Получение результатов в виде линейныш матречных 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

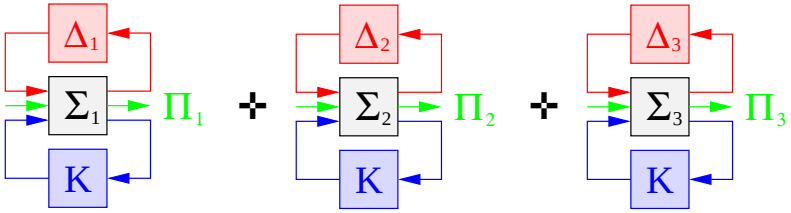
peaucelle@laas.fr http://homepages.laas.fr/peaucell

Traditional School

"Control, Information and Optimization"

Pereslavl'-Zalesskii June 2010

❶ Robust multi-objective problems



❷ Linear Matrix Inequalities & Optimization tools $A^T P + PA < 0 \leftrightarrow \mathsf{A}\mathsf{x} = \mathsf{b}\,,\ \mathsf{x} \in \mathcal{K}$

❸ Manipulating inequalities to obtain LMI results

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

❹ Some LMI results

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

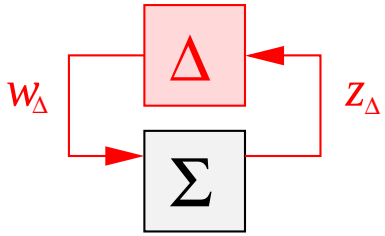
$$\Updownarrow$$

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

❺ Solving robust multi-objective problems in RoMulOC

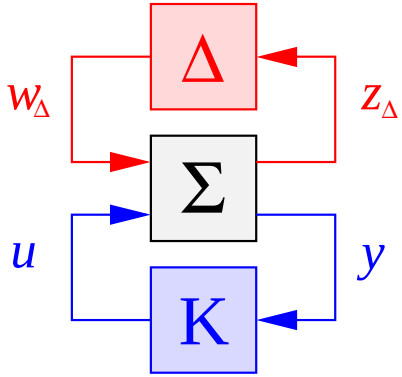
```
» quiz= ctrpb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);
```

- Standard robust analysis problem:



prove stability of $\left(\Sigma \overset{w_\Delta, z_\Delta}{\star} \Delta\right)$ for all $\Delta \in \mathbb{\Delta}$

- Standard robust design problem:



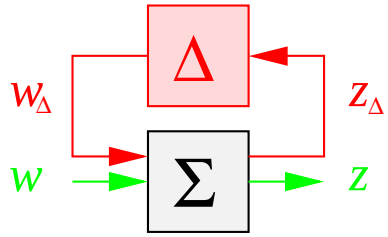
Find K that guarantees stability of $\left(\left(\Sigma \overset{w_\Delta, z_\Delta}{\star} \Delta\right) \overset{u,y}{\star} K\right)$ for all $\Delta \in \mathbb{\Delta}$

- $\mathbb{\Delta}$ contains unknown parameters, scheduling parameters,

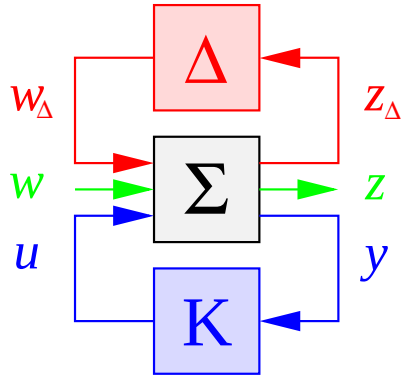
  approximations of non-linearities, delays...

