j t_j h_j + s_0 + \sum_i s_i g_i + \sum_{i \neq j} r_{ij} g_i g_j + \sum_{i \neq j \neq k} r_{ijk} g_i g_j g_k + \cdots$$

Note that Schmudgen's Positivstellensatz is essentially the same as Stengle's except for a single term.

- Now we can include both f and s_i, r_{ij} as variables.
- Reduces the number of variables substantially.

The complexity is still high (Lots of SOS multipliers).

Putinar's Positivstellensatz

If the semialgebraic set is P-Compact, then we can improve the situation further.

Definition 20 (P-Compact).

We say that $f_i \in \mathbb{R}[x]$ for $i = 1, \ldots, n_K$ define a **P-compact** set K_f, if there exist $h \in \mathbb{R}[x]$ and $s_i \in \Sigma_s$ for $i = 0, \ldots, n_K$ such that the level set $\{x \in \mathbb{R}^n : h(x) \geq 0\}$ is compact and such that the following holds.

$$h(x) - \sum_{i=1}^{n_K} s_i(x) f_i(x) \in \Sigma_s$$

The condition that a region be P-compact may be difficult to verify. However, some important special cases include:

- Any region K_f such that all the f_i are linear.
- Any region K_f defined by f_i such that there exists some i for which the superlevel set $\{x : f_i(x) \geq 0\}$ is compact.

P-Compact is not hard to satisfy.

Corollary 21.

Any compact set can be made P-compact by inclusion of a redundant constraint of the form $f_i(x) = \beta - x^T x$ for sufficiently large β.

Thus P-Compact is a property of the *representation* and not the set.

Example: The interval $[a, b]$.

- Not Obviously P-Compact:

$$\{x \in \mathbb{R} : x^2 - a^2 \geq 0,\, b - x \geq 0\}$$

- P-Compact:

$$\{x \in \mathbb{R} : (x-a)(b-x) \geq 0\}$$

Putinar's Positivstellensatz

If S is P-Compact, Putinar's Positivstellensatz dramatically reduces the complexity

Theorem 22 (Putinar's Positivstellesatz).

Suppose that $S = \{x : g_i(x) \geq 0,\, h_i(x) = 0\}$ is P-Compact. If $f(x) > 0$ for all $x \in S$, then there exist $s_i \in \Sigma_s$ and $t_i \in \mathbb{R}[x]$ such that

$$f = s_0 + \sum_i s_i g_i + \sum_j t_j h_j$$

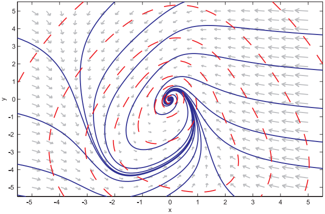
A single multiplier for each constraint.

- We are back to the original condition
- A Good representation of the set is P-compact

We can now recast the search for a Lyapunov function.

Let

$$X := \left\{ x : p_i(x) \geq 0 \quad i = 1, \ldots, k \right\}$$



Theorem 23.

Suppose there exists a V, an $\epsilon > 0$, and $s_0, s_i, t_0, t_i \in \Sigma_s$ such that

$$V(x) = s_0(x) + \sum_i s_i(x) p_i(x) + \epsilon\, x^T x$$

$$-\dot{V}(x) = -\nabla V(x)^T f(x) = t_0(x) + \sum_i t_i(x) p_i(x) + \epsilon\, x^T x$$

Then the system is exponentially stable on any $Y_\gamma := \{x : v(x) \leq \gamma\}$ where $Y_\gamma \subset X$.

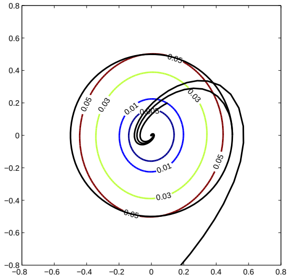
Note: Find the largest Y_γ via bisection.

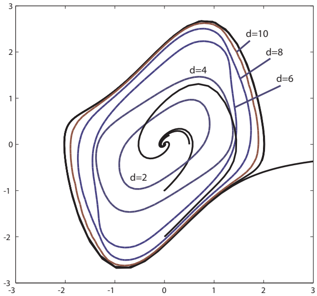
Van-der-Pol Oscillator

$$\dot{x}(t) = -y(t)$$
$$\dot{y}(t) = -\mu(1 - x(t)^2)y(t) + x(t)$$

Procedure:

1. Use Bisection to find the largest ball on which you can find a Lyapunov function.
2. Use Bisection to find the largest level set of that Lyapunov function on which you can find a Lyapunov function. Repeat





First, Find the Lyapunov function

SOSTOOLS Code: Find a Local Lyapunov Function

```
> pvar x y
> mu=1; r=2.8;
> g = r - (x^2 + y^2);
> f = [-y; -mu * (1 - x^2) * y + x];
> prog=sosprogram([x y]);
> Z2=monomials([x y],0:2);
> Z4=monomials([x y],0:4);
> [prog,V]=sossosvar(prog,Z2);
> V = V + .0001 * (x^4 + y^4);
> prog=soseq(prog,subs(V,[x, y]',[0, 0]'));
> nablaV=[diff(V,x);diff(V,y)];
> [prog,s]=sossosvar(prog,Z2);
> prog=sosineq(prog,-nablaV'*f-s*g);
> prog=sossolve(prog);
> Vn=sosgetsol(prog,V)
```

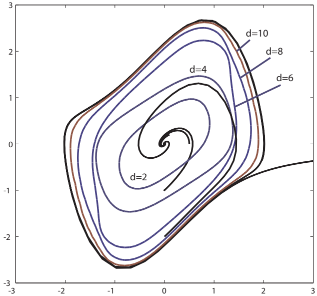
This finds a Lyapunov function which is decreasing on the ball of radius $\sqrt{2.8}$

- Lyapunov function is of degree 4.

Local Stability Analysis

Next find the largest level set which is contained in the ball of radius $\sqrt{2.8}$.

```
> pvar x y
> gamma=6.6;
> Vg=gamma-Vn;
```
> $\mathrm{g} = \mathrm{r} - (\mathrm{x}^2 + \mathrm{y}^2);$
```
> prog=sosprogram([x y]);
> Z2=monomials([x y],0:2);
> [prog,s]=sossosvar(prog,Z2);
> prog=sosineq(prog,g-s*Vg);
> prog=sossolve(prog);
```



In this case, the maximum γ is 6.6

- Estimate of the DOA will increase with degree of the variables.

