Получение результатов в виде линейных матричных неравенств для робастности

Дмитрий Жанович Коновалов Dmitry Peaucelle
Димитрий Посель Dimitri Peaucelle

Традиционная Школа "Управление, Информация и Оптимизация"

Переславль-Залесский Июнь 2010
Obtaining LMI results for robust control problems

Dimitri Peaucelle
LAAS-CNRS, Toulouse, FRANCE
peaucelle@laas.fr  http://homepages.laas.fr/peaucell

Traditional School
"Control, Information and Optimization"

Pereslavl’-Zalesskii  June 2010
Robust multi-objective problems

Linear Matrix Inequalities & Optimization tools

Manipulating inequalities to obtain LMI results

Some LMI results

Solving robust multi-objective problems in RoMulOC

```matlab
g = ctrb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);
```
Robust multi-objective problems

- Standard robust analysis problem:
  \[ w_{\Delta} \rightarrow \Delta \rightarrow z_{\Delta} \rightarrow \Sigma \]
  prove stability of \((\sum w_{\Delta} \star z_{\Delta} \Delta)\) for all \(\Delta \in \Delta\)

- Standard robust design problem:
  \[ w_{\Delta} \rightarrow \Delta \rightarrow z_{\Delta} \rightarrow \Sigma \rightarrow K \rightarrow u \rightarrow y \]
  Find \(K\) that guarantees stability of \(((\sum w_{\Delta} \star z_{\Delta} \Delta) \star u, y \star K)\) for all \(\Delta \in \Delta\)

- \(\Delta\) contains unknown parameters, scheduling parameters, approximations of non-linearities, delays...

- \(\Sigma\) is a linear model: crude but simple representation of the system
Robust multi-objective problems

- Generalizes to robust performance problems

- Guarantee an input/output property for all $\Delta \in \Delta$

- Find a controller that guarantees input/output properties for all $\Delta \in \Delta$

- Same holds for polytopic-type models
Robust multi-objective problems

- Robust multi-objective problem
- Find a controller $K$ that guarantees simultaneously several robust specifications $\Pi_1, \Pi_2, \ldots$ each of which being defined for a possibly different models $(\Sigma_1 \ast \Delta_1), (\Sigma_2 \ast \Delta_2), \ldots$

**Example: Robust $H_2/H_\infty$ problem**

$$
\begin{align*}
\dot{x} &= A(\Delta)x + B_u(\Delta)u \\
\dot{z} &= u \\
y &= C_y(\Delta)x + w \\
u &= Ky
\end{align*}
$$

$$
\begin{align*}
\dot{x_f} &= A_f x_f + B_f w \\
\dot{x} &= A(\Delta)x + B_w(\Delta)x_f + B_u(\Delta)u \\
\dot{z} &= C_z(\Delta)x \\
y &= C_y(\Delta)x \\
u &= Ky
\end{align*}
$$

$$
\leq \gamma_{\infty}
$$
Robust multi-objective problems

Naturally defined as existence (feasibility) problem over several constraints.

While a nominal performance $\| \Sigma \star K \| = \gamma$ may be defined by an equality.

Robust performance $\| (\Sigma \star \Delta) \star K \| \leq \gamma$, $\forall \Delta \in \Delta$ can only be an inequality.


LMI: Upper-bounds, Convex optimization, polynomial-time.
1 Robust multi-objective problems

2 Linear Matrix Inequalities & Optimization tools

\[ A^T P + PA < 0 \iff Ax = b, \ x \in K \]

3 Manipulating inequalities to obtain LMI results

\[ \begin{cases} 
A > BC^{-1}B^T \\
C > 0 
\end{cases} \iff \begin{bmatrix} 
A & B \\
B^T & C 
\end{bmatrix} > 0 \]

\[ A^T(\zeta)P + PA(\zeta) < 0 \ \forall \zeta \in \Delta \]

\[ \Downarrow \]

\[ A^{[v]^T}P + PA^{[v]} < 0 \ \forall v \in \{1 \ldots \bar{v}\} \]

4 Some LMI results

5 Solving robust multi-objective problems in RoMulOC

» quiz = ctrpb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);
Convex cones

A set $\mathcal{K}$ is a cone if for every $x \in \mathcal{K}$ and $\lambda \geq 0$ we have $\lambda x \in \mathcal{K}$.

A set is a convex cone if it is convex and a cone.
Convex cones

- Convex cone of positive reals: \( x \in \mathbb{R}_+ \)
- Second order (Lorentz) cone: \( \mathcal{K}_{soc}^n = \left\{ x = \begin{pmatrix} x_1 & \ldots & x_n \end{pmatrix}, x_1^2 + \ldots + x_{n-1}^2 \leq x_n^2 \right\} \)
Convex cones

Convex cone of positive reals: \( x \in \mathbb{R}_+ \)

Second order (Lorentz) cone:
\[
K^{n}_{soc} = \left\{ x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}, \ x_1^2 + \cdots + x_{n-1}^2 \leq x_n^2 \right\}
\]

Positive semi-definite matrices:
\[
K^{n}_{psd} = \left\{ x = \begin{pmatrix} x_1 & \cdots & x_n^2 \end{pmatrix}, \ \text{mat}(x) = \text{mat}(x)^T \geq 0 \right\}
\]

\( K^2_{psd} \):
\[
\begin{bmatrix}
  x_1 & x_2 \\
  x_2 & x_3
\end{bmatrix} \geq 0
\]
Convex cones

- Convex cone of positive reals: $x \in \mathbb{R}_+$

- Second order (Lorentz) cone: $\mathcal{K}^n_{soc} = \left\{ x = \begin{pmatrix} x_1 & \ldots & x_n \end{pmatrix}, \ x_1^2 + \ldots + x_{n-1}^2 \leq x_n^2 \right\}$

- Positive semi-definite matrices: $\mathcal{K}^n_{psd} = \left\{ x = \begin{pmatrix} x_1 & \cdots & x_{n^2} \end{pmatrix}, \ \text{mat}(x) \geq 0 \right\}$

- Unions of such: $\mathcal{K} = \mathbb{R}_+ \times \cdots \times \mathcal{K}^n_{soc} \times \cdots \times \mathcal{K}^n_{psd} \times \cdots$
Optimization over convex cones

\[ p^* = \min cx : Ax = b , \ x \in \mathcal{K} \]

*Linear programming:* \( \mathcal{K} = \mathbb{R}_+ \times \cdots \mathbb{R}_+ \).

