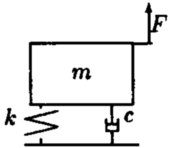
LMI Methods in Optimal and Robust Control

Matthew M. Peet
Arizona State University

Lecture 18: SOS for Robust Stability and Control

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 1.

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 2.

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 MAE 598 download

- Also on Blackboard
- I may add features associated with later Lectures in the future.

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:

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

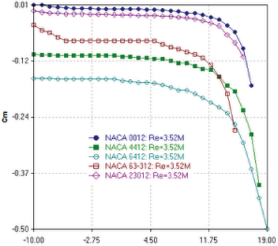
$$y(t) = C(\delta(t))x(t) + D(\delta(t))u(t) \qquad \dot{\delta}(t) \in \Delta_2$$

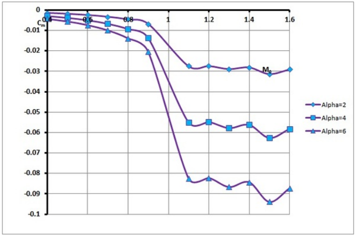
Simple Example: Angle of attack (α)

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

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 3.

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 4.

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*

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

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

Trace$W(\delta) < \gamma^2$

for all $\delta \in \Delta$.

The KYP Lemma with Time-Varying Uncertainty

Lemma 5.

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

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

Proof.

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

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

□

H_∞-optimal robust control with Time-Varying Uncertainty

However, Controller Synthesis is a Problem!

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

Lemma 6.

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)

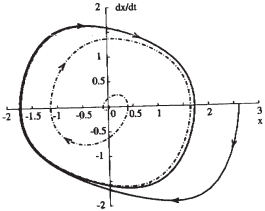
- Optimize P, then optimize K.
- rinse and repeat.

Robust Local Stability

Search for a Parameter-Dependent Lyapunov Function

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
```
> $\mathtt{zmin} = .8; \mathtt{zmax} = 1.2; \mathtt{amin} = 1.8; \mathtt{amax} = 2.2; \mathtt{g1} = \mathtt{r} - (\mathtt{x1}^2 + \mathtt{x2}^2);$

> $\mathtt{r} = .3; \mathtt{g2} = (\mathtt{amax} - \mathtt{a})(\mathtt{a} - \mathtt{amin}); \mathtt{g3} = (\mathtt{zmax} - \mathtt{z})(\mathtt{z} - \mathtt{zmin});$

> $\mathtt{f} = [2 * \mathtt{z} * (1 - \mathtt{a} * \mathtt{x2}^2) * \mathtt{x2} - \mathtt{x1}; \mathtt{x1}];$

```
> 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);
```
> $\mathtt{V} = \mathtt{V0} + .001 * (\mathtt{x1}^2 + \mathtt{x2}^2) + \mathtt{g1} * \mathtt{r1} + \mathtt{g2} * \mathtt{r2} + \mathtt{g3} * \mathtt{r3};$

```
> 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

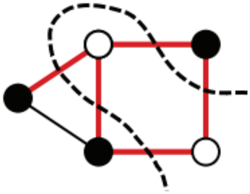


Figure: Division of a set of nodes to maximize the weighted cost of separation

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

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

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

$$\begin{aligned}
&\min_{p_i \in \mathbb{R}[x],\, s_0 \in \Sigma_s} \gamma : \\
&\gamma - \frac{1}{2} \sum_{i,j} w_{i,j}(1 - x_i x_j) + \sum_i p_i(x)(x_i^2 - 1) = s_0(x)
\end{aligned}$$

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

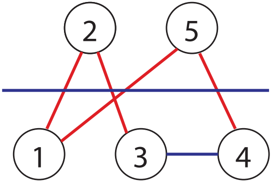


Figure: A Proposed Cut

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.

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

- δ_i represent unknown parameters.

Definition 7.

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 $\mathbf{\Delta}_\gamma = \{\text{diag}(\delta_i), : \sum_i \delta_i^2 \leq \gamma\}$. 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

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

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 1 (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

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