LMI Methods in Optimal and Robust Control

Matthew M. Peet
Arizona State University
Thanks to S. Lall and P. Parrilo for guidance and supporting material

Lecture 17: The PositivStellenSatz and an LMI for Local Stability

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.

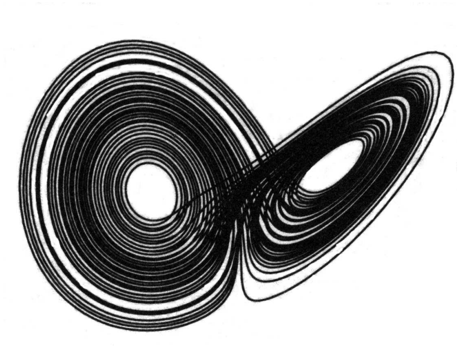


Figure: The Lorentz Attractor

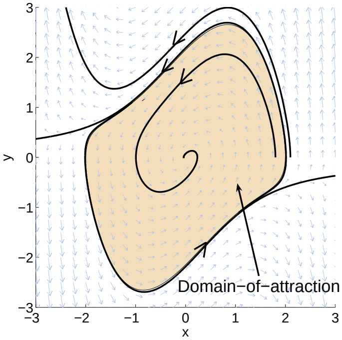


Figure: The van der Pol oscillator in reverse

Local Positivity

A more interesting question is the question of local positivity.
Question: Is $y(x) \geq 0$ for $x \in X$, where $X \subset \mathbb{R}^n$.

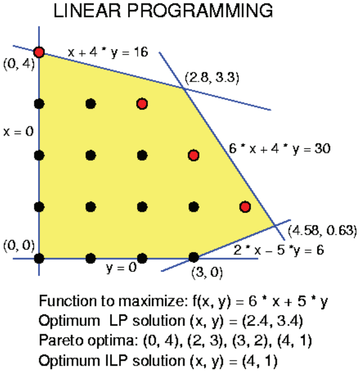
Examples:

- 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**.

Positivity on Which Sets?

Semialgebraic Sets (Defined by *Polynomial* Inequalities)

How are these sets represented???

Definition 1.

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

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

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...

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

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

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

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

Recall 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* $x \in \mathbb{R}^n$

SOS Positivity on a Subset

Recall the S-Procedure

Corollary 2 (S-Procedure).

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

This works because

- $\tau \geq 0$ and $z^T G z \geq 0$ for all $z \in S$
- Hence $\tau z^T G z \geq 0$ for all $z \in S$

If $F \geq \tau G$, then

$$\begin{aligned} z^T F z &\geq \tau z^T G z \qquad \text{for all } \; z \in \mathbb{R}^n \\ &\geq 0 \qquad \text{for all } \; z \in S \end{aligned}$$

Now Consider *Polynomials*

Proposition 2.

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

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

Summary of SOS Positivity on a set

The Main Idea

Proposition 3.

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

$$f(x) \geq 0 \qquad \text{for all} \quad x \in S := \{x \;:\; g_i(x) \geq 0,\; h_i(x) = 0\}$$

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!

How to Represent a Set???

A Problem of Representation and Inference

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.

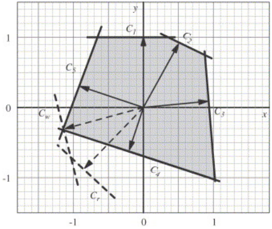
$$\{x \in \mathbb{R} : (x - a)(b - x) \geq 0,\ x - a \geq 0\}$$
$$\{x \in \mathbb{R} : (x^2 + 1)(x - a)(b - x) \geq 0\}$$
$$\{x \in \mathbb{R} : (x-a)(b-x) \geq 0,\ (x^2+1)(x-a)(b-x) \geq 0,\ (x - a)(b - x) \geq 0\}$$

There are infinite ways to represent the same set

- Some Work well and others Don't!

A Problem of Representation and Inference

Computer-Based Logic and Reasoning



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 3.

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 4.

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 5.

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 6.

$f(x) \geq 0$ 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 7.

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 8.

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

Corollary 9.

$h \in$ 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.
- The equality constraint acts on the coefficients of f, s_i, r_{ij}.

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: To show that $h(x) = 5x - 9x^2 + 5x^3 - x^4$ is PSD on the interval $[0,1] = \{x \in \mathbb{R}^n : x(1-x) \geq 0\}$, we use $f_1(x) = x(1-x)$. This yields the constraint

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

We find $s_0(x) = 0$, $s_1(x) = (2-x)^2 + 1$ so that

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

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 10 (Stengle's Positivstellensatz).

$S = \emptyset$ if and only if $-1 \in \textit{cone}(\{f_i\})$. 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 11 (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 12 (Stengle's Positivstellensatz).

$f(x) \geq 0$ for all $x \in \{x : g_i(x) \geq 0\}$ if and only if there exist $s_i, q_{ij}, r_{ij}, \cdots \in \Sigma_s$ 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 13 (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* $s_i, r_{ij}, \cdots \in \Sigma_s$ *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 14.

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 level set $\{x : f_i(x) \geq 0\}$ is compact.

P-Compact is not hard to satisfy.

Corollary 15.

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 16 (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.

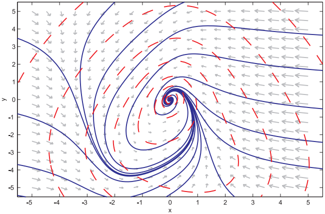
- We are back to the original condition
- A Good representation of the set is P-compact

Return to Lyapunov Stability

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 17.

Suppose there exists a polynomial v, a constant $\epsilon > 0$, and sum-of-squares polynomials s_0, s_i, t_0, t_i such that

$$v(x) - \sum_i s_i(x) p_i(s) - s_0(s) - \epsilon\, x^T x = 0$$

$$-\nabla v(x)^T f(x) - \sum_i t_i(x) p_i(s) - t_0(x) - \epsilon\, x^T x = 0$$

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.

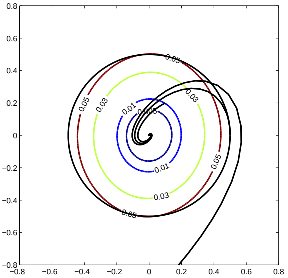
Local Stability Analysis

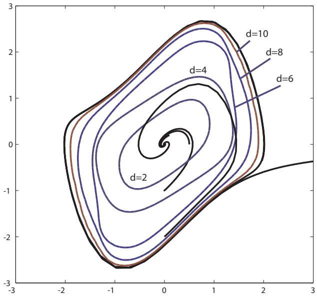
Van-der-Pol Oscillator

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

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





Local Stability Analysis

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)
```

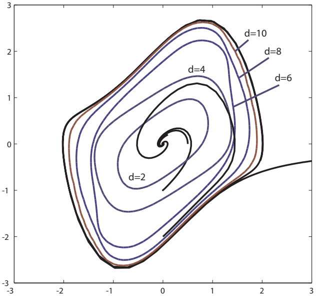
This finds a Lyapunov function which is decreasing on the ball of radius $\sqrt{2.8}$

- 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;
```
> $g = r - (x^2 + y^2)$;
```
> 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 $u_1(t) = k_1(x(t))$, $u_2(t) = k_2(x(t))$