- Σ is a linear model: crude but simple representation of the system

- Generalizes to robust performance problems

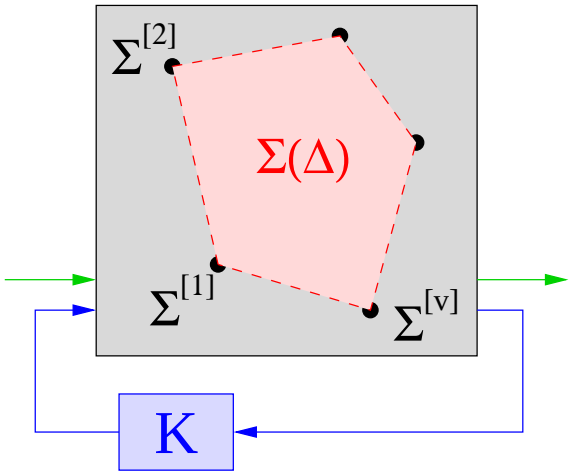


- Guarantee an input/output property for all $\Delta \in \mathbb{\Delta}$



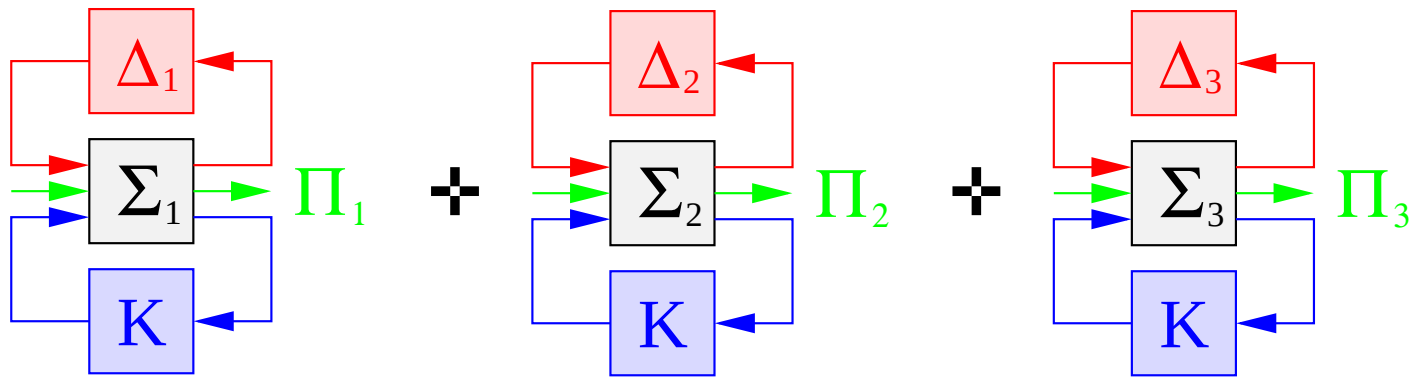
- Find a controller that guarantees input/output properties for all $\Delta \in \mathbb{\Delta}$

- Same holds for polytopic-type models



Robust multi-objective problem

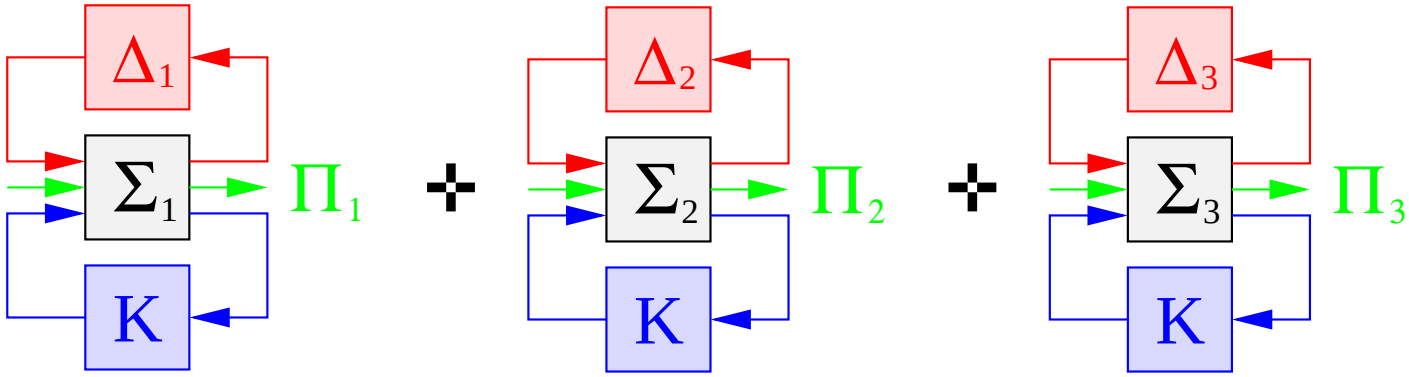
- Find a controller K that guarantees simultaneously several robust specifications $\Pi_1, \Pi_2, \ldots$ each of which being defined for a possibly different models $(\Sigma_1 \star \Delta_1), (\Sigma_2 \star \Delta_2), \ldots$



- Example: Robust H_2/H_∞ problem

$$\min \left\| \begin{array}{l} \dot{x} = A(\Delta)x + B_u(\Delta)u \\ z = u \\ y = C_y(\Delta)x + w \\ u = Ky \end{array} \right\|_2 : \left\| \begin{array}{l} \dot{x}_f = A_f x_f + B_f w \\ \dot{x} = A(\Delta)x + B_w(\Delta)x_f + B_u(\Delta)u \\ z = C_z(\Delta)x \\ y = C_y(\Delta)x \\ u = Ky \end{array} \right\|_\infty \leq \gamma_\infty$$

❶ Robust multi-objective problems

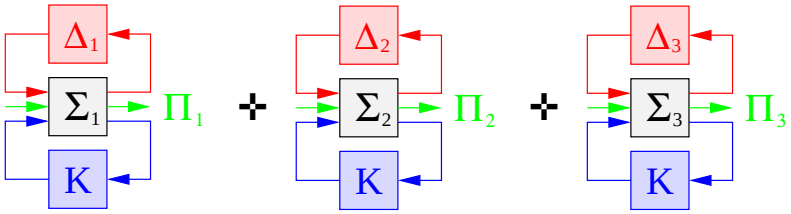


- Naturally defined as existence (feasibility) problem over several constraints
- While a nominal performance $\|\Sigma \star K\| = \gamma$ may be defined by an equality
- Robust performance $\|(\Sigma \star \Delta) \star K\| \leq \gamma,\ \forall \Delta \in \mathbb{\Delta}$ can only be an inequality
- Finding the 'best' robust controller: optimization problem over inequality constraints

▲ What type of optimization problem? Unique optimum? Convex? Convergence time? ...

▲ LMI: Upper-bounds, Convex optimization, polynomial-time.