Making Sense of Positivity Constraints

$$-\dot{V}(x) - g(x) \cdot s(x) \geq 0 \qquad \forall x$$

means

$$\dot{V}(x) \leq -g(x) \cdot s(x) \leq 0$$

when $g(x) \geq 0$ (since $s(x) \geq 0$ and $g(x) \geq 0$ on $x \in X$).

- but $\|x\|^2 \leq r$ implies $g(x) \geq 0$
- hence $\dot{V}(x) \leq 0$ for all $x \in B_{\sqrt{r}}$

Likewise

$$g(x) - s(x) \cdot (\gamma - V(x)) \geq 0 \qquad \forall x$$

means

$$g(x) \geq s(x) \cdot (\gamma - V(x)) \geq 0$$

whenever $V(x) \leq \gamma$.

- So $g(x) \geq 0$ whenever $x \in V_\gamma$
- But $g(x) \geq 0$ means $\|x\| \leq \sqrt{r}$
- So if $x \in V_\gamma$, then $g(x) \geq 0$ and hence $\|x\| \leq \sqrt{r}$.
- So $V_\gamma \subset B_{\sqrt{r}}$

An Example of Global Stability Analysis

SOSTOOLS Code: Globally Stabilizing Controller

```
> pvar w1 w2 w3
> J1=2;J2=1;J3=1;
> k1=1;k2=1;k3=1;
> u1=-k1*w1;u2=-k2*w2;u3=-k3*w3;
> f = [(J2 - J3)/J1 * w2 * w3 + u1;
> (J3 - J1)/J2 * w3 * w1 + u2;
> (J1 - J2)/J3 * w1 * w2 + u3];
> prog=sosprogram([w1 w2 w3]);
> Z=monomials([w1 w2 w3],1:2);
> [prog,V]=sossosvar(prog,Z);
> V = V + .0001 * (w1^4 + w2^4 + w3^4);
> prog=soseq(prog,subs(V,[w1; w2; w3],[0; 0;
0]));
> nablaV=[diff(V,w1);diff(V,w2);diff(V,w3)];
> prog=sosineq(prog,-nablaV'*f-4.0*V);
> prog=sossolve(prog);
> Vn=sosgetsol(prog,V)
```

$$J_1\dot{\omega}_1 = (J_2 - J_3)\omega_2\omega_3 + u_1$$
$$J_2\dot{\omega}_2 = (J_3 - J_1)\omega_3\omega_1 + u_2$$
$$J_3\dot{\omega}_3 = (J_1 - J_2)\omega_1\omega_2 + u_3$$
$$u_1 = -k_1\omega_1$$
$$u_2 = -k_2\omega_2$$
$$u_3 = -k_3\omega_3$$

This is feasible and proves exponential stability with decay rate $\gamma = 4$

An Example of Globally Stabilizing Controller Synthesis

SOSTOOLS Code: Globally Stabilizing Controller

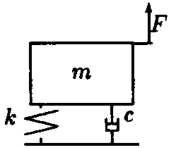
```
> pvar x1 x2 x3
> prog=sosprogram([x1 x2 x3]);
> Z4=monomials([x1 x2 x3],0:3);
> Z2=monomials([x1 x2 x3],0:3);
> [prog,k1]=sospolyvar(prog,Z4);
> [prog,k2]=sospolyvar(prog,Z4);
> u1=k1; u2=k2;
> f=[-x1+x2-x3;-x1*x3-x2+u1;-x1+u2];
>  V = x1^2 + x2^2 + x3^2;
> prog=soseq(prog,subs(V,[x1, x2, x3]',[0,
0, 0]'));
> nablaV=[diff(V,x1);diff(V,x2);diff(V,x3)];
> prog=sosineq(prog,-(nablaV'*f));
> prog=sossolve(prog);
> k1n=sosgetsol(prog,k1)
> k2n=sosgetsol(prog,k2)
```

$$\dot{x}_1 = -x_1 + x_2 - x_3$$
$$\dot{x}_2 = -x_1 x_3 - x_2 + u_1$$
$$\dot{x}_3 = -x_1 + u_2$$

Find $u_1(t) = k_1(x(t))$, $u_2(t) = k_2(x(t))$

Example of Parametric Uncertainty

Recall The Spring-Mass Example



$$\ddot{y}(t) + c\dot{y}(t) + \frac{k}{m}y(t) = \frac{F(t)}{m}$$

Multiplicative Uncertainty

- $m \in [m_-, m_+]$
- $c \in [c_-, c_+]$
- $k[k_-, k_+]$

Questions:

- Can we do robust optimal control without the LFT framework??
- Consider static uncertainty?
  - Can we do better than Quadratic Stabilization??

General Formulation

$$\dot{x} = A(\delta)x(t) + B(\delta)u(t)$$
$$y(t) = C(\delta)x(t) + D(\delta)u(t)$$

Lets Start with Stability with Static Uncertainty

General Formulation

$$\dot{x}(t) = A(\delta)x(t) + B(\delta)u(t)$$
$$y(t) = C(\delta)x(t) + D(\delta)u(t)$$

Where A, B, C, D are rational (denominators $d(\delta) > 0$ for all $\delta \in \Delta$)

Theorem 24.

Suppose there exists $P(\delta) - \epsilon I \geq 0$ for all $\delta \in \Delta$ and such that

$$A(\delta)^T P(\delta) + P(\delta)A(\delta) \leq 0 \qquad \text{for all } \; \delta \in \Delta$$

Then $A(\delta)$ is Hurwitz for all $\delta \in \Delta$.

Theorem 25.

Suppose there exists $s_i, r_i \in \Sigma_s$ such that $P(\delta) = s_0(\delta) + \sum_i s_i(\delta)g_i(\delta)$ and

$$-A(\delta)^T P(\delta) - P(\delta)A(\delta) = r_0(\delta) + \sum_i r_i(\delta)g_i(\delta)$$

Then $A(\delta)$ is Hurwitz for all $\delta \in \{\delta \, : \, g_i(\delta) \geq 0\}$.