*Semi-definite programming:* \( \mathcal{K} = \mathcal{K}_{psd}^{n_1} \times \cdots \mathcal{K}_{psd}^{n_q} \)

Dual problem

\[ d^* = \max b^T y : A^T y - c^T = z , \ z \in \mathcal{K} \]

Primal feasible \(\rightarrow\) Dual infeasible

Dual feasible \(\rightarrow\) Primal infeasible

If primal and dual strictly feasible \( p^* = d^* \)

Polynomial-time algorithms \( (\mathcal{O}(n^{6.5} \log(1/\epsilon))) \)
Optimization over convex cones

\[ p^* = \min c x : A x = b , \ x \in \mathcal{K} \]

Dual problem

\[ d^* = \max b^T y : A^T y - c^T = z , \ z \in \mathcal{K} \]

Possibility to perform convex optimization, primal/dual, interior-point methods, etc.

- Primal-dual path-following predictor-corrector algorithms:
  - SeDuMi (Sturm), SDPT3 (Toh, Tütüncü, Todd), CSDP (Borchers), SDPA (Kojima et al.)
- Primal-dual potential reduction: MAXDET (Wu, Vandenberghe, Boyd)
- Dual-scaling path-following algorithms: DSDP (Benson, Ye, Zhang)
- Barrier method and augmented Lagrangian: PENSDP (Kocvara, Stingl)
- Cutting plane algorithms ...
Semi-Definite Programming and LMIs

**SDP formulation**
\[
\begin{align*}
p^* &= \min cx : \quad Ax = b \quad , \quad x \in \mathcal{K} \\
d^* &= \max b^T y : \quad A^T y - c^T = z \quad , \quad z \in \mathcal{K}
\end{align*}
\]

**LMI formalism**
\[
\begin{align*}
d^* &= \min \sum g_i y_i : \quad F_0 + \sum F_i y_i \succeq 0 \\
p^* &= \max \text{Tr}(F_0 X) : \quad \text{Tr}(F_i X) + g_i = 0 \quad , \quad X \succeq 0
\end{align*}
\]

In control problems: variables are matrices

The $H_\infty$ norm computation example for $G(s) \sim (A, B, C, D)$:

\[
\|G(s)\|_\infty^2 = \min \gamma : \quad P > 0 ,
\]

\[
\begin{bmatrix}
A^T P + PA + C_z^T C_z & B_w P + C_z^T D_z w \\
P B_w^T + D_z w C_z & -\gamma 1 + D_z^T w D_z w
\end{bmatrix} < 0
\]


Need for a nice parser
Parsers: LMIlab, tkLmitool, sdpsol, SeDuMiInterface...

YALMIP

Convert LMIs to SDP solver format (all available solvers!)

Simple to use

```matlab
>> P = sdpvar(3,3,'symmetric');
>> lmiprob = lmi(A'*P+P*A<0) + lmi(P>0);
>> solvesdp(lmiprob);
```

Works in Matlab - free!

http://users.isy.liu.se/johanl/yalmip

Extends to other non-SDP optimization problems (BMI...)

SDP dedicated version in Scilab [S. Solovyev]

http://www.laas.fr/OLOCEP/SciYalmip
SDP-LMI issues and prospectives

- Any SDP representable problem is "solved" (numerical problems due to size and structure)
  - Find "SDP-ables" problems
    - (linear systems, performances, robustness, LPV, saturations, delays, singular systems...)
  - Equivalent SDP formulations \( \Rightarrow \) distinguish which are numerically efficient
  - New SDP solvers: faster, precise, robust (need for benchmark examples)
SDP-LMI issues and prospectives

- Any SDP representable problem is "solved" (numerical problems due to size and structure)
- Find "SDP-ables" problems
  (linear systems, performances, robustness, LPV, saturations, delays, singular systems...)
- Equivalent SDP formulations ⟹ distinguish which are numerically efficient
- New SDP solvers: faster, precise, robust (need for benchmark examples)

- Any "SDP-able" problem has a dual interpretation
- New theoretical results (worst case)
- New proofs (Lyapunov functions = Lagrange multipliers; related to SOS)
- SDP formulas numerically stable (KYP-lemma)
SDP-LMI issues and prospectives

- Any SDP representable problem is "solved" (numerical problems due to size and structure)
  - Find "SDP-ables" problems
  - (linear systems, performances, robustness, LPV, saturations, delays, singular systems...)
  - Equivalent SDP formulations \(\Rightarrow\) distinguish which are numerically efficient
  - New SDP solvers: faster, precise, robust (need for benchmark examples)

- Any "SDP-able" problem has a dual interpretation
  - New theoretical results (worst case)
  - New proofs (Lyapunov functions = Lagrange multipliers; related to SOS)
  - SDP formulas numerically stable (KYP-lemma)

- Non "SDP-able": Robustness & Multi-objective & Relaxation of NP-hard problems
  - Optimistic / Pessimistic (conservative) results
  - Reduce the gap (upper/lower bounds) while handling numerical complexity growth.
Linear Matrix Inequalities & Optimization tools

- SDP-LMI issues and prospectives
  - Any SDP representable problem is "solved" (numerical problems due to size and structure)
  - Find "SDP-ables" problems (linear systems, performances, robustness, LPV, saturations, delays, singular systems...)
  - Equivalent SDP formulations ⇒ distinguish which are numerically efficient
  - New SDP solvers: faster, precise, robust (need for benchmark examples)
  - Any "SDP-able" problem has a dual interpretation
  - New theoretical results (worst case)
  - New proofs (Lyapunov functions = Lagrange multipliers; related to SOS)
  - SDP formulas numerically stable (KYP-lemma)

- Non "SDP-able": Robustesse & Multi-objective & Relaxation of NP-hard problems
  - Optimistic / Pessimistic (conservative) results
  - Reduce the gap (upper/lower bounds) while handling numerical complexity growth.
  - Develop software for "industrial" application / adapted to the application field

⇒ RoMuLOC toolbox
Robust multi-objective problems

Linear Matrix Inequalities & Optimization tools

Manipulating inequalities to obtain LMI results

Some LMI results

Solving robust multi-objective problems in RoMulOC

```
> quiz= ctrpb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);
```
Manipulating inequalities to obtain LMI results