❶ Robust multi-objective problems



❷ Linear Matrix Inequalities & Optimization tools $A^T P + PA < 0 \leftrightarrow \mathsf{A}\mathsf{x} = \mathsf{b}\,,\ \mathsf{x} \in \mathcal{K}$

❸ Manipulating inequalities to obtain LMI results

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

❹ Some LMI results

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

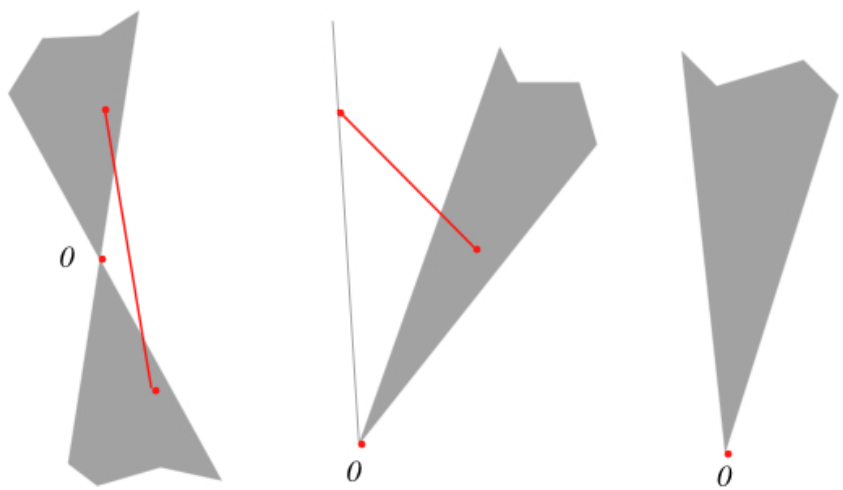
$$\Updownarrow$$

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

❺ Solving robust multi-objective problems in RoMulOC

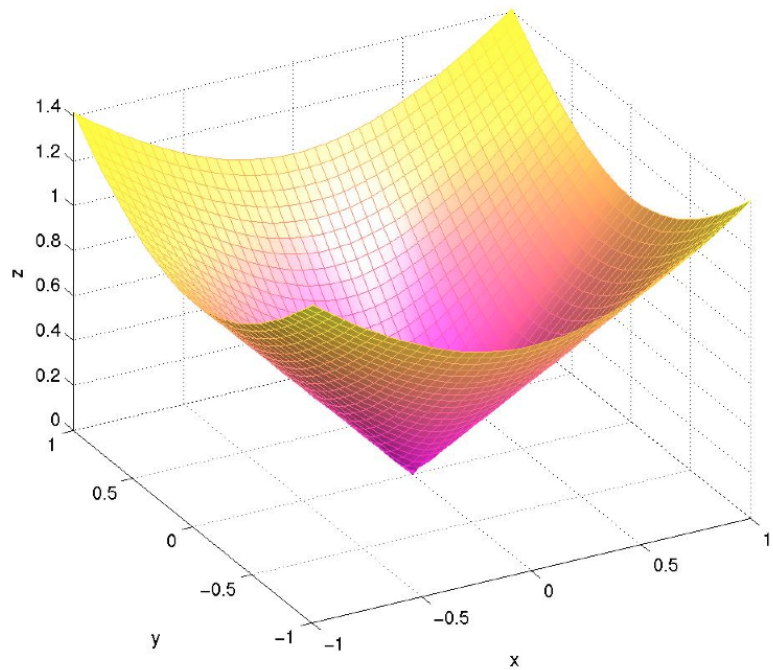
```
» quiz= ctrpb('sf','unique')+h2(usys_h2)+hinf(usys_hinf,10);
```

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

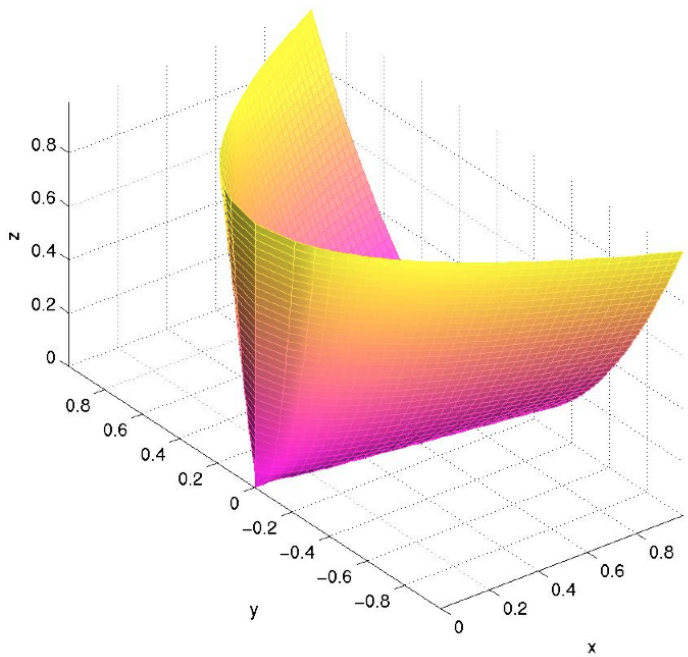
$\mathcal{K}^3_{soc}$:



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

$$\mathcal{K}^n_{psd} = \left\{ \begin{array}{l} x = \\ \begin{pmatrix} x_1 & \cdots & x_{n^2} \end{pmatrix} \end{array} , \begin{array}{l} \mathrm{mat}(x) \\ = \mathrm{mat}(x)^T \end{array} = \begin{bmatrix} x_1 & x_{n+1} & \cdots & x_{n(n-1)+1} \\ \vdots & & & \vdots \\ x_n & x_{2n} & \cdots & x_{n^2} \end{bmatrix} \geq 0 \right\}$$