Proof: Use $V(x) = x^T P(\delta)x$.

Lets Start With Stability

Apply this to The Spring-Mass Example

$$\ddot{y}(t) = -c\dot{y}(t) - \frac{k}{m}y(t) = \frac{F(t)}{m}$$
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -c & -\frac{k}{m} \end{bmatrix}}_{A(c,k,m)} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$$

Semi-Algebraic Form:

- $g_1(m) = (m - m_-)(m_+ - m) \geq 0$
- $g_2(c) = (c - c_-)(c_+ - c) \geq 0$
- $g_3(k) = (k - k_-)(k_+ - k) \geq 0$

We are searching for a P, $s_i, r_i \in \Sigma_s$ such that

$$P(c,k,m) = s_0(c,k,m) + s_1(c,k,m)g_i(m) + s_2(c,k,m)g_2(c) + s_3(c,k,m)g_3(k)$$

such that

$$\begin{aligned} &-mA(c,k,m)^T P(c,k,m) - P(c,k,m)mA(c,k,m) \\ &= m(r_0(c,k,m) + r_1(c,k,m)g_i(m) + r_2(c,k,m)g_2(c) + r_3(c,k,m)g_3(k)) \end{aligned}$$

SOSTOOLS does not work with Matrix-Valued Problems

You should instead download SOSMOD

SOSMOD_vMAE598 is my personal toolbox and is compatible with the code presented in these lecture notes.

- May have issues with versions of Matlab 2016a and later. Working to correct these.
- Folder Must be added to the Matlab PATH
- Also contains example scripts for the code listed in the lecture notes.

Link: SOSMOD for download

- Also on Code Ocean

SOSTOOLS Code for Robust Stability Analysis

```
> pvar m c k
> Am=[0 m;-c*m -k];
> mmin=.1;mmax=1;cmin=.1;cmax=1;kmin=.1;kmax=1;
> g1=(mmax-m)(m-mmin);g2=(cmax-c)(c-cmin);g3=(kmax-k)(k-kmin);
> vartable=[m c k];
> prog=sosprogram(vartable);
> [prog,S0]=sosposmatrvar(prog,2,4,vartable);
> [prog,S1]=sosposmatrvar(prog,2,4,vartable);
> [prog,S2]=sosposmatrvar(prog,2,4,vartable);
> [prog,S3]=sosposmatrvar(prog,2,4,vartable);
> P=S0+g1*S1+g2*S2+g3*S3+.00001*eye(2);
> [prog,R1]=sosposmatrvar(prog,2,4,vartable);
> [prog,R2]=sosposmatrvar(prog,2,4,vartable);
> [prog,R3]=sosposmatrvar(prog,2,4,vartable);
> [prog,R4]=sosposmatrvar(prog,2,4,vartable);
> constr=-(Am'*P+P*Am)-m*(R0+R1*g1+R2*g2+R3*g3);
> prog=sosmateq(prog,constr);
> prog=sossolve(prog);
> Pn=sosgetsol(prog,P)
```

Now we can do Time-Varying Uncertainty

Time-Varying Formulation:

$$\dot{x}(t) = A(\delta(t))x(t) + B(\delta(t))u(t) \qquad \delta(t) \in \Delta_1$$

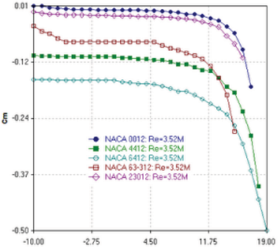
$$y(t) = C(\delta(t))x(t) + D(\delta(t))u(t) \qquad \dot{\delta}(t) \in \Delta_2$$

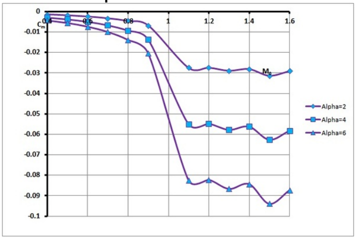
Simple Example: Angle of attack (α)

$$\dot{\alpha}(t) = -\frac{\rho(t)v(t)^2 c_\alpha(\alpha(t), M(t))}{2I}\alpha(t)$$

The time-varying parameters are:

- velocity, v and Mach number, M (M depends on Reynolds $\#$);
- density of air, ρ;
- Also, we sometimes treat α itself as an uncertain parameter.





Exponential Stability with Time-Varying Uncertainty

$$\dot{x}(t) = A(\delta(t))x(t)$$

Theorem 26.

Suppose there exists $P(\delta) - \epsilon I \geq 0$ for all $\delta \in \Delta$ and such that

$$A(\delta)^T P(\delta) + P(\delta)A(\delta) + \sum_i \frac{\partial}{\partial \delta_i} P(\delta)\dot{\delta}_i \leq 0 \qquad \text{for all } \; \delta \in \Delta_2, \quad \dot{\delta} \in \Delta_2$$

Then $\dot{x}(t) = A(\delta(t))x(t)$ is exponentially stable.

Proof: Use $V(t,x) = x^T P(\delta(t))x$.

- Treat δ_i and $\dot{\delta}_i$ as independent (Usually not conservative).
- If $\Delta_2 = \mathbb{R}^n$, then requires $\frac{\partial}{\partial \delta_i} P(\delta) = 0$ (Quadratic Stability).

Example: Gain Scheduling Choose K_i based on δ

$$\dot{x}(t) = \begin{cases} (A(\delta) + BK_i)x(t) & \delta \in \Delta_i \end{cases}$$

No Bound on rate of variation! $(\Delta_2 = \mathbb{R}^n)$

- Unless δ depends on x....

Extension to Optimal Controller Synthesis

We have two cases

- Time-Varying Parametric Uncertainty $\dot{x}(t) = A(\delta(t))x(t)$
- Static Parametric Uncertainty $\dot{x}(t) = A(\delta)x(t)$

Most of the LMIs in this course can be adapted to either case using the Positivstellensatz.

- Need to be careful with TV uncertainty, however.