**Congruence**

- **A > 0** \(\Rightarrow\) for any non zero vector \(x\):
  \[ x^T A x > 0. \]

- **A > 0** \(\Rightarrow\) for any full column rank matrix \(B\):
  \[ B^T A B > 0. \]

- **A > 0** \(\Rightarrow\) for any matrix \(B\):
  \[ B^T A B \geq 0. \]

- **A > 0** \(\iff\) exists a square non-singular matrix \(B\):
  \[ B^T A B > 0. \]

**Most LMI results are formulated as (sufficiency)**

*If \(\exists P \ldots : \mathcal{L}(P \ldots) > 0\) then the system \(\dot{x} = f(x, w \ldots)\) is such that...*

To prove these results: perform congruence with vectors \(x, w \ldots\)

**Example (Lyapunov):**

*If \(\exists P : P > 0\), \(A^T P + PA < 0\) then the system \(\dot{x} = Ax\) is stable.*

Proof: \(V(x) = x^T P x > 0\), \(\dot{V}(x) = x^T (A^T P + PA) x = 2 \dot{x}^T P x < 0\) for all \(x \neq 0\).
Manipulating inequalities to obtain LMI results

Examples of nominal performance analysis: \((P > 0)\)

- **Stability (discrete-time)**
  \[ A^T P A - P < 0 \]

- **Regional pole placement**
  \[
  \begin{bmatrix}
  1 & A^* \\
  A^* & r_{11} P & r_{12} P \\
  r_{12}^* P & r_{22} P \\
  \end{bmatrix}
  \begin{bmatrix}
  1 \\
  A \\
  \end{bmatrix} < 0
  \]

- **\(H_\infty\) norm**
  \[
  \begin{bmatrix}
  A^T P + PA + C^T_z C_z & PB_w + C^T_z D_{zw} \\
  B^T_{wz} P + D^T_{zw} C_z & -\gamma^2 1 + D^T_{zw} D_{zw} \\
  \end{bmatrix} < 0
  \]

- **\(H_2\) norm**
  \[ A^T P + PA + C^T_z C_z < 0 \]
  \[ \text{trace}(B^T_w P B_w) < \gamma^2 \]

- **Impulse-to-peak**
  \[ A^T P + PA < 0 \]
  \[ B^T_{wz} P B_w < \gamma^2 1 \]
  \[ C^T_z C_z < P \]
  \[ D^T_{zw} D_{zw} < \gamma^2 1 \]
Tools to ‘build’ LMI results

Schur complement

\[
\begin{cases}
A > BC^{-1}B^T \\
C > 0
\end{cases}
\Leftrightarrow
\begin{bmatrix}
A & B \\
B^T & C
\end{bmatrix} > 0
\]

Example:

\[
\begin{cases}
(AX + BS)X^{-1}(XA^T + STB^T) - X < 0 \\
X > 0
\end{cases}
\Leftrightarrow
\begin{bmatrix}
-X & AX + BS \\
XA^T + STB^T & -X
\end{bmatrix} < 0
\]
Manipulating inequalities to obtain LMI results

- Tools to ‘build’ LMI results
  - Finsler lemma - Elimination lemma - Creation lemma

\[ x^T A x < 0 \quad \forall x : B x = 0 \iff \exists \tau \in \mathbb{R} : A < \tau B^T B \]
\[ \iff \exists X = X^T : A < B^T X B \]
\[ \iff \exists G : A < B^T G^T + G B \]
\[ \iff B^\perp^T A B^\perp < 0 \]

△ where \( B^\perp \) columns generate the null space of \( B \):

\[ B \in \mathbb{R}^{p \times m}, \ \text{rank}(B) = r < m, \ BB^\perp = 0, \ B^\perp \in \mathbb{R}^{m \times (m-r)}, \ B^\perp^T B^\perp > 0 \]

△ \( G \) is a ‘Slack variable’ (Lagrange multiplier)
Manipulating inequalities to obtain LMI results

- Tools to ‘build’ LMI results
  - Finsler lemma - Elimination lemma - Creation lemma

\[ x^T Ax < 0 \quad \forall x : Bx = 0 \iff \exists \tau \in \mathbb{R} : A < \tau B^T B \]
\[ \iff \exists X = X^T : A < B^T XB \]
\[ \iff \exists G : A < B^T G^T + GB \]
\[ \iff B_{\perp}^T A B_{\perp} < 0 \]

Example:

\[ \dot{V}(x) = \begin{pmatrix} \dot{x} \\ x \end{pmatrix}^T \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} < 0 , \quad \forall \begin{bmatrix} 1 & -A \end{bmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} = 0 \]
\[ \iff \exists G : \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} + \begin{bmatrix} -1 \\ A^T \end{bmatrix} G^T + G \begin{bmatrix} -1 & A \end{bmatrix} < 0 \]
Manipulating inequalities to obtain LMI results

Tools to ‘build’ LMI results

- Finsler lemma
- Elimination lemma
- Creation lemma

\[
\begin{align*}
C^\top A C^\top &< 0 \\
B^\top A B^\top &< 0
\end{align*}
\implies \exists H : A < B^T H^T C + C^T H B
\]

Example

\[
\begin{align*}
-P &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} < 0 \\
A^T P A - P &= \begin{bmatrix} A & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} < 0
\end{align*}
\]

\[
\implies \exists H : \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} < \begin{bmatrix} -1 \\ A^T \end{bmatrix} H^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} H \begin{bmatrix} -1 & A \end{bmatrix}
\]

Manipulating inequalities to obtain LMI results

- Tools to ‘build’ LMI results
- S-procedure [Yakubovich]

\[ x^T M x < 0 \quad \forall x : x^T N x \leq 0 \iff \exists \tau > 0 : M < \tau N \]