$\mathcal{K}^2_{psd}$:



$$\begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix} \geq 0$$

- Convex cones
  - Convex cone of positive reals: $x \in \mathrm{R}_+$
  - Second order (Lorentz) cone: $\mathcal{K}^n_{soc} = \left\{ x = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \ , \ x_1^2 + \dots x_{n-1}^2 \leq x_n^2 \right\}$
  - Positive semi-definite matrices: $\mathcal{K}^n_{psd} = \left\{ x = \begin{pmatrix} x_1 & \cdots & x_{n^2} \end{pmatrix} \ , \ \mathrm{mat}(x) \geq 0 \right\}$
  - Unions of such: $\mathcal{K} = \mathrm{R}_+ \times \cdots \times \mathcal{K}^{n_1}_{soc} \times \dots \times \mathcal{K}^{n_q}_{psd} \times \cdots$

- Optimization over convex cones

$$p^\star = \min\ cx \quad : \quad Ax = b \quad , \quad x \in \mathcal{K}$$

- Linear programming: $\mathcal{K} = \mathsf{R}_+ \times \cdots \mathsf{R}_+$.
- Semi-definite programming: $\mathcal{K} = \mathcal{K}_{psd}^{n_1} \times \cdots \mathcal{K}_{psd}^{n_q}$

- Dual problem

$$d^\star = \max\ b^T y \quad : \quad A^T y - c^T = z \quad , \quad z \in \mathcal{K}$$

- Primal feasible $\longrightarrow$ Dual infeasible
- Dual feasible $\longrightarrow$ Primal infeasible
- If primal and dual strictly feasible $p^\star = d^\star$

- Polynomial-time algorithms ($\mathcal{O}(n^{6.5}\log(1/\epsilon))$)

- Optimization over convex cones

$$p^\star = \min\ cx \quad : \quad Ax = b \quad , \quad x \in \mathcal{K}$$

- Dual problem

$$d^\star = \max\ b^T y \quad : \quad A^T y - c^T = z \quad , \quad z \in \mathcal{K}$$

- Possibility to perform convex optimization, primal/dual, interior-point methods, etc.
- Interior-point methods [Nesterov, Nemirovski 1988] - Matlab Control Toolbox [Gahinet et al.]
- 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
$$\begin{cases} p^\star = \min\ cx \quad : \quad Ax = b \quad , \quad x \in \mathcal{K} \\ d^\star = \max\ b^T y \quad : \quad A^T y - c^T = z \quad , \quad z \in \mathcal{K} \end{cases}$$

- LMI formalism
$$\begin{cases} d^\star = \min \sum g_i y_i \quad : \quad F_0 + \sum F_i y_i \geq 0 \\ p^\star = \max \mathrm{Tr}(F_0 X) \quad : \quad \mathrm{Tr}(F_i X) + g_i = 0 \quad , \quad X \geq 0 \end{cases}$$

- In control problems: variables are matrices

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

$$\|G(s)\|_\infty^2 \quad = \quad \min\ \gamma \quad : \quad P > 0 \ , \quad \underbrace{\begin{bmatrix} A^T P + PA + C_z^T C_z & B_w P + C_z^T D_{zw} \\ P B_w^T + D_{zw}^T C_z & -\gamma 1 + D_{zw}^T D_{zw} \end{bmatrix}}_{\begin{bmatrix} C_z^T \\ D_{zw}^T \end{bmatrix} \begin{bmatrix} C_z^T \\ D_{zw}^T \end{bmatrix}^T - \gamma \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T + p_{11} \ldots} < 0$$