Popular Uses:

- H_2 optimal control with uncertainty
  - Makes H_2 robust (H_∞ is already robust to some extent).
  - NOT RIGOROUS when $\delta(t)$ is time-varying.
- Robust Kalman Filtering
  - The Kalman Filter is not always stable in closed-Loop...

H_2-optimal robust control

Static Formulation

$$\dot{x}(t) = A(\delta)x(t) + B(\delta)u(t)$$
$$y(t) = C(\delta)x(t) + D(\delta)u(t)$$

H_2**-optimal State Feedback Synthesis**

Theorem 27.

Suppose $\hat{P}(s,\delta) = C(\delta)(sI - A(\delta))^{-1}B(\delta)$. *Then the following are equivalent.*

1. $\|S(K(\delta), P(\delta))\|_{H_2} < \gamma$ *for all* $\delta \in \Delta$..
2. $K(\delta) = Z(\delta)X(\delta)^{-1}$ *for some* $Z(\delta)$ *and* $X(\delta)$ *such that* $X(\delta) > 0$ *for all* $\delta \in \Delta$ *and*

$$\begin{bmatrix} A(\delta) & B_2(\delta) \end{bmatrix} \begin{bmatrix} X(\delta) \\ Z(\delta) \end{bmatrix} + \begin{bmatrix} X(\delta) & Z(\delta)^T \end{bmatrix} \begin{bmatrix} A(\delta)^T \\ B(\delta)_2^T \end{bmatrix} + B_1(\delta)B_1(\delta)^T < 0$$

$$\begin{bmatrix} X(\delta) & (C_1(\delta)X(\delta) + D_{12}(\delta)Z(\delta))^T \\ C_1(\delta)X(\delta) + D_{12}(\delta)Z(\delta) & W(\delta) \end{bmatrix} > 0$$

$$\mathit{Trace} W(\delta) < \gamma^2$$

for all $\delta \in \Delta$.

The KYP Lemma with Time-Varying Uncertainty

Lemma 28.

Suppose

$$G(\delta(t)) = \left[\begin{array}{c|c} A(\delta(t)) & B\delta(t) \\ \hline C\delta(t) & D\delta(t) \end{array}\right].$$

Then $\|G(\delta(t))\|_{\mathcal{L}(L_2)} \le \gamma$ *for all* $\delta(t)$ *with* $\delta(t) \in \Delta_1$ *and* $\dot\delta(t) \in \Delta_2$ *if there exists a* $X(\delta)$ *such that* $X(\delta) > 0$ *for all* $\delta \in \Delta_1$ *and*

$$\begin{bmatrix} A(\delta)^T X(\delta) + X(\delta)A(\delta) + \sum_i \beta_i \frac{\partial}{\partial \delta_i} X(\delta) & X(\delta)B(\delta) \\ B(\delta)^T X(\delta) & -\gamma I \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} C(\delta)^T \\ D(\delta)^T \end{bmatrix}\begin{bmatrix} C(\delta) & D(\delta) \end{bmatrix} < 0$$

for all $\delta \in \Delta_1$ *and* $\beta \in \Delta_2$.

The KYP Lemma with Time-Varying Uncertainty

$$\begin{aligned} \dot{x}(t) &= A(\delta(t))x(t) + B(\delta(t))u(t) \qquad && \delta(t) \in \Delta_1 \\ y(t) &= C(\delta(t))x(t) + D(\delta(t))u(t) \qquad && \dot{\delta}(t) \in \Delta_2 \end{aligned}$$

Proof.

Let $V(x,t) = x^T X(\delta(t))x$. Then

$$\begin{aligned} &\dot{V}(x(t),t) - (\gamma - \epsilon)\|u(t)\|^2 + \frac{1}{\gamma}\|y(t)\|^2 < 0 \\ &= \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \left[\begin{bmatrix} A(\delta)^T X(\delta) + X(\delta)A(\delta) + \sum_i \dot{\delta}_i \frac{\partial}{\partial \delta_i} X(\delta) & X(\delta)B(\delta) \\ B(\delta)^T X(\delta) & -(\gamma - \epsilon)I \end{bmatrix} \right. \\ &\qquad\qquad \left. + \frac{1}{\gamma} \begin{bmatrix} C(\delta)^T \\ D(\delta)^T \end{bmatrix} \begin{bmatrix} C(\delta) & D(\delta) \end{bmatrix} \right] \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \\ &\leq 0 \end{aligned}$$

□

H_∞-optimal robust control with Time-Varying Uncertainty

However, Controller Synthesis is a Problem!

- Schur Complement Still works.
- **Duality Doesn't work**.

Lemma 29.

Suppose

$$G(\delta(t)) = \left[\begin{array}{c|c} A(\delta(t)) & B(\delta(t)) \\ \hline C(\delta(t)) & D(\delta(t)) \end{array}\right].$$

Then $\|G(\delta(t))\|_{\mathcal{L}(L_2)} \le \gamma$ *for all* $\delta(t)$ *with* $\delta(t) \in \Delta_1$ *and* $\dot{\delta}(t) \in \Delta_2$ *if there exists a* $X(\delta)$ *such that* $X(\delta) > 0$ *for all* $\delta \in \Delta_1$ *and*

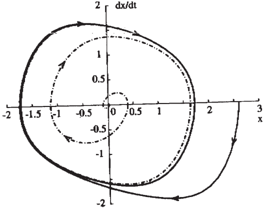
$$\begin{bmatrix} (A(\delta)+B_2(\delta)K(\delta))^T X(\delta) + X(\delta)(A(\delta)+B_2(\delta)K(\delta)) + \sum_i \beta_i \frac{\partial}{\partial \delta_i} X(\delta) & *^T & *^T \\ B_1(\delta)^T X(\delta) & -\gamma I & *^T \\ C_1(\delta) + D_{12}(\delta)K(\delta) & D_{11}(\delta) & -\gamma I \end{bmatrix} < 0$$

for all $\delta \in \Delta_1$ *and* $\beta \in \Delta_2$.

We fall back on iterative methods (Similar to D-K iteration)

- Optimize P, then optimize K.
- rinse and repeat.

