Robust Multi-Objective Control for Linear Systems

Elements of theory and ROMULOC toolbox

Dimitri PEAUCELLE & Denis ARZELIER

LAAS-CNRS, Toulouse, FRANCE

with the participation of Master's Students Maud SEVIN & Alberto BORTOTT

General Features of RoMulOC

→ Matlab Toolbox - freely distributed `www/laas.fr/OLOCEP/romuloc`

I - Modeling features of uncertain LTI systems

→ State-space systems with performance input/output channel

→ Both polytopic and LFT uncertain systems - large variety of uncertainty models

→ Basic model manipulations

II - Robust performances in Lyapunov framework

→ Robust control objectives: stability, transient response (pole location),
perturbation rejection (H_∞, H_2 and impulse-to-peak)

→ All performances are recast in a Lyapunov framework

→ Robustness is achieved with either Unique Lyapunov function or PDLF

→ LMI results derived using Quadratic Separation and Slack Variables

III - LMIs and convex polynomial-time optimization

→ Semi-Definite Programming and LMIs (black box)

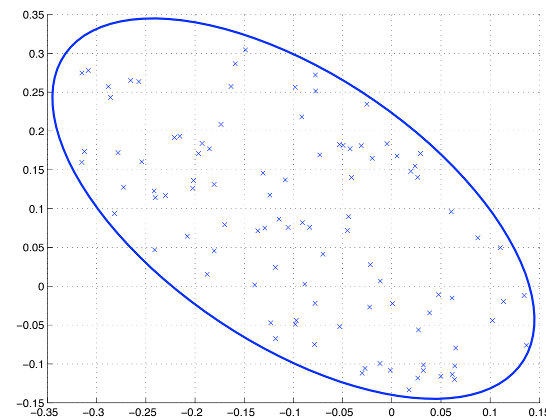
→ SDP solvers and YALMIP parser (user can tune solvers)

6-th order mechanical system ($x \in \mathrm{R}^3$)

$$\ddot{x}(t) + M^{-1}(\Delta_M)D(\Delta_D)\dot{x}(t) + M^{-1}(\Delta_M)Kx(t) = M^{-1}(\Delta_M)E(\Delta_E)w(t)$$
$$z(t) = C(\Delta_C)x(t) + Fw(t)$$

where

$M(\Delta_M) = M_0 + M_1\Delta_M M_2$ with $\Delta_M \in \mathrm{R}^{1\times 2}$ in ellipsoid



$D(\Delta_D) = D_0 + D_1\Delta_D D_2$ with $\Delta_D \in \mathrm{R}^{2\times 2}$ norm-bounded $\Delta_D^T\Delta_D \leq 0.25^2\mathrm{I}$

$E(\Delta_E) = E_0 + E_1\Delta_E E_2$ with $\Delta_E \in [-0.25\ 0.25]$ scalar in interval

$C(\Delta_C) = C_0 + C_1\Delta_C C_2$ with $\Delta_C \in \mathrm{R}^{2\times 2}$ in polytope co $\left\{\Delta_C^{[1]}, \Delta_C^{[2]}, \Delta_C^{[3]}\right\}$

Robust analysis

→ Robust pole location in a sector (robust bound on damping of all modes)

→ Robust H_∞ norm of $w \to z$ transfer : $\|T_{z/w}(s,\Delta)\|_\infty \leq \gamma$.

Demo example - solved with RoMulOC

```
>> sys=ssmodel('mechanical system');
>> sys.A = [ zeros(n) , eye(n) ; -iM0*D0 , -iM0*K ]; ...
>> sys.Bw = [ zeros(n) ; iM0*E0 ];
>> Dm = udiss( X, Y, Z, 'Inertia');
>> Dd = unb( 2, 2, 0.25, 'Damping');
>> De = uinter(-0.25, 0.25, 'Input');
>> Dc = upoly( Dcv, 'Output');
>> usys = ussmodel( sys, diag(Dm, Dd, De, Dc) );
>> r1 = region( 'plane', 0, asin(0.35) );
>> pb1 = dstability( usys, r1 );
>> pb1 = pb1 + ctrpb( 'analysis', 'Lyap unique' );
>> IsDstable = solvesdp( pb1 );
>> pb2 = hinfty( pb2, usinf );
>> pb2unique = pb2 + ctrpb( 'analysis', 'Lyap unique' );
>> HinfLyapUnique = solvesdp( pb2unique );
>> pb2PDLF = pb2 + ctrpb( 'analysis' , 'PDLF' );
>> HinfPDLF = solvesdp( pb2PDLF );
```