- Need for a nice parser

- Parsers: LMIlab, tklmitool, sdpsol, SeDuMiInterface...
- YALMIP
  - Convert LMIs to SDP solver format (all available solvers!)
  - Simple to use

```
>> 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 ⇛ 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 ⇛ 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.

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)

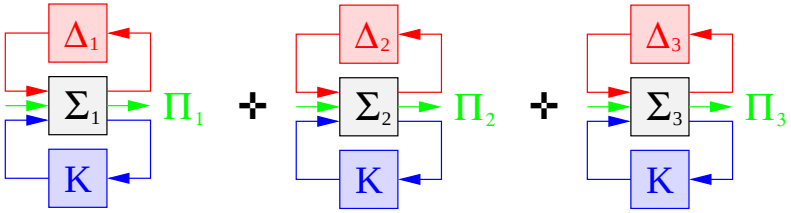
- Any "SDP-able" problem has a dual interpretation
  - New theoretical results (worst case)
  - New proofs (Lyapunov functions = Lagrange multipliers; related to SOS)
  - SDP formulas numerically stable (KYP-lemma)

- 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

❶ Robust multi-objective problems



❷ Linear Matrix Inequalities & Optimization tools $A^T P + PA < 0 \leftrightarrow \mathsf{A}\mathsf{x} = \mathsf{b}\,,\ \mathsf{x} \in \mathcal{K}$

❸ Manipulating inequalities to obtain LMI results

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

❹ Some LMI results

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

$$\Updownarrow$$

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

❺ Solving robust multi-objective problems in RoMulOC

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

❸ 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: $B^T A B \geq 0.$
- $A > 0 \Leftrightarrow$ exists a square non-singular matrix B: $B^T A B > 0.$

▲ Most LMI results are formulated as (sufficiency)

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

Proof: $V(x) = x^T P x > 0, \ \dot{V}(x) = x^T (A^T P + P A) x = 2 \dot{x}^T P x < 0$ for all $x \neq 0$.

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

- Stability (discrete-time) $A^T P A - P < 0$
- Regional pole placement $\begin{bmatrix} 1 & A^* \end{bmatrix} \begin{bmatrix} r_{11}P & r_{12}P \\ r_{12}^* P & r_{22}P \end{bmatrix} \begin{bmatrix} 1 \\ A \end{bmatrix} < 0$
- H_∞ norm $\begin{bmatrix} A^T P + PA + C_z^T C_z & PB_w + C_z^T D_{zw} \\ B_w^T P + D_{zw}^T C_z & -\gamma^2 1 + D_{zw}^T D_{zw} \end{bmatrix} < 0$
- H_2 norm $A^T P + PA + C_z^T C_z < 0$

  trace$(B_w^T P B_w) < \gamma^2$
- Impulse-to-peak $A^T P + PA < 0 \quad B_w^T P B_w < \gamma^2 1$

  $C_z^T C_z < P \quad D_{zw}^T D_{zw} < \gamma^2 1$

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- Schur complement

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

- Example:

$$\left\{ \begin{array}{l} (AX + BS)X^{-1}(XA^T + S^TB^T) - X < 0 \\ X > 0 \end{array} \right.$$

$$\Leftrightarrow \left[\begin{array}{cc} -X & AX + BS \\ XA^T + S^TB^T & -X \end{array} \right] < 0$$

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- Finsler lemma - Elimination lemma - Creation lemma

$$
\begin{aligned}
x^T A x < 0 \;\; \forall x \; : \; Bx = 0 \quad &\Leftrightarrow \quad \exists \tau \in \mathrm{R} \;\; : \;\; A < \tau B^T B \\
&\Leftrightarrow \quad \exists X = X^T \;\; : \;\; A < B^T X B \\
&\Leftrightarrow \quad \exists G \;\; : \;\; A < B^T G^T + G B \\
&\Leftrightarrow \quad B^{\perp T} A B^{\perp} < 0
\end{aligned}
$$

- where $B^{\perp}$ columns generate the null space of B:

$$B \in \mathrm{R}^{p \times m} \;\; , \;\; \mathrm{rank}(B) = r < m \;\; , \;\; B B^{\perp} = 0 \;\; , \;\; B^{\perp} \in \mathrm{R}^{m \times (m-r)} \;\; , \;\; B^{\perp T} B^{\perp} > 0$$

- G is a 'Slack variable' (Lagrange multiplier)

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- Finsler lemma - Elimination lemma - Creation lemma

$$
\begin{aligned}
x^T A x < 0 \;\; \forall x \;:\; Bx = 0 \quad &\Leftrightarrow \quad \exists \tau \in \mathsf{R} \;:\; A < \tau B^T B \\
&\Leftrightarrow \quad \exists X = X^T \;:\; A < B^T X B \\
&\Leftrightarrow \quad \exists G \;:\; A < B^T G^T + G B \\
&\Leftrightarrow \quad B^{\perp T} A B^{\perp} < 0
\end{aligned}
$$

- Example:

$$
\dot{V}(x) = \begin{pmatrix} \dot{x} \\ x \end{pmatrix}^T \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} < 0 \;,\; \forall \begin{bmatrix} 1 & -A \end{bmatrix} \begin{pmatrix} \dot{x} \\ x \end{pmatrix} = 0
$$

$$
\Leftrightarrow \exists G \;:\; \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} + \begin{bmatrix} -1 \\ A^T \end{bmatrix} G^T + G \begin{bmatrix} -1 & A \end{bmatrix} < 0
$$

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- Finsler lemma - Elimination lemma - Creation lemma

$$\begin{cases} C^{\perp T} A C^{\perp} < 0 \\ B^{\perp T} A B^{\perp} < 0 \end{cases} \quad \Leftrightarrow \quad \exists H \ : \ A < B^T H^T C + C^T H B$$

- Example

$$\begin{cases} -P = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} < 0 \\ A^T P A - P = \begin{bmatrix} A & 1 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} A \\ 1 \end{bmatrix} < 0 \end{cases}$$

$$\Leftrightarrow \quad \exists H \ : \ \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} < \begin{bmatrix} -1 \\ A^T \end{bmatrix} H^T \begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} H \begin{bmatrix} -1 & A \end{bmatrix}$$