The Rayleigh Equation:



$$\ddot{y} - 2\zeta(1 - \alpha\dot{y}^2)\dot{y} + y = u$$

Uncertainty:

$$\zeta \in [1.8, 2.2]$$
$$\alpha \in [.8, 1.2]$$

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}2\zeta(1-\alpha x_1^2)x_1 + x_2\\ x_1\end{bmatrix}$$

Find a Lyapunov Function: $V(y, \dot{y}, \alpha, \zeta)$

$$V(x_1, x_2, \alpha, \zeta) \geq .01 * (x_1^2 + x_2^2) \qquad \forall x \in B_r, \quad \alpha, \zeta \in \Delta$$

and $V(0, 0, \alpha, \zeta) = 0$ and

$$\nabla_x V(x_1, x_2, \alpha, \zeta)^T f(x_1, x_2, \alpha, \zeta) \leq 0 \qquad \forall x \in B_r, \quad \alpha, \zeta \in \Delta$$

SOSTOOLS Code for Robust Nonlinear Stability Analysis

```
> pvar x1 x2 z a
>  zmin = .8; zmax = 1.2; amin = 1.8; amax = 2.2; g1 = r - (x1^2 + x2^2);
>  r = .3; g2 = (amax - a)(a - amin); g3 = (zmax - z)(z - zmin);
>  f = [2 * z * (1 - a * x2^2) * x2 - x1; x1];
> vartable=[x1 x2 a z];
> prog=sosprogram(vartable);
> Z1=monomials(vartable,0:1); Z2=monomials(vartable,0:2);
> Z3=monomials(vartable,0:3);
> [prog,V0]=sossosvar(prog,Z2);
> [prog,r1]=sossosvar(prog,Z1); [prog,r2]=sossosvar(prog,Z1);
> [prog,r3]=sossosvar(prog,Z1);
>  V = V0 + .001 * (x1^2 + x2^2) + g1 * r1 + g2 * r2 + g3 * r3;
> prog=soseq(prog,subs(V,[x1, x2]',[0, 0]'));
> nablaV=[diff(V,x1);diff(V,x2)];
> P=S0+g1*S1+g2*S2+g3*S3+.00001*eye(2);
> [prog,s1]=sossosvar(prog,Z2); [prog,s2]=sossosvar(prog,Z2);
> [prog,s3]=sossosvar(prog,Z2);
> prog=sosineq(prog,-nablaV'*f-s1*g1-s2*g2-s3*g3);
> prog=sossolve(prog);
```

Integer Programming Example

MAX-CUT

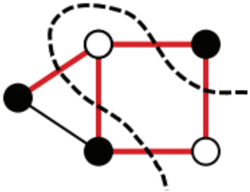


Figure: Division of a set of nodes to maximize the weighted cost of separation

Goal: Assign each node i an index $x_i = -1$ or $x_j = 1$ to maximize overall cost.

- The cost if x_i and x_j do not share the same index is w_{ij}.
- The cost if they share an index is 0
- The weight $w_{i,j}$ are given.
- Thus the total cost is

$$\frac{1}{2}\sum_{i,j} w_{i,j}(1 - x_i x_j)$$

MAX-CUT

The optimization problem is the integer program:

$$\max_{x_i^2=1} \frac{1}{2} \sum_{i,j} w_{i,j}(1 - x_i x_j)$$

The MAX-CUT problem can be reformulated as

$$\min \gamma :$$

$$\gamma \geq \max_{x_i^2=1} \frac{1}{2} \sum_{i,j} w_{i,j}(1 - x_i x_j) \quad \text{for all} \quad x \in \{x \,:\, x_i^2 = 1\}$$

We can compute a bound on the max cost using the Nullstellensatz

$$\min_{p_i \in \mathbb{R}[x],\, s_0 \in \Sigma_s} \gamma :$$

$$\gamma - \frac{1}{2} \sum_{i,j} w_{i,j}(1 - x_i x_j) + \sum_i p_i(x)(x_i^2 - 1) = s_0(x)$$

MAX-CUT

Consider the MAX-CUT problem with 5 nodes

$$w_{12} = w_{23} = w_{45} = w_{15} = .5 \quad \text{and} \quad w_{14} = w_{24} = w_{25} = w_{34} = 0$$

where $w_{ij} = w_{ji}$. The objective function is

$$f(x) = 2.5 - .5x_1x_2 - .5x_2x_3 - .5x_3x_4 - .5x_4x_5 - .5x_1x_5$$

We use SOSTOOLS and bisection on γ to solve

$$\min_{p_i \in \mathbb{R}[x],\, s_0 \in \Sigma_s} \gamma :$$
$$\gamma - f(x) + \sum_i p_i(x)(x_i^2 - 1) = s_0(x)$$

We achieve a least upper bound of $\gamma = 4$.
However!

- we don't know if the optimization problem achieves this objective.
- Even if it did, we could not recover the values of $x_i \in [-1, 1]$.

MAX-CUT

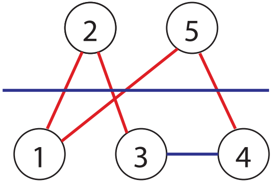


Figure: A Proposed Cut

Upper bounds can be used to VERIFY optimality of a cut.

We Propose the Cut

- $x_1 = x_3 = x_4 = 1$
- $x_2 = x_5 = -1$

This cut has objective value

$$f(x) = 2.5 - .5x_1x_2 - .5x_2x_3 - .5x_3x_4 - .5x_4x_5 - .5x_1x_5 = 4$$

Thus verifying that the cut is optimal.

MAX-CUT code

```
 pvar x1 x2 x3 x4 x5;
vartable = [x1; x2; x3; x4; x5];
prog = sosprogram(vartable);

gamma = 4;
f = 2.5 - .5*x1*x2 - .5*x2*x3 - .5*x3*x4 - .5*x4*x5 - .5*x5*x1;

bc1 = x1^2 - 1 ;
bc2 = x2^2 - 1 ;
bc3 = x3^2 - 1 ;
bc4 = x4^2 - 1 ;
bc5 = x5^2 - 1 ;

for i = 1:5
[prog, p{1+i}] = sospolyvar(prog,Z);
end;

expr = (gamma-f)+p{1}*bc1+p{2}*bc2+p{3}*bc3+p{4}*bc4+p{5}*bc5;

prog = sosineq(prog,expr);
prog = sossolve(prog);
```

The Structured Singular Value

For the case of structured parametric uncertainty, we define the structured singular value.

$$\mathbf{\Delta} = \{\Delta = \operatorname{diag}(\delta_1 I_{n1}, \cdots, \delta_s I_{ns} \, : \, \delta_i \in \mathbb{R}\}$$

- δ_i represent unknown parameters.