Example:

\[
\begin{bmatrix}
1 \\
\Delta
\end{bmatrix}^T M \begin{bmatrix}
1 \\
\Delta
\end{bmatrix} < 0, \quad \forall \Delta^T \Delta \leq 1
\]

\[\iff \begin{pmatrix}
z_{\Delta} \\
w_{\Delta}
\end{pmatrix}^T M \begin{pmatrix}
z_{\Delta} \\
w_{\Delta}
\end{pmatrix} < 0, \quad \forall \begin{pmatrix}
z_{\Delta} \\
w_{\Delta}
\end{pmatrix}^T \begin{bmatrix}
-1 & 0 \\
0 & 1
\end{bmatrix} \begin{pmatrix}
z_{\Delta} \\
w_{\Delta}
\end{pmatrix} \leq 0
\]

\[\iff \exists \tau > 0 : M < \tau \begin{bmatrix}
-1 & 0 \\
0 & 1
\end{bmatrix}\]
Manipulating inequalities to obtain LMI results

Tools to ‘build’ LMI results

- S-procedure [Yakubovich]

\[ x^T M x < 0 \quad \forall x : x^T N x \leq 0 \iff \exists \tau > 0 : M < \tau N \]

Example:

\[
\begin{bmatrix}
1 \\
\Delta(1 - D\Delta)^{-1}C
\end{bmatrix}^T M \begin{bmatrix}
1 \\
\Delta(1 - D\Delta)^{-1}C
\end{bmatrix} < 0, \quad \forall \Delta^T \Delta \leq 1
\]

\[
\Leftrightarrow \begin{pmatrix} x \\ w_\Delta \end{pmatrix}^T M \begin{pmatrix} x \\ w_\Delta \end{pmatrix} < 0, \quad \forall \begin{pmatrix} z_\Delta \\ w_\Delta \end{pmatrix} \begin{bmatrix}
-1 & 0 \\
0 & 1
\end{bmatrix} \begin{pmatrix} z_\Delta \\ w_\Delta \end{pmatrix} \leq 0
\]

\[
z_\Delta = Cx + Dw_\Delta
\]

\[
\Leftrightarrow \exists \tau > 0 : M < \tau \begin{bmatrix}
C & D \\
0 & 1
\end{bmatrix}^T \begin{bmatrix}
-1 & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
C & D \\
0 & 1
\end{bmatrix}
\]

Special case:

\[ X + C^T \Delta^T B^T + B\Delta C < 0, \quad \forall \Delta^T \Delta \leq 1 \iff \exists \tau > 0 : X + \tau C^T C + \tau^{-1} B B^T < 0 \]
Manipulating inequalities to obtain LMI results

Tools to ‘build’ LMI results

\[ \begin{bmatrix} 1 \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix}^T \begin{bmatrix} M \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix} < 0, \quad \forall \Delta = \delta I : \delta \in C, \quad |\delta| \leq 1 \]

\[ \Leftrightarrow \exists Q > 0 : M < \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} -Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} 1 \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix}^T \begin{bmatrix} M \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix} < 0, \quad \forall \Delta = \delta I : \delta \in R, \quad |\delta| \leq 1 \]

\[ \Leftrightarrow \exists Q > 0, \quad T = -T^T : M < \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} -Q & T \\ T^T & Q \end{bmatrix} \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix} \]
Manipulating inequalities to obtain LMI results

- Tools to ‘build’ LMI results
  - Kalman-Yakubovich-Popov KYP lemma

\[ \begin{bmatrix} 1 \\ (j\omega 1 - A)^{-1}B \end{bmatrix}^* M \begin{bmatrix} 1 \\ (j\omega 1 - A)^{-1}B \end{bmatrix} < 0, \quad \forall \omega \in \mathbb{R} \]

\[ \Leftrightarrow \exists Q : M < \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0 & Q \\ Q & 0 \end{bmatrix} \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix} \]

- And so on... Full-Block S-procedure [Scherer], Quadratic Separation [Iwasaki]

\[ \begin{bmatrix} 1 \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix}^* M \begin{bmatrix} 1 \\ \Delta(1 - D\Delta)^{-1}C \end{bmatrix} < 0, \quad \forall \Delta \in \Delta \]

\[ \Leftrightarrow M < \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}^T \Theta \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ \Delta \end{bmatrix}^* \Theta \begin{bmatrix} 1 \\ \Delta \end{bmatrix} < 0 \quad \forall \Delta \in \Delta \]

\[ \Delta \text{ Difficulty: build the ‘separator’ } \Theta, \text{ losslessly...} \]
Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- S-procedure [Yakubovich]

\[ x^T M x < 0 \quad \forall x : x^T N x \leq 0 \quad \iff \quad \exists \tau > 0 : M < \tau N \]

\[ x^T M x < 0 \quad \forall x : \begin{cases} x^T N_1 x \leq 0 \\ \vdots \\ x^T N_p x \leq 0 \end{cases} \quad \iff \quad \exists \tau_1 > 0, \ldots, \tau_p > 0 : M < \tau_1 N_1 + \cdots + \tau_p N_p \]

⚠️ Not lossless except in few special cases: \( \Delta \) composed of

- \( m_r \) scalar real repeated,
- \( m_c \) scalar complex repeated,
- \( m_F \) full complex non-repeated blocks

[Meinsma et al.] DG-scaling lossless if \( 2(m_r + m_c) + m_F \leq 3 \)!
Outline

1. Robust multi-objective problems

2. Linear Matrix Inequalities & Optimization tools

3. Manipulating inequalities to obtain LMI results

4. Some LMI results

5. Solving robust multi-objective problems in RoMulOC

» quiz = ctrpb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);

\[ A^T P + P A < 0 \iff A x = b , \ x \in \mathcal{K} \]

\[ \begin{cases} 
A > B C^{-1} B^T \\
C > 0 
\end{cases} \iff \begin{bmatrix} A & B \\
B^T & C \end{bmatrix} > 0 \]

\[ A^T(\zeta) P + P A(\zeta) < 0 \ \forall \zeta \in \Delta \]

\[ \uparrow \]

\[ A^{[v]} P + P A^{[v]} < 0 \ \forall v \in \{1 \ldots \bar{v}\} \]
Examples of nominal performance analysis: \((P > 0)\)

- **Stability (discrete-time)** \(A^T PA - P < 0\)

- **Regional pole placement**

  \[
  \begin{bmatrix}
  1 & A^* \\
  r_1^* & r_2 \\
  \end{bmatrix}
  \begin{bmatrix}
  r_{11}P & r_{12}P \\
  r_{12}^*P & r_{22}P \\
  \end{bmatrix}
  \begin{bmatrix}
  1 \\
  A \\
  \end{bmatrix} < 0
  \]

- **\(H_\infty\) norm**

  \[
  \begin{bmatrix}
  A^T P + PA + C_z^T C_z & PB_w + C_z^T D_{zw} \\
  B_{zw}^T P + D_{zw}^T C_z & -\gamma^2 1 + D_{zw}^T D_{zw} \\
  \end{bmatrix} < 0
  \]

- **\(H_2\) norm**

  \[
  A^T P + PA + C_z^T C_z < 0
  \]

  \[
  \text{trace}(B_{zw}^T PB_w) < \gamma^2
  \]

- **Impulse-to-peak**

  \[
  A^T P + PA < 0 \quad B_{zw}^T PB_w < \gamma^2 1 \\
  C_z^T C_z < P \quad D_{zw}^T D_{zw} < \gamma^2 1
  \]
Robust performance analysis: $V(x, \Delta)$ parameter-dependent Lyapunov function.