❸ Manipulating inequalities to obtain LMI results

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

$$x^T M x < 0 \;\; \forall x \; : \; x^T N x \leq 0 \quad \Leftrightarrow \quad \exists \tau > 0 \; : \; M < \tau N$$

- Example:

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

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

$$\Leftrightarrow \quad \exists \tau > 0 \; : \; M < \tau \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

❸ Manipulating inequalities to obtain LMI results

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

$$x^T M x < 0 \;\; \forall x \; : \; x^T N x \leq 0 \quad \Leftrightarrow \quad \exists \tau > 0 \; : \; M < \tau N$$

- Example:

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

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

$$z_\Delta = Cx + Dw_\Delta$$

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

- Special case:

$$X + C^T \Delta^T B^T + B \Delta C < 0, \quad \forall \Delta^T \Delta \leq 1 \quad \Leftrightarrow \quad \exists \tau > 0 \; : \; X + \tau C^T C + \tau^{-1} B B^T < 0$$

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- D and DG-scalling

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

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

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

$$\Leftrightarrow \quad \exists Q > 0, \ T = -T^T \ : \ M < \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} -Q & T \\ T^T & Q \end{bmatrix} \begin{bmatrix} C & D \\ 0 & 1 \end{bmatrix}$$

❸ Manipulating inequalities to obtain LMI results

- Tools to 'build' LMI results
- Kalman-Yakubovich-Popov KYP lemma

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

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

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

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

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

- Difficulty: build the 'separator' Θ, losslessly...

❸ Manipulating inequalities to obtain LMI results

■ Tools to 'build' LMI results

● S-procedure [Yakubovich]

$$x^T M x < 0 \;\; \forall x \; : \; x^T N x \leq 0 \quad \Leftrightarrow \quad \exists \tau > 0 \;\; : \;\; M < \tau N$$

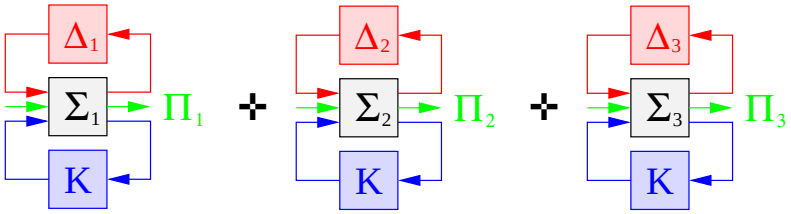
$$x^T M x < 0 \;\; \forall x \; : \; \left\{ \begin{array}{l} x^T N_1 x \leq 0 \\ \quad \vdots \\ x^T N_p x \leq 0 \end{array} \right. \quad \Leftarrow \quad \begin{array}{l} \exists \tau_1 > 0, \ldots \tau_p > 0 \;\; : \\ M < \tau_1 N_1 + \cdots \tau_p N_p \end{array}$$

▲ Not lossless except in few special cases: Δ composed of

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

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

❶ Robust multi-objective problems



❷ Linear Matrix Inequalities & Optimization tools $A^T P + PA < 0 \leftrightarrow \mathsf{A}\mathsf{x} = \mathsf{b}\,,\ \mathsf{x} \in \mathcal{K}$

❸ Manipulating inequalities to obtain LMI results

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

❹ Some LMI results

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

$$\Updownarrow$$

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

❺ Solving robust multi-objective problems in RoMulOC

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

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

- Stability (discrete-time) $A^T P A - P < 0$
- Regional pole placement $\begin{bmatrix} 1 & A^* \end{bmatrix} \begin{bmatrix} r_{11}P & r_{12}P \\ r_{12}^* P & r_{22}P \end{bmatrix} \begin{bmatrix} 1 \\ A \end{bmatrix} < 0$
- H_∞ norm $\begin{bmatrix} A^T P + PA + C_z^T C_z & PB_w + C_z^T D_{zw} \\ B_w^T P + D_{zw}^T C_z & -\gamma^2 1 + D_{zw}^T D_{zw} \end{bmatrix} < 0$
- H_2 norm $A^T P + PA + C_z^T C_z < 0$

  trace$(B_w^T P B_w) < \gamma^2$
- Impulse-to-peak $A^T P + PA < 0 \quad B_w^T P B_w < \gamma^2 1$

  $C_z^T C_z < P \quad D_{zw}^T D_{zw} < \gamma^2 1$

❹ Some LMI results

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

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

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

- Test over sample values $\{\Delta_{1\ldots N}\} \in \mathbb{\Delta}$ gives optimistic results

  (some results exist if $\{\Delta_{1\ldots N}\}$ is uniform distribution of $\mathbb{\Delta}$ and large N)

❹ Some LMI results

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

▲ Nominal analysis (LMI) $\longrightarrow$ Robust analysis (NP-hard)

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

▲ Test over sample values $\{\Delta_{1\ldots N}\} \in \mathbb{\Delta}$ gives optimistic results

(some results exist if $\{\Delta_{1\ldots N}\}$ is uniform distribution of $\mathbb{\Delta}$ and large N)

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

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

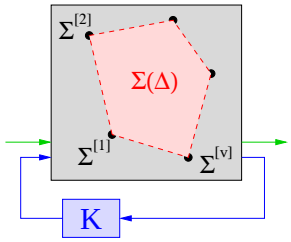
➙ Polytopic PDLF: $P(\Delta) = \sum \xi_v P^{[v]}$

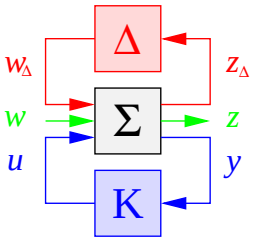
➙ $P(\Delta)$ polynomial w.r.t. ξ_v