Definition 30.

Given system $M \in \mathcal{L}(L_2)$ and set $\mathbf{\Delta}$ as above, we define the **Structured Singular Value** of $(M, \mathbf{\Delta})$ as

$$\mu(M, \mathbf{\Delta}) = \frac{1}{\inf_{\substack{\Delta \in \mathbf{\Delta} \\ I - M\Delta \text{ is singular}}} \|\Delta\|}$$

The fundamental inequality we have is $\mathbf{\Delta}_\gamma = \{\operatorname{diag}(\delta_i), : \sum_i \delta_i^2 \leq \gamma\}$. We want to find the largest γ such that $I - M\Delta$ is stable for all $\Delta \in \mathbf{\Delta}_\gamma$

The Structured Singular Value, μ

The system

$$\dot{x}(t) = A_0 x(t) + Mp(t), \qquad p(t) = \Delta(t) q(t),$$
$$q(t) = Nx(t) + Qp(t), \qquad \Delta \in \boldsymbol{\Delta}$$

is stable if there exists a $P(\delta) \in \Sigma_s$ such that

$$\dot{V} = x^T P(\delta)(A_0 x + Mp) + (A_0 x + Mp)^T P(\delta) x < \epsilon x^T x$$

for all x, p, δ such that

$$(x, p, \delta) \in \left\{ x, p, \delta \,:\, p = \operatorname{diag}(\delta_i)(Nx + Qp),\, \sum_i \delta_i^2 \le \gamma \right\}$$

Proposition 5 (Lower Bound for μ).

$\mu \ge \gamma$ if there exist polynomial $h \in \mathbb{R}[x, p, \delta]$ and $s_i \in \Sigma_s$ such that

$$x^T P(\delta)(A_0 x + Mp) + (A_0 x + Mp)^T P(\delta) x - \epsilon x^T x$$
$$= -s_0(x, p, \delta) - (\gamma - \sum_i \delta_i^2) s_1(x, p, \delta) - (p - \operatorname{diag}(\delta_i)(Nx + Qp)) h(x, p, \delta)$$

Variations - Polya's Formulation

Recall that Hilbert's 17th was resolved in the affirmative by E. Artin in 1927.

- Any PSD polynomial p is the sum, product and ratio of squared polynomials.
$$p(x) = \frac{g(x)}{h(x)}$$
where $g, h \in \Sigma_s$.

It was later shown by Habricht that if p is strictly positive, then we may assume $h(x) = (\sum_i x_i^2)^d$ for some d. That is,
$$(x_1^2 + \cdots + x_n^2)^d p(x) \in \Sigma_s$$

Question: Given properties of p, may we assume a structure for h?

Yes: Polya was able to show that if $p(x)$ has the structure
$$p(x) = \tilde{p}(x_i^2, \cdots, x_n^2),$$
then we may assume that s is a sum of squared monomials (prima facie SOS).
$$s(x) = \sum_{\alpha \in \mathbb{N}^n} (c_\alpha x^\alpha)^2$$
where $x^\alpha = \prod_i x_i^{\alpha_i}$.

Variations - Polya's Formulation

Consider polynomials on the positive orthant:

$$X := \{x : x_i \geq 0, i = 1, \cdots\}$$

Then: $f(x_1, \cdots, x_n) > 0$ for all $x \in X$ iff $f(x_1^2, \cdots, x_n^2) \geq 0$ for all $x \in \mathbb{R}^n$.
Polya's result: if $f(x_1, \cdots, x_n) > 0$ for all $x \in X$, then

$$\left(\sum_i x_i^2\right)^{d_p} f(x_1^2, \cdots, x_n^2) = \sum_{\alpha \in \mathbb{N}^n} (c_\alpha x^\alpha)^2$$

for some $d_p > 0$.
Now making the substitution $x_i^2 \to y_i$ and $c_\alpha^2 \to b_\alpha$, we have the condition

Theorem 31.

If $f(x_1, \cdots, x_n) > 0$ for all $x \in X$ then there exist $b_\alpha \geq 0$ and $d_p \geq 0$ such that

$$\left(\sum_i y_i\right)^{d_p} f(y_1, \cdots, y_n) = \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha|_1 \leq d + d_p}} b_\alpha y^\alpha$$

where d is the degree of polynomial f.

Variations - Polya's Formulation

Define the Unit Simplex:

$$\Delta := \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0\}$$

Theorem 32 (Polya's Theorem).

Suppose F is a homogeneous polynomial and $F(x) > 0$ for all $x \in \Delta$. Then for a sufficiently large $d \in \mathbb{N}$,

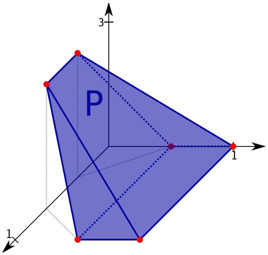
$$(x_1 + x_2 + \cdots + x_n)^d F(x)$$

has all its coefficients strictly positive.

The algorithmic nature was noted by Polya himself:

> *"The theorem gives a systematic process for deciding whether a given form F is strictly positive for positive x. We multiply repeatedly by $\sum x$, and, if the form is positive, we shall sooner or later obtain a form with positive coefficients."* -G. Pólya, 1934

Variations - Polya's Formulation



For example, if we have a finite number of operating points A_i, and want to ensure performance for all combinations of these points.

$$\dot{x}(t) = Ax(t) \quad \text{where} \quad A \in \left\{ \sum_i A_i \mu_i \,:\, \mu_i \geq 0,\ \sum_i \mu_i = 1 \right\}$$

This is equivalent to the existence of a polynomial P such that $P(\mu) > 0$ for all $\mu \in \Delta$ and such that

$$A(\mu)^T P(\mu) + P(\mu)A(\mu) < 0 \quad \text{for all} \quad \mu \in \Delta$$

$$\text{where} \quad A(\mu) = \sum_i A_i \mu_i$$

Variations - Polya's Formulation

A more challenging case is if $A(\alpha)$ is *nonlinear* in some parameters, α.
Simple Example: Angle of attack (α)

$$\dot{\alpha}(t) = -\frac{\rho v^2 c_\alpha(\alpha, M)}{2I}\alpha(t)$$

The time-varying parameters are:

- velocity, v and Mach number, M (M depends on Reynolds #);
- density of air, ρ;
- Also, we sometimes treat α itself as an uncertain parameter.