Nominal analysis (LMI) $\rightarrow$ Robust analysis (NP-hard)

$$\exists P : \mathcal{L}_\Sigma(P) < 0 \quad \rightarrow \quad \forall \Delta \in \Delta , \ \exists P(\Delta) : \mathcal{L}_\Sigma(\Delta)(P(\Delta)) < 0$$

Test over sample values $\{\Delta_1\ldots N\} \in \Delta$ gives optimistic results

(some results exist if $\{\Delta_1\ldots N\}$ is uniform distribution of $\Delta$ and large $N$)
Some LMI results

- Robust performance analysis: $V(x, \Delta)$ parameter-dependent Lyapunov function.

- Nominal analysis (LMI) $\rightarrow$ Robust analysis (NP-hard)
  \[
  \exists P : \mathcal{L}_\Sigma(P) < 0 \quad \rightarrow \quad \forall \Delta \in \Delta, \exists P(\Delta) : \mathcal{L}_\Sigma(\Delta)(P(\Delta)) < 0
  \]

- Test over sample values $\{\Delta_1...N\} \in \Delta$ gives optimistic results (some results exist if $\{\Delta_1...N\}$ is uniform distribution of $\Delta$ and large $N$)

- Choice of $P(\Delta)$ for having a finite number of decision variables:
  
  - "Quadratic Stability": $P(\Delta) = P$
  - Polytopic PDLF: $P(\Delta) = \sum \xi_v P[v]$
  - $P(\Delta)$ polynomial w.r.t. $\xi_v$
  - Quadratic-LFT PDLF: $P(\Delta) = \begin{bmatrix} 1 & \Delta_C^T \end{bmatrix} \hat{P} \begin{bmatrix} 1 \\ \Delta_C \end{bmatrix}$
    \[\Delta_C = \Delta(1 - D_{\Delta\Delta})^{-1}C_{\Delta}\]
  - $P(\Delta)$ polynomial w.r.t. $\Delta_C$
Some LMI results

Robust performance analysis: $V(x, \Delta)$ parameter-dependent Lyapunov function.

Nominal analysis (LMI) $\rightarrow$ Robust analysis (NP-hard)

$$\exists \ P \ : \ \mathcal{L}_{\Sigma}(P) < 0 \ \rightarrow \ \forall \ \Delta \in \Delta, \ \exists \ P(\Delta) \ : \ \mathcal{L}_{\Sigma(\Delta)}(P(\Delta)) < 0$$

Test over sample values $\{\Delta_1...N\} \in \Delta$ gives optimistic results
(some results exist if $\{\Delta_1...N\}$ is uniform distribution of $\Delta$ and large $N$)

Choice of $P(\Delta)$ for having a finite number of decision variables:

- "Quadratic Stability": $P(\Delta) = P$
- Polytopic PDLF: $P(\Delta) = \sum \xi_v P[v]$
- Quadratic-LFT PDLF: $P(\Delta) = \begin{bmatrix} 1 & \Delta_T^T \end{bmatrix} \hat{P} \begin{bmatrix} 1 \\ \Delta_C \end{bmatrix}$

LMIs over infinite number of variables

$$\forall \ \Delta \in \Delta, \ \exists \ P(\Delta) \ : \ \mathcal{L}_{\Sigma(\Delta)}(P(\Delta)) < 0$$

$$\iff \exists \ P[v] \text{ or } \hat{P} \ : \ \forall \ \Delta \in \Delta, \ \mathcal{L}_{\Sigma(\Delta)}(P(\Delta)) < 0$$
Conservative LMIs for polytopic models (Example of stability analysis)

\[
\dot{x} = A(\Delta)x \quad \text{with} \quad A(\Delta) = \sum_{v=1}^{\bar{v}} \xi_v A^{[v]} : \quad \xi \in \Xi = \{ \xi_v \geq 0, \sum_{v=1}^{\bar{v}} \xi_v = 1 \}
\]
Some LMI results

- Conservative LMIs for polytopic models (Example of stability analysis)

\[ \dot{x} = A(\Delta) x \text{ with } A(\Delta) = \sum_{v=1}^{\bar{v}} \xi_v A^v : \xi \in \Xi = \{ \xi_v \geq 0, \sum_{v=1}^{\bar{v}} \xi_v = 1 \} \]

- “Quadratic Stability”: \( P(\Delta) = P \)

\[ \dot{V}(x) < 0 \iff A^T(\Delta) P + PA(\Delta) < 0 \iff A^v P + PA^v < 0 \]
Some LMI results

- Conservative LMIs for polytopic models (Example of stability analysis)

\[ \dot{x} = A(\Delta)x \] with \[ A(\Delta) = \sum_{v=1}^{\bar{v}} \xi_v A^v \] : \[ \xi \in \Xi = \{ \xi_v \geq 0, \sum_{v=1}^{\bar{v}} \xi_v = 1 \} \]

- "Quadratic Stability": \( P(\Delta) = P \)

\[ \dot{V}(x) < 0 \iff A^T(\Delta)P + PA(\Delta) < 0 \iff A^{[v]^T}P + PA^{[v]} < 0 \]