➙ Quadratic-LFT PDLF: $P(\Delta) = \begin{bmatrix} 1 & \Delta_C^T \end{bmatrix} \hat{P} \begin{bmatrix} 1 \\ \Delta_C \end{bmatrix}$

$$\Delta_C = \Delta(1 - D_{\Delta\Delta}\Delta)^{-1} C_{\Delta}$$

➙ $P(\Delta)$ polynomial w.r.t. Δ_C





❹ Some LMI results

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

▲ Nominal analysis (LMI) $\longrightarrow$ Robust analysis (NP-hard)

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

▲ Test over sample values $\{\Delta_{1\ldots N}\} \in \mathbb{\Delta}$ gives optimistic results

(some results exist if $\{\Delta_{1\ldots N}\}$ is uniform distribution of $\mathbb{\Delta}$ 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: $P(\Delta) = \begin{bmatrix} 1 & \Delta_C^T \end{bmatrix} \hat{P} \begin{bmatrix} 1 \\ \Delta_C \end{bmatrix}$

▲ LMIs over infinite number of variables

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

$$\Leftarrow \exists\, P^{[v]} \text{or} \hat{P} \;:\; \forall\, \Delta \in \mathbb{\Delta}\,,\; \mathcal{L}_{\Sigma(\Delta)}(P(\Delta)) < 0$$

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

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

- 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": $P(\Delta) = P$

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

■ 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": $P(\Delta) = P$

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

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

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

$\Leftrightarrow$ Finsler Lemma

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

$\Leftarrow$ $G(\Delta) = G$ & 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)

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

- 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": $P(\Delta) = P$

$$\dot{V}(x) < 0 \Leftrightarrow$$

$$\begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1} C_\Delta \end{bmatrix}^* \begin{bmatrix} A^* P + PA & PB_\Delta \\ B_\Delta^T P & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1} C_\Delta \end{bmatrix} < 0$$

$$\forall \Delta \in \mathbb{\Delta}$$

$\Leftrightarrow$ Quadratic separation

$$\begin{bmatrix} A^* P + PA & PB_\Delta \\ B_\Delta^T P & 0 \end{bmatrix} < \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix}^T \Theta \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix}$$

$$, \quad \begin{bmatrix} 1 \\ \Delta \end{bmatrix}^T \Theta \begin{bmatrix} 1 \\ \Delta \end{bmatrix} < 0 \quad \forall \Delta \in \mathbb{\Delta}$$

- 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": $P(\Delta) = P$

$$\dot{V}(x) < 0 \Leftrightarrow$$

$$\begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1} C_\Delta \end{bmatrix}^* \begin{bmatrix} A^* P + PA & PB_\Delta \\ B_\Delta^T P & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \Delta(1 - D_{\Delta\Delta}\Delta)^{-1} C_\Delta \end{bmatrix} < 0$$

$$\forall \Delta \in \mathbb{\Delta}$$

$\Leftrightarrow$ Quadratic separation

$$\begin{bmatrix} A^* P + PA & PB_\Delta \\ B_\Delta^T P & 0 \end{bmatrix} < \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix}^T \Theta \begin{bmatrix} C_\Delta & D_{\Delta\Delta} \\ 0 & 1 \end{bmatrix}$$

$$, \quad \begin{bmatrix} 1 \\ \Delta \end{bmatrix}^T \Theta \begin{bmatrix} 1 \\ \Delta \end{bmatrix} < 0 \quad \forall \Delta \in \mathbb{\Delta}$$

- 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 $S = KX$

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

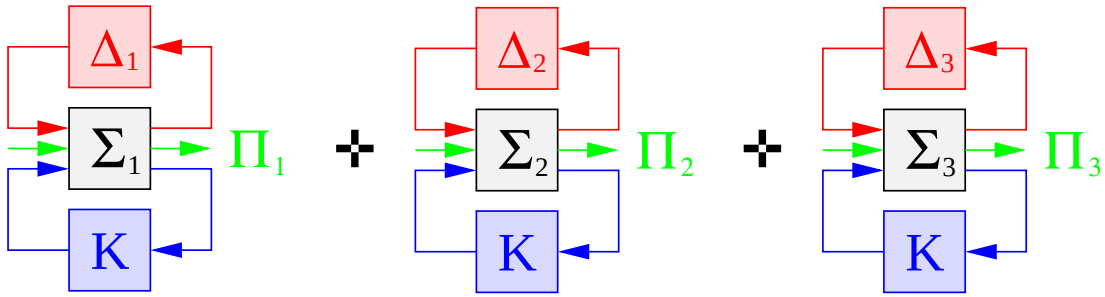
- 4- Eliminate Δ with previously explained methodology

$$(A^{[v]}X + B^{[v]}S) + (A^{[v]}X + B^{[v]}S)^T < 0$$

- 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



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

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

$$K = SX^{-1}$$

● 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.

❹ 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]

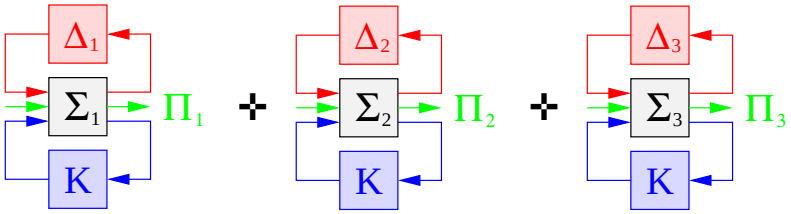
- Other control design problems ?
- Not LMI, or still some unknown changes of variables to be found
- Many (non LMI) results for static-output feedback $u = Ky$

  (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]

❶ Robust multi-objective problems



❷ Linear Matrix Inequalities & Optimization tools $A^T P + PA < 0 \leftrightarrow \mathsf{A}\mathsf{x} = \mathsf{b}\,,\ \mathsf{x} \in \mathcal{K}$

❸ Manipulating inequalities to obtain LMI results

$$\left\{ \begin{array}{l} A > BC^{-1}B^T \\ C > 0 \end{array} \right. \Leftrightarrow \left[\begin{array}{cc} A & B \\ B^T & C \end{array} \right] > 0$$

❹ Some LMI results

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

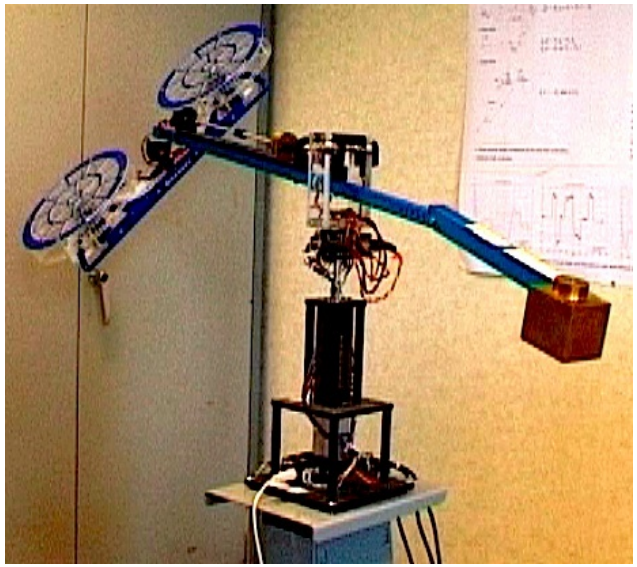
$$\Updownarrow$$

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

❺ Solving robust multi-objective problems in RoMulOC

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

- ”Helicopter“ example



- System defined at maximal value of parameters