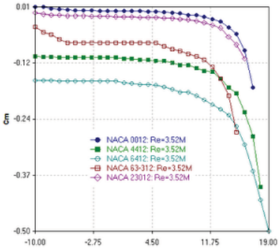


Figure: C_M vs. α and Re #

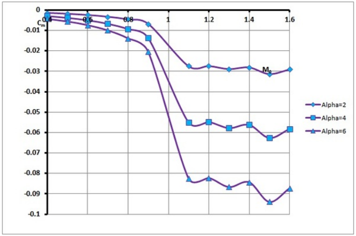


Figure: C_M vs. Mach # and α

ROLMIP: LMIs with parameters lying in the simplex I

- Parameter-dependent LMIs associated to polytopic systems (parameters lying in the simplex) as continuous-time stability ($A(\alpha)'P(\alpha) + P(\alpha)A(\alpha) < 0$), discrete-time stability ($A(\alpha)'P(\alpha)A(\alpha) - P(\alpha) < 0$), $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms computation, controller synthesis, etc., can be put in the general form

$$X(\alpha) = \sum \alpha_1^{k_1} \alpha_2^{k_2} \cdots \alpha_N^{k_N} \left(X_{k_1 \cdots k_N} \right)$$

where $\alpha_1^{k_1} \alpha_2^{k_2} \cdots \alpha_N^{k_N}$ are monomials and $X_{k_1 \cdots k_N}$ are matrix-valued coefficients depending affinely on the decision variables (Lyapunov matrix and possibly some slack variables).

- How to check the positivity of $X(\alpha)$? Easy sufficient test: as $\alpha_1^{k_1} \alpha_2^{k_2} \cdots \alpha_N^{k_N}$ are always non-negative, just impose $X_{k_1 \cdots k_N} > 0$ (linear matrix inequality) for all monomials.

- Problem: How to obtain $X_{k_1 \cdots k_N}$ systematically, for decision variables (Lyapunov matrix and slack variables) of arbitrary degrees, possibly with some Pólya's relaxations?

ROLMIP: LMIs with parameters lying in the simplex II

Solution: With ROLMIP your problems are over, because all trick polynomial manipulations are performed for you and the LMIs are delivered automatically. For instance, consider the problem of continuous-time robust stability analysis of a polytopic system with dynamic matrix given by

$$A(\alpha) = \alpha_1 A_1 + \alpha_2 A_2,\ A_i \in \mathbb{R}^{2\times 2}$$

where A_1 and A_2 are given. Considering a Lyapunov matrix of degree g and d Pólya's relaxations, we have the code:

```
N=2;
n=2;
A=rolmipvar({A1,A2},'A(\alpha)',N,1);
P=rolmipvar(n,n,'P','symmetric',N,g);
LMIs = [polya(A'*P+P*A,d)<=0,  polya(P,d)>=0.000001*eye(n)];
optimize(LMIs,[])
```

- New paper about ROLMIP (version 3.0) to appear in ACM Transactions on Mathematical Software (TOMS). New stuff: multi-simplex uncertainty and the treatment of time-varying parameters (continuous- and discrete-time cases).

Variations - Handelman's Formulation

Polya was not alone in looking for structure on s.

Recall Schmudgen's Positivstellensatz.

Theorem 33 (Schmudgen).

Suppose that $S = \{x : g_i(x) \geq 0\}$ is compact. If $f(x) > 0$ for all $x \in S$, then there exist $s_i, r_{ij}, \cdots \in \Sigma_s$ such that

$$f = s_0 + \sum_i s_i g_i + \sum_{i \neq j} r_{ij} g_i g_j + \sum_{i \neq j \neq k} r_{ijk} g_i g_j g_k + \cdots$$

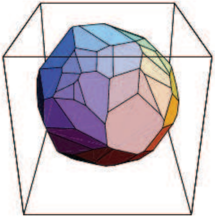
Suppose that S is a **CONVEX** polytope

$$S := \{x \in \mathbb{R}^n \,:\, a_i^T x \leq b_i,\ i = 1, \cdots\}$$

Then we may assume all the s_i are positive scalars.

Variations - Handelman's Formulation

Let $S := \{x \in \mathbb{R}^n \,:\, a_i^T \le b_i\}$.



Theorem 34 (Handelman).

Suppose that $S := \{x \in \mathbb{R}^n \,:\, a_i^T x \le b_i\}$ *is compact and convex with non-empty interior. If* $p(x) > 0$ *for all* $x \in S$*, then there exist* **CONSTANTS** $s_i, r_{ij}, \cdots > 0$ *such that*

$$p = s_0 + \sum_i s_i g_i + \sum_{i \ne j} r_{ij} g_i g_j + \sum_{i \ne j \ne k} r_{ijk} g_i g_j g_k + \cdots$$

Handelman's Formulation (LP Implementation)

Example: Consider the hypercube

$$S := \{(x, y) \, : \, -1 \le x \le 1, \; -1 \le y \le 1\}$$

Now the polytope is defined by 4 inequalities

$$g_1(x,y) = -x + 1; \quad g_2(x,y) = x + 1; \quad g_3(x,y) = -y + 1; \quad g_4(x,y) = y + 1$$

Which yields the following vector of bases

$$\begin{bmatrix} g_1 \\ \vdots \\ g_3 g_4 \end{bmatrix} = \begin{bmatrix} -x + 1 \\ x + 1 \\ -y + 1 \\ y + 1 \\ x^2 - 2x + 1 \\ x^2 + 2x + 1 \\ y^2 - 2y + 1 \\ y^2 + 2y + 1 \\ -x^2 + 1 \\ xy - x - y + 1 \\ -xy - x + y + 1 \\ -xy + x - y + 1 \\ -y^2 + 1 \end{bmatrix}$$