- Polytopic PDLF: \( P(\Delta) = \sum \xi_i P^{[i]} \)

\[ \begin{pmatrix} x \\ \dot{x} \end{pmatrix}^T \begin{bmatrix} 0 & P(\Delta) \\ P(\Delta) & 0 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} < 0 : \begin{bmatrix} A(\Delta) & -1 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} = 0 \]

\[ \iff \text{Finsler Lemma} \]

\[ \begin{bmatrix} 0 & P(\Delta) \\ P(\Delta) & 0 \end{bmatrix} + G(\Delta) \begin{bmatrix} A(\Delta) & -1 \end{bmatrix} + \begin{bmatrix} A^T(\Delta) \\ -1 \end{bmatrix} G^T(\Delta) < 0 \]

\[ \iff G(\Delta) = G \text{ & convexity} \]

\[ \begin{bmatrix} 0 & P^{[v]} \\ P^{[v]} & 0 \end{bmatrix} + G \begin{bmatrix} A^{[v]} & -1 \end{bmatrix} + \begin{bmatrix} A^{[v]^T} \\ -1 \end{bmatrix} G^T < 0 \]
Conservative LMIs for LFT models (Example of stability analysis)
\[ \dot{x} = Ax + B_{\Delta} w_{\Delta} \quad \text{with} \quad w_{\Delta} = \Delta z_{\Delta} = \Delta C_{\Delta} x + \Delta D_{\Delta\Delta} w_{\Delta} \]
Some LMI results

Conservative LMIs for LFT models (Example of stability analysis)

\[
\dot{x} = Ax + B_\Delta w_\Delta \quad \text{with} \quad w_\Delta = \Delta z_\Delta = \Delta C_\Delta x + \Delta D_{\Delta\Delta} w_\Delta
\]

"Quadratic Stability": \( P(\Delta) = P \)

\[
\dot{V}(x) < 0 \iff \begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \end{bmatrix}^* \begin{bmatrix} A^*P + PA & PB_\Delta \\ B_\Delta^TP & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \end{bmatrix} < 0 
\forall \Delta \in \mathcal{D}
\]

\( \iff \) Quadratic separation

\[
\begin{bmatrix} A^*P + PA & PB_\Delta \\ B_\Delta^TP & 0 \end{bmatrix} < \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \end{bmatrix}^T \Theta \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix} \Theta^T < 0 
\forall \Delta \in \mathcal{D}
\]
Some LMI results

Conservative LMIs for LFT models (Example of stability analysis)

\[ \dot{x} = Ax + B_\Delta w_\Delta \text{ with } w_\Delta = \Delta z_\Delta = \Delta C_\Delta x + \Delta D_{\Delta\Delta} w_\Delta \]

“Quadratic Stability”: \( P(\Delta) = P \)

\[ \dot{V}(x) < 0 \Leftrightarrow \begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \end{bmatrix}^* \begin{bmatrix} A^*P + PA & PB_\Delta \\ B_\Delta^TP & 0 \end{bmatrix} \begin{bmatrix} \Delta(1 - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \end{bmatrix} < 0 \]

\[ \Leftrightarrow \text{Quadratic separation} \]

\[ \begin{bmatrix} A^*P + PA & PB_\Delta \\ B_\Delta^TP & 0 \end{bmatrix} < \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \end{bmatrix}^T \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix} < 0 \quad \forall \Delta \in \Delta \]

Quadratic-LFT PDLF - same methodology (yet needs many matrix manipulations).
State-feedback designs results

“Quadratic Stability” case \( P(\Delta) = P \) - Procedure

1- Write the analysis LMI results for the closed-loop

\[
(A(\Delta) + B(\Delta)K)^T P + P(A(\Delta) + B(\Delta)K) < 0
\]

2- Write it for the ‘dual’ system (or perform congruence by \( X = P^{-1} \))

\[
(A(\Delta) + B(\Delta)K)X + X(A(\Delta) + B(\Delta)K)^T < 0
\]

3- Perform the change of variables \( S = KX \)

\[
(A(\Delta)X + B(\Delta)S)^T + (A(\Delta)X + B(\Delta)S) < 0
\]

4- Eliminate \( \Delta \) with previously explained methodology

\[
\]

5- If a solution is found \( K = SX^{-1} \) solves the robust design problem.
State-feedback designs results

- “Quadratic Stability” case $P(\Delta) = P$ - Multi-objective

\[ K_1 \quad K_1 \quad K_1 \]

\[ \Sigma_1 \quad \Pi_1 + \Sigma_2 \quad \Pi_2 + \Sigma_3 \quad \Pi_3 \]

\[ \mathcal{L}_{\Pi_1}(X_1, S_1) < 0 \quad \mathcal{L}_{\Pi_2}(X_2, S_2) < 0 \quad \mathcal{L}_{\Pi_3}(X_3, S_3) < 0 \]

Lyapunov Shaping Paradigm: Solve the set of LMI for the same $X = X_i, S = S_i$

\[ K = S X^{-1} \]

Procedure not fully applicable to Polytopic PDLF (non conservative structure on $G$?)

\[ \begin{bmatrix} 0 & X^{[v]} \\ X^{[v]} & 0 \end{bmatrix} + G^T \begin{bmatrix} A^{[v]}^T + K^T B^{[v]}^T & -1 \end{bmatrix} + \begin{bmatrix} A^{[v]} + B^{[v]} K \\ -1 \end{bmatrix} G < 0 \]

No results available for state-feedback with PDLF in the LFT case.
4 Some LMI results

- Full-order dynamic output-feedback results
- Only for some LFT models and “Quadratic Stability” case $P(\Delta) = P$
- Change of variables proposed by [Scherer 97]

- Other control design problems?
- Not LMI, or still some unknown changes of variables to be found
- Many (non LMI) results for static-output feedback $u = Ky$
  (includes all control problems)

- Software based on variational analysis - nonsmooth optimization
  ▲ HIFOO: H-Infinity Fixed Order Optimization
  www.cs.nyu.edu/overton/software/hifoo/
  ▲ [Noll, Apkarian] Sold to Matlab©

- Other solvers
  ▲ PENBMI in PENOPT www.penopt.com [Kocvara, Stingl]
1 Robust multi-objective problems

2 Linear Matrix Inequalities & Optimization tools

\[ A^T P + PA < 0 \iff Ax = b, \ x \in K \]

3 Manipulating inequalities to obtain LMI results

\[
\begin{align*}
A > BC^{-1}B^T \\
C > 0
\end{align*}
\implies
\begin{bmatrix}
A & B \\
B^T & C
\end{bmatrix} > 0
\]

\[
A^T(\zeta)P + PA(\zeta) < 0 \ \forall \zeta \in \Delta
\]

\[
\updownarrow
\]

\[
A^{[v]} P + PA^{[v]} < 0 \ \forall v \in \{1...\bar{v}\}
\]