```
>> 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 )
```

- System defined at maximal value of parameters

```
>> 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

```
>> 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

```
>> 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

```
>> 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 )
```

❺ Solving robust multi-objective problems in RoMulOC

- Declare a state-feedback design problem

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

- Add an H_∞ performance objective

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

- Add a pole location performance objective

```
>> 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

```
>> 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

```
>> quiz
control problem: STATE-FEEDBACK design
Lyapunov function: UNIQUE (quadratic stability)
Specified performances / systems:
# Hinfty < 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'
```

- Constrained by LMIs

```
>> quiz.lmi
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|     ID|       Constraint|                          Type|        Tag|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|     #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|
```

```
|   #18|   Numeric value|   Matrix inequality 3x3|   Constraint 1|
|   #19|   Numeric value|   Matrix inequality 4x4|   Constraint 2|
|   #20|   Numeric value|   Matrix inequality 3x3|   Constraint 3|
|   #21|   Numeric value|        Element-wise 1x1|   Constraint 4|
|   #22|   Numeric value|   Matrix inequality 3x3|   Constraint 1|
|   #23|   Numeric value|   Matrix inequality 4x4|   Constraint 2|
|   #24|   Numeric value|   Matrix inequality 3x3|   Constraint 3|
|   #25|   Numeric value|        Element-wise 1x1|   Constraint 4|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

- And can be solved (SeDuMi solver by default)

```
>> K = solvesdp( quiz )
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
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
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            6.96E+01 0.000
  1 :  -1.79E+02 1.85E+01 0.000 0.2657 0.9000 0.9000  -0.09  1  1  5.0E+02
  2 :  -1.05E+02 5.96E+00 0.000 0.3223 0.9000 0.9000   1.55  1  1  1.1E+02
  3 :  -2.56E+01 1.38E+00 0.000 0.2312 0.9000 0.9000   1.73  1  1  1.9E+01
  4 :  -5.54E+00 2.62E-01 0.000 0.1902 0.9000 0.9000   1.21  1  1  3.2E+00
  5 :  -1.84E+00 8.00E-02 0.000 0.3050 0.9000 0.9000   1.29  1  1  8.3E-01
  6 :  -7.08E-01 2.90E-02 0.000 0.3621 0.9000 0.9000   1.35  1  1  2.6E-01
  7 :  -2.95E-01 1.05E-02 0.000 0.3637 0.9000 0.9000   1.27  1  1  8.3E-02
  8 :  -2.30E-01 3.57E-03 0.000 0.3393 0.9000 0.9000   1.12  1  1  2.7E-02
  9 :  -1.97E-01 6.73E-04 0.000 0.1882 0.9000 0.9000   1.00  1  1  5.1E-03
 10 :  -1.91E-01 2.02E-05 0.000 0.0300 0.9900 0.9900   0.98  1  1  1.6E-04
 11 :  -1.91E-01 1.13E-06 0.000 0.0558 0.9900 0.9900   1.00  1  1  8.7E-06
 12 :  -1.91E-01 3.08E-07 0.000 0.2737 0.9000 0.9000   1.00  1  1  2.4E-06
 13 :  -1.91E-01 1.33E-08 0.000 0.0433 0.9900 0.9900   1.00  1  1  1.0E-07
 14 :  -1.91E-01 3.01E-09 0.000 0.2261 0.9000 0.9000   1.00  2  2  2.3E-08
```

```
 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*x               b*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)
   Pre          IPM          Post
1.800E-01    4.000E-01    7.000E-02
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(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

- Robustnes: 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