Handelman's Basis (LP Implementation)

First put the function in the linear basis

$$p(x) = -(y^2 + xy + y) + 3 = \begin{bmatrix} 3 & 0 & -1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}$$

Then convert the Handelman basis to the original basis

$$\begin{bmatrix} g_1 \\ \vdots \\ g_3 g_4 \end{bmatrix} = \begin{bmatrix} -x+1 \\ x+1 \\ -y+1 \\ y+1 \\ x^2-2x+1 \\ x^2+2x+1 \\ y^2-2y+1 \\ y^2+2y+1 \\ -x^2+1 \\ xy-x-y+1 \\ -xy-x+y+1 \\ -xy+x-y+1 \\ -y^2+1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & & & & \\ 1 & 1 & & & & \\ 1 & & -1 & & & \\ 1 & & 1 & & & \\ 1 & -2 & & & 1 & \\ 1 & 2 & & & 1 & \\ 1 & & -2 & & & 1 \\ 1 & & 2 & & & 1 \\ 1 & & & & -1 & \\ 1 & -1 & -1 & 1 & & \\ 1 & -1 & 1 & -1 & & \\ 1 & 1 & -1 & -1 & & \\ 1 & & & & & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}$$

Now the positivity constraint becomes $c_i > 0$ and

$$p(x) = \begin{bmatrix} c_1 \\ \vdots \\ c_{13} \end{bmatrix}^T \begin{bmatrix} g_1(x) \\ \vdots \\ g_3(x) g_4(x) \end{bmatrix}.$$

Therefore, substituting the expressions of the previous slide

$$\underbrace{\underbrace{\begin{bmatrix} 3 \\ 0 \\ -1 \\ -1 \\ 0 \\ -1 \end{bmatrix}^T}_{b^T} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}}_{p(x)} = \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_{13} \end{bmatrix}^T}_{x^T} \underbrace{\underbrace{\begin{bmatrix} 1 & -1 & & & & \\ 1 & 1 & & & & \\ 1 & & -1 & & & \\ 1 & & 1 & & & \\ 1 & -2 & & & 1 & \\ 1 & 2 & & & 1 & \\ 1 & & -2 & & & 1 \\ 1 & & 2 & & & 1 \\ 1 & & & & -1 & \\ 1 & -1 & -1 & 1 & & \\ 1 & -1 & 1 & -1 & & \\ 1 & 1 & -1 & -1 & & \\ 1 & & & & & -1 \end{bmatrix}}_{A} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}}_{g_i \cdots \prod g_i}$$

Handelman's Basis (LP Implementation)

Finally, we have that positivity of p can be expressed as the search for $c_i > 0$ such that

$$\underbrace{\begin{bmatrix} 1 & -1 & & & & \\ 1 & 1 & & & & \\ 1 & & -1 & & & \\ 1 & & 1 & & & \\ 1 & -2 & & & 1 & \\ 1 & 2 & & & 1 & \\ 1 & & -2 & & & 1 \\ 1 & & 2 & & & 1 \\ 1 & & & & -1 & \\ 1 & -1 & -1 & 1 & & \\ 1 & -1 & 1 & -1 & & \\ 1 & 1 & -1 & -1 & & \\ 1 & & & & & -1 \end{bmatrix}^T}_{A^T} \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_{13} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 3 \\ 0 \\ -1 \\ -1 \\ 0 \\ -1 \end{bmatrix}}_{b}$$

Which is of the form $A^T x = b$ in variables $x > 0$.

Recall: Optimization over the positive orthant is called *Linear Programming*.

- b is determined by the coefficients of the polynomial, p
- b may itself be a variable if we are searching over positive polynomials.

Handelman's Basis (LP Implementation)

For the polynomial

$$p(x) = -(y^2 + xy + y) + 3 = \begin{bmatrix} 3 & 0 & -1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}$$

The Linear Program is feasible with

$$x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

This corresponds to the form

$$\begin{aligned} p(x) &= g_3(x)g_4(x) + g_2(x)g_4(x) + g_1(x) \\ &= (-y^2 + 1) + (-xy + x - y + 1) + (-x + 1) \\ &= -y^2 - xy - y + 3 \end{aligned}$$

Problems with Interior ZEROS!!!

Now consider the polynomial

$$p(x) = x^2 + y^2 = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 1\end{bmatrix}\begin{bmatrix}1 & x & y & xy & x^2 & y^2\end{bmatrix}^T$$

Clearly, $p(x,y) \geq 0$ for all $(x,y) \in S$. However the LP is NOT feasible. Consider the point $(x,y) = (0,0)$. Then $p(0,0) = 0$ and

$$p(0) = \begin{bmatrix}0\\0\\0\\0\\1\\1\end{bmatrix}^T \begin{bmatrix}1\\x\\y\\xy\\x^2\\y^2\end{bmatrix}_{(x,y)=0} = \begin{bmatrix}0\\0\\0\\0\\1\\1\end{bmatrix}^T \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}c_1\\ \vdots \\ c_{13}\end{bmatrix}^T \begin{bmatrix}-x+1\\x+1\\-y+1\\y+1\\x^2-2x+1\\x^2+2x+1\\y^2-2y+1\\y^2+2y+1\\-x^2+1\\xy-x-y+1\\-xy-x+y+1\\-xy+x-y+1\\-y^2+1\end{bmatrix}_{(x,y)=(0,0)} = \begin{bmatrix}c_1\\ \vdots \\ c_{13}\end{bmatrix}^T \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}$$

Which implies $\sum_i c_i = 0$. Since the $c_i \geq 0$, this means $c = 0$ - NOT FEASIBLE!

Variations - Handelman's Formulation

Conclusion: For many representations, the strict positivity is necessary.

- Polya's representation precludes interior-point zeros.
- Handelman's representation precludes interior-point zeros.
- Bernstein's representation precludes interior-point zeros.

In each of these cases, we may have zeros at vertices of the set.

- This makes searching for a Lyapunov function impossible.
  - Must be positive on a neighborhood of the $x = 0$ with $V(0) = 0$.

One Solution: Partition the space so that the zero point is a vertex of each set.