4 Some LMI results

5 Solving robust multi-objective problems in RoMulOC

\[
\text{quiz} = \text{ctrpb}('sf', 'unique') + \text{h2}(\text{usys_h2}) + \text{hinf}(\text{usys_hinf}, 10);
\]
■ "Helicopter" example

● System defined at maximal value of parameters

```matlab
>> sysmax = ssmodel('Helicopter');
>> sysmax.A = [0 1 0; 0 0 1; 0 -2.8 -0.14];
>> sysmax.Bw = [0;0;-14];
>> sysmax.Bu = [0;0;8];
>> sysmax.Dzu = 1
name: Helicopter
   n=3     mw=1     mu=1
   n=3    dx = A*x + Bw*w + Bu*u
pz=1     z = Dzu*u
continuous time ( dx : derivative operator )
```
Solving robust multi-objective problems in RoMuOC

- System defined at maximal value of parameters

```matlab
>> sysmax = ssmodel( 'Helicopter' );
>> sysmax.A = [0 1 0; 0 0 1; 0 -2.8 -0.14];
>> sysmax.Bw = [0;0;-14];
>> sysmax.Bu = [0;0;8];
>> sysmax.Dzu = 1;
```

- System defined at minimal value of parameters

```matlab
>> sysmin = ssmodel( 'Helicopter' );
>> sysmin.A = [0 1 0; 0 0 1; 0 -3 -0.2];
>> sysmin.Bw = [0;0;-14];
>> sysmin.Bu = [0;0;8];
>> sysmin.Dzu = 1
name: Helicopter
    n=3  mw=1  mu=1
    n=3  dx = A*x + Bw*w + Bu*u
    pz=1  z = Dzu*u
continuous time ( dx : derivative operator )
```
Uncertain system defined as interval of max and min

```matlab
>> usys = uinter( sysmin, sysmax )
Uncertain model : interval 2 param
-------- WITH --------
name: Helicopter
    n=3  mw=1  mu=1
    n=3  dx = A*x  +  Bw*w  +  Bu*u
pz=1  z = Dzu*u
continuous time ( dx : derivative operator )
```

Interval model converted to polytopic model

```matlab
>> usys = u2poly( usys )
Uncertain model : polytope 4 vertices
-------- WITH --------
name: Helicopter
    n=3  mw=1  mu=1
    n=3  dx = A*x  +  Bw*w  +  Bu*u
pz=1  z = Dzu*u
continuous time ( dx : derivative operator )
```
• Declare a state-feedback design problem

```matlab
>> quiz = ctrpb( 'state-feedback', 'Lyap-unique' )
control problem: STATE-FEEDBACK design
Lyapunov function: UNIQUE (quadratic stability)
No specified performance
```

• Add an $H_\infty$ performance objective

```matlab
>> quiz = quiz + hinfty( usys, 4 );
```

• Add a pole location performance objective

```matlab
>> r = region( 'plane', -0.1 )
Half-plane such that: Re(z)<-0.1
>> quiz = quiz + dstability( usys, r )
```

• Add an impulse-to-peak performance minimization objective

```matlab
>> quiz = quiz + i2p( usys )
control problem: STATE-FEEDBACK design
Lyapunov function: UNIQUE (quadratic stability)
Specified performances / systems:
# Hinfty < 4 / Helicopter
# D-stability / Helicopter
# minimize I2P / Helicopter
```
The quiz object

control problem: STATE-FEEDBACK design
Lyapunov function: UNIQUE (quadratic stability)
Specified performances / systems:
# H_{\infty} < 4 / Helicopter
# D-stability / Helicopter
# minimize I2P / Helicopter

Contains decision variables

```
>> quiz.vars
[3x3 sdpvar] 'Lyapunov matrix'
[1x3 sdpvar] 'S=-K*P'
[1x1 sdpvar] 'S-procedure scaling'
[1x1 sdpvar] 'g > (I2P cost)^2'
```
Solving robust multi-objective problems in RoMuLOC

Constrained by LMIs

```plaintext
>> quiz.lmi

+----------------------------------------------------------------+
<table>
<thead>
<tr>
<th>ID</th>
<th>Constraint</th>
<th>Type</th>
<th>Tag</th>
</tr>
</thead>
</table>
+----------------------------------------------------------------+
| #1 | Numeric value | Matrix inequality 3x3 | Lyap >0 |
| #2 | Numeric value | Matrix inequality 4x4 | Var Lyap <0 |
| #3 | Numeric value | Matrix inequality 4x4 | Var Lyap <0 |
| #4 | Numeric value | Matrix inequality 4x4 | Var Lyap <0 |
| #5 | Numeric value | Matrix inequality 4x4 | Var Lyap <0 |
| #6 | Numeric value | Matrix inequality 3x3 | Var Lyap <0 |
| #7 | Numeric value | Matrix inequality 3x3 | Var Lyap <0 |
| #8 | Numeric value | Matrix inequality 3x3 | Var Lyap <0 |
| #9 | Numeric value | Matrix inequality 3x3 | Var Lyap <0 |
| #10| Numeric value | Matrix inequality 3x3 | Constraint 1 |
| #11| Numeric value | Matrix inequality 4x4 | Constraint 2 |
| #12| Numeric value | Matrix inequality 3x3 | Constraint 3 |
| #13| Numeric value | Element-wise 1x1 | Constraint 4 |
| #14| Numeric value | Matrix inequality 3x3 | Constraint 1 |
| #15| Numeric value | Matrix inequality 4x4 | Constraint 2 |
| #16| Numeric value | Matrix inequality 3x3 | Constraint 3 |
| #17| Numeric value | Element-wise 1x1 | Constraint 4 |
+----------------------------------------------------------------+
```
<table>
<thead>
<tr>
<th>#</th>
<th>Numeric value</th>
<th>Matrix inequality 3x3</th>
<th>Constraint 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>#18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Numeric value</th>
<th>Matrix inequality 4x4</th>
<th>Constraint 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>#19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Numeric value</th>
<th>Matrix inequality 3x3</th>
<th>Constraint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Numeric value</th>
<th>Element-wise 1x1</th>
<th>Constraint 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>#21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#25</th>
<th>Numeric value</th>
<th>Element-wise 1x1</th>
<th>Constraint 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#25</th>
<th>Numeric value</th>
<th>Element-wise 1x1</th>
<th>Constraint 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
And can be solved (SeDuMi solver by default)

```matlab
>> K = solvesdp( quiz )
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 11, order n = 76, dim = 250, blocks = 22
nnz(A) = 282 + 0, nnz(ADA) = 117, nnz(L) = 64
```