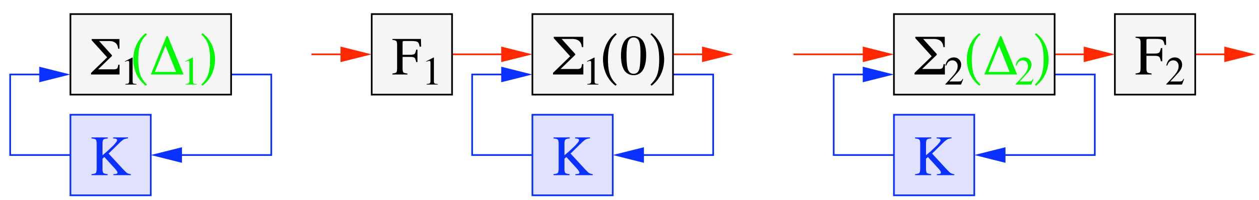
General Robust Multi-Objective Control Problem

Δ : errors in modeling, operating conditions, mass-production...

Δ : uncertainty belongs to a set $\mathbf{\Delta}$.

$$
\begin{aligned}
sx(t) &= A(\Delta)x(t) + B_w(\Delta)w(t) + B_u(\Delta)u(t) \\
z(t) &= C_z(\Delta)x(t) + D_{zw}(\Delta)w(t) + D_{zu}(\Delta)u(t) \\
y(t) &= C_y(\Delta)x(t) + D_{yw}(\Delta)w(t) + D_{yu}(\Delta)u(t)
\end{aligned}
$$

Find controller K that fulfills robust specifications Π_i defined for models $\Sigma_i(\Delta_i)$ with $\Delta_i \in \mathbf{\Delta}_i$.



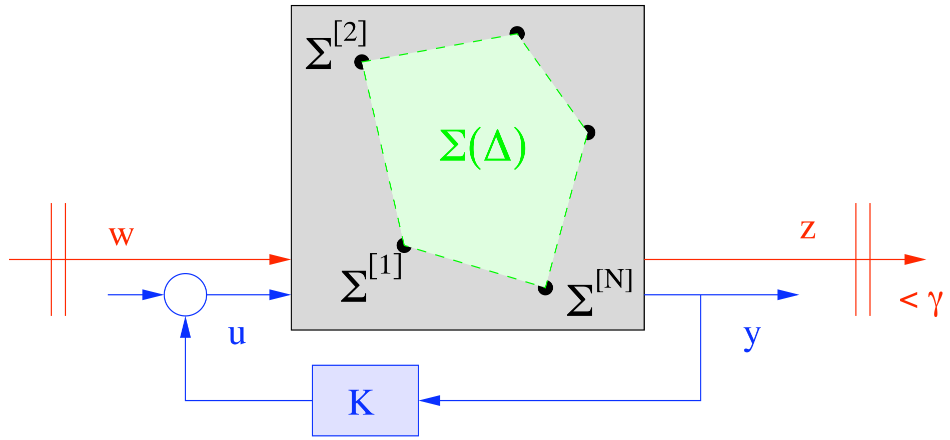
RoMulOC today

→ Modeling tools ready for the global design problem (uncertainties restricted to be constant)