<table>
<thead>
<tr>
<th>it</th>
<th>b*y</th>
<th>gap</th>
<th>delta</th>
<th>rate</th>
<th>t/tP*</th>
<th>t/tD*</th>
<th>feas</th>
<th>cg</th>
<th>cg</th>
<th>prec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.96E+01</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.0E+02</td>
</tr>
<tr>
<td>1</td>
<td>-1.79E+02</td>
<td>1.85E+01</td>
<td>0.000</td>
<td>0.2657</td>
<td>0.9000</td>
<td>0.9000</td>
<td>-0.09</td>
<td>1</td>
<td>1</td>
<td>5.0E+02</td>
</tr>
<tr>
<td>2</td>
<td>-1.05E+02</td>
<td>5.96E+00</td>
<td>0.000</td>
<td>0.3223</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.55</td>
<td>1</td>
<td>1</td>
<td>1.1E+02</td>
</tr>
<tr>
<td>3</td>
<td>-2.56E+01</td>
<td>1.38E+00</td>
<td>0.000</td>
<td>0.2312</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.73</td>
<td>1</td>
<td>1</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>4</td>
<td>-5.54E+00</td>
<td>2.62E-01</td>
<td>0.000</td>
<td>0.1902</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.21</td>
<td>1</td>
<td>1</td>
<td>3.2E+00</td>
</tr>
<tr>
<td>5</td>
<td>-1.84E+00</td>
<td>8.00E-02</td>
<td>0.000</td>
<td>0.3050</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.29</td>
<td>1</td>
<td>1</td>
<td>8.3E-01</td>
</tr>
<tr>
<td>6</td>
<td>-7.08E-01</td>
<td>2.90E-02</td>
<td>0.000</td>
<td>0.3621</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.35</td>
<td>1</td>
<td>1</td>
<td>2.6E-01</td>
</tr>
<tr>
<td>7</td>
<td>-2.95E-01</td>
<td>1.05E-02</td>
<td>0.000</td>
<td>0.3637</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.27</td>
<td>1</td>
<td>1</td>
<td>8.3E-02</td>
</tr>
<tr>
<td>8</td>
<td>-2.30E-01</td>
<td>3.57E-03</td>
<td>0.000</td>
<td>0.3393</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.12</td>
<td>1</td>
<td>1</td>
<td>2.7E-02</td>
</tr>
<tr>
<td>9</td>
<td>-1.97E-01</td>
<td>6.73E-04</td>
<td>0.000</td>
<td>0.1882</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>5.1E-03</td>
</tr>
<tr>
<td>10</td>
<td>-1.91E-01</td>
<td>2.02E-05</td>
<td>0.000</td>
<td>0.0300</td>
<td>0.9900</td>
<td>0.9900</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>1.6E-04</td>
</tr>
<tr>
<td>11</td>
<td>-1.91E-01</td>
<td>1.13E-06</td>
<td>0.000</td>
<td>0.0558</td>
<td>0.9900</td>
<td>0.9900</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>8.7E-06</td>
</tr>
<tr>
<td>12</td>
<td>-1.91E-01</td>
<td>3.08E-07</td>
<td>0.000</td>
<td>0.2737</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>2.4E-06</td>
</tr>
<tr>
<td>13</td>
<td>-1.91E-01</td>
<td>1.33E-08</td>
<td>0.000</td>
<td>0.0433</td>
<td>0.9900</td>
<td>0.9900</td>
<td>1.00</td>
<td>1</td>
<td>1</td>
<td>1.0E-07</td>
</tr>
<tr>
<td>14</td>
<td>-1.91E-01</td>
<td>3.01E-09</td>
<td>0.000</td>
<td>0.2261</td>
<td>0.9000</td>
<td>0.9000</td>
<td>1.00</td>
<td>2</td>
<td>2</td>
<td>2.3E-08</td>
</tr>
</tbody>
</table>
Solving robust multi-objective problems in RoMuLOC

15 : -1.91E-01 7.53E-10 0.000 0.2498 0.9000 0.9000 1.00 2 2 5.8E-09
16 : -1.91E-01 4.50E-11 0.087 0.0598 0.9900 0.9900 1.00 2 2 3.5E-10

iter seconds digits c\times x b \times y
16 0.4 Inf -1.9059654950e-01 -1.9059654919e-01
|Ax-b| = 3.6e-10, [Ay-c]_+ = 2.6E-11, |x|= 5.0e-01, |y|= 3.7e+02

Detailed timing (sec)

<table>
<thead>
<tr>
<th>Pre</th>
<th>IPM</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.800E-01</td>
<td>4.000E-01</td>
<td>7.000E-02</td>
</tr>
</tbody>
</table>

Max-norms: \|b\|=1, \|c\|= 196,
Cholesky |add|=0, |skip| = 0, \|L.L\|= 42153.4.

Feasibility is not strictly determined
Worst constraint residual is -2.59066e-11 < 0

0.436574 (=\sqrt{\text{double(CTRPB.vars\{4\})}}) may be a guaranteed I2P norm

K =
0.0442 0.0091 0.0305
Welcome to new RoMulOC users

http://www.laas.fr/OLOCEP/romuloc

Version 1 - started in June 2005
Contains robust analysis results presented in this talk.
LMIs are coded using YALMIP parser, all available SDP solvers can be used.

Version 2 - started in February 2007
Includes design facilities for robust multi-objective state-feedback.
Coding LMI results for full order output-feedback design is planned.

Version 3 - maybe this year
Would include heuristic tools for solving static output-feedback design problems.
Conclusions

- Robustness: important issue in control theory
- Robustness results have impact for many other problems

- LMI: central tool for robustness
- Effective to transfer theory to industrial applications with software

- Quadratic Separation framework not yet fully exploited
- Extensions to non-linearities, time-varying operators...
- Descriptor version of RoMuLOC: Romuald