→ Analysis : Unique Lyapunov function and a PDLF method

→ Control design : only for unique Lyapunov function

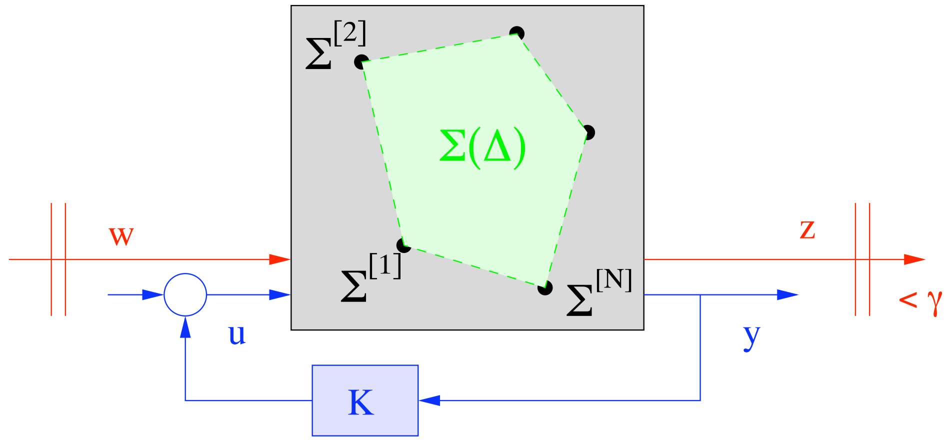
→ Polytopic models



✪ Affine polytopic models : convex hull of N vertices

$$A(\Delta) = \sum \zeta_i A^{[i]} \ , \ B_w(\Delta) = \sum \zeta_i B_w^{[i]} \ \ldots \qquad : \ \zeta_i \geq 0 \ , \ \sum \zeta_i = 1$$

→ Polytopic models



✪ Affine polytopic models : convex hull of N vertices

$$A(\Delta) = \sum \zeta_i A^{[i]} \;,\; B_w(\Delta) = \sum \zeta_i B_w^{[i]} \;\ldots \qquad : \; \zeta_i \geq 0 \;,\; \sum \zeta_i = 1$$

➥ Parallelotopic models with N_P axes

$$A(\Delta) = A^{[0]} + \sum \xi_i A^{[i]} \;,\; B_w(\Delta) = B_w^{[0]} + \sum \xi_i B_w^{[i]} \;\ldots \qquad : \; |\xi_i| \leq 1$$

⇛ polytope with $N = 2^{N_P}$ vertices

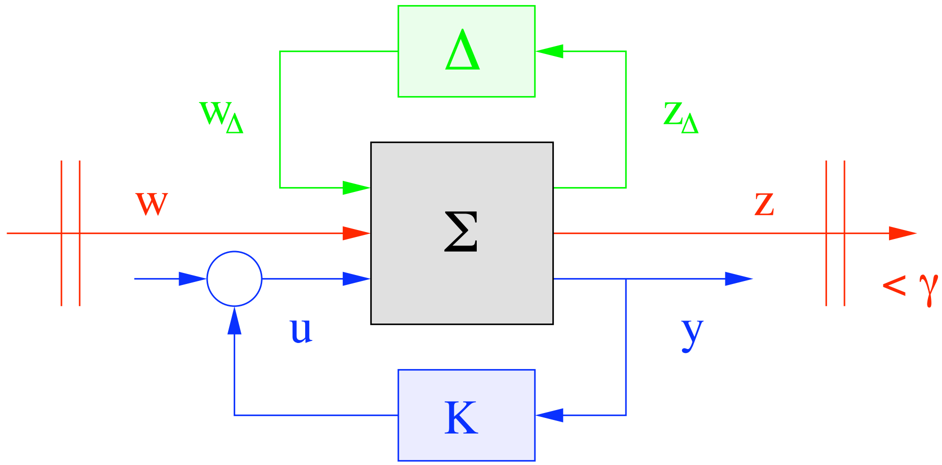
➥ Interval models with N_I non equal coefficients

$$A^{[1]} \preceq A(\Delta) \preceq A^{[2]} \;:\; a_{ij}^{[1]} \leq a_{ij}(\Delta) \leq a_{ij}^{[2]}$$

⇛ parallelotope with axes in the euclidian basis of matrices

⇛ polytope with $N = 2^{N_I}$ vertices

→ LFT models



$$
\begin{aligned}
sx(t) &= Ax(t) + B_\Delta w_\Delta(t) + B_w w(t) + B_u u(t) \\
z_\Delta(t) &= C_\Delta x(t) + D_{\Delta\Delta} w_\Delta(t) + D_{\Delta w} w(t) + D_{\Delta u} u(t) \\
z(t) &= C_z x(t) + D_{z\Delta} w_\Delta(t) + D_{zw} w(t) + D_{zu} u(t) \\
y(t) &= C_y x(t) + D_{y\Delta} w_\Delta(t) + D_{yw} w(t) + D_{yu} u(t)
\end{aligned}
\quad : \quad
\begin{aligned}
w_\Delta &\in \mathbb{C}^{q_\Delta} \\
z_\Delta &\in \mathbb{C}^{p_\Delta}
\end{aligned}
$$

Linear - Fractional Transformation:

$$A(\Delta) = A + B_\Delta \Delta (\mathrm{I} - D_{\Delta\Delta}\Delta)^{-1} C_\Delta \ , \ B_w(\Delta) = B_w + B_\Delta \Delta (\mathrm{I} - D_{\Delta\Delta}\Delta)^{-1} D_{\Delta w} \ \ldots$$

Any model rational in δ_i parameters $\Rightarrow$ LFT (not unique) with diagonal Δ =diag$(\delta_1, \delta_1, ..., \delta_2, ...)$.

➞ Uncertainty sets

✪ $\{X, Y, Z\}-$dissipative matrices

$$\left\{ \ \Delta \in \mathbf{C}^{q_w \times p_z} \ : \ X + Y\Delta + \Delta^* Y^* + \Delta^* Z \Delta \leq 0 \ , \ X \leq 0 \ , \ Z \geq 0 \ \right\}$$

➥ Norm-bounded uncertainties : $\|\Delta\| \leq \rho 1$ $\Rrightarrow$ $\{-\rho^2 \mathbf{I}, 0, \mathbf{I}\}-$dissipative

➥ Positive real uncertainties : $\Delta + \Delta^* \geq 0$ (eg. s) $\Rrightarrow$ $\{0, -\mathbf{I}, 0\}-$dissipative

→ **Uncertainty sets**

✪ $\{X, Y, Z\}-$dissipative matrices

$$\left\{ \ \Delta \in \mathbf{C}^{q_w \times p_z} \ : \ X + Y\Delta + \Delta^* Y^* + \Delta^* Z \Delta \leq \mathbf{O} \ , \ X \leq \mathbf{O} \ , \ Z \geq \mathbf{O} \ \right\}$$

➥ Norm-bounded uncertainties : $\|\Delta\| \leq \rho 1$ $\quad \Rrightarrow \quad \{-\rho^2 \mathbf{I}, \mathbf{O}, \mathbf{I}\}-$dissipative

➥ Positive real uncertainties : $\Delta + \Delta^* \geq \mathbf{O}$ (eg. s) $\quad \Rrightarrow \quad \{\mathbf{O}, -\mathbf{I}, \mathbf{O}\}-$dissipative

✪ Polytopic uncertainties

polytope N vertices

$$\left\{ \ \Delta = \sum \zeta_i \Delta^{[i]} \ : \ \zeta_i \geq 0 \ , \ \sum \zeta_i = 1 \ \right\}$$

➥ Parallelotopic uncertainties

N_P axes $\Rrightarrow$ polytope $N = 2^{N_P}$

$$\left\{ \ \Delta = \Delta^{[0]} + \sum \xi_i \Delta^{[i]} \ : \ |\xi_i| \leq 1 \ \right\}$$

➥ Interval uncertainties

N_I coef. $\neq$ $\Rrightarrow$ polytope $N = 2^{N_I}$

$$\left\{ \ \Delta^{[1]} \preceq \Delta \preceq \Delta^{[2]} \ : \ \delta_{ij}^{[1]} \leq \delta_{ij} \leq \delta_{ij}^{[2]} \ \right\}$$

Demo example - solved with RoMulOC

```
>> sys=ssmodel('mechanical system');
>> sys.A = [ zeros(n) , eye(n) ; -iM0*D0 , -iM0*K ]; ...
>> sys.Bw = [ zeros(n) ; iM0*E0 ];
>> Dm = udiss( X, Y, Z, 'Inertia');
>> Dd = unb( 2, 2, 0.25, 'Damping');
>> De = uinter(-0.25, 0.25, 'Input');
>> Dc = upoly( Dcv, 'Output');
>> usys = ussmodel( sys, diag(Dm, Dd, De, Dc) );
>> r1 = region( 'plane', 0, asin(0.35) );
>> pb1 = dstability( usys, r1 );
>> pb1 = pb1 + ctrpb( 'analysis', 'Lyap unique' );
>> IsDstable = solvesdp( pb1 );
>> pb2 = hinfty( pb2, usinf );
>> pb2unique = pb2 + ctrpb( 'analysis', 'Lyap unique' );
>> HinfLyapUnique = solvesdp( pb2unique );
>> pb2PDLF = pb2 + ctrpb( 'analysis' , 'PDLF' );
>> HinfPDLF = solvesdp( pb2PDLF );
```

Nominal performance analysis $V(x) = x^T P x$ Lyapunov function ($P > 0$)

- **Stability** $A^T P + PA < 0 \quad | \quad A^T PA - P < 0$

- **D-Stability**
$$\begin{bmatrix} I & A^* \end{bmatrix} \begin{bmatrix} r_{11}P & r_{12}P \\ r_{12}^*P & r_{22}P \end{bmatrix} \begin{bmatrix} I \\ A \end{bmatrix} < 0$$

- **H_∞ norm**
$$\begin{bmatrix} A^T P + PA + C_z^T C_z & PB_w + C_z^T D_{zw} \\ B_w^T P + D_{zw}^T C_z & -\gamma^2 I + D_{zw}^T D_{zw} \end{bmatrix} < 0$$

- **H_2 norm**
$$A^T P + PA + C_z^T C_z < 0$$
$$\text{trace}(B_w^T P B_w) < \gamma^2$$

- **Impulsion-to-peak**
$$A^T P + PA < 0 \quad B_w^T P B_w < \gamma^2 I$$
$$C_z^T C_z < P \quad D_{zw}^T D_{zw} < \gamma^2 I$$

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

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

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

Test over sample values in $\mathbf{\Delta}$ gives optimistic results.

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

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

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

Test over sample values in $\mathbf{\Delta}$ gives optimistic results.

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

(and, conservative, techniques to have finite nb of LMIs)

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

➙ Polytopic PDLF: $P(\Delta) = \sum \zeta_i P^{[i]}$ (and ”Slack Variables” technique)

➙ Quadratic-LFT PDLF: (and Quadratic Separation)

$$P(\Delta) = \begin{bmatrix} \mathbf{I} & \Delta_C^T \end{bmatrix} P \begin{bmatrix} \mathbf{I} \\ \Delta_C \end{bmatrix} ,\quad \Delta_C = (\mathbf{I} - \Delta D_{\Delta\Delta})^{-1} \Delta C_\Delta$$

```
>> sys=ssmodel('mechanical system');
>> sys.A = [ zeros(n) , eye(n) ; -iM0*D0 , -iM0*K ]; ...
>> sys.Bw = [ zeros(n) ; iM0*E0 ];
>> Dm = udiss( X, Y, Z, 'Inertia');
>> Dd = unb( 2, 2, 0.25, 'Damping');
>> De = uinter(-0.25, 0.25, 'Input');
>> Dc = upoly( Dcv, 'Output');
>> usys = ussmodel( sys, diag(Dm, Dd, De, Dc) );
>> r1 = region( 'plane', 0, asin(0.35) );
>> pb1 = dstability( usys, r1 );
>> pb1 = pb1 + ctrpb( 'analysis', 'Lyap unique' );
>> IsDstable = solvesdp( pb1 );
>> pb2 = hinfty( pb2, usinf );
>> pb2unique = pb2 + ctrpb( 'analysis', 'Lyap unique' );
>> HinfLyapUnique = solvesdp( pb2unique );
>> pb2PDLF = pb2 + ctrpb( 'analysis' , 'PDLF' );
>> HinfPDLF = solvesdp( pb2PDLF );
```

Semi-Definite Programming and LMIs

→ The LMI problem is defined in YALMIP format

→ Can be solved with any available SDP software

```
result = solvesdp( problem , sdpsettings('solver', 'sdpt3'))
```

→ `result` is the answer to the `problem`, e.g. the guaranteed H_∞ norm

→ Constraints and variables are also available

```
problem.vars                    problem.lmi
```

Results for the demo example with SeDuMi solver

	Nominal	200 rand. val.	2000 rand. val.	Robust	PDLF	Unique Lyapunov
H_∞	2	2.484	2.4957	??	2.516	2.866
time	0.036s	13.79s	2 min	NP	17.99s	6.772s

Conclusions and Prospective work

Tests and users feedback

→ Test yourself : `www.laas.fr/OLOCEP/romuloc/`

→ Tests done at LAAS with fitted to application improvements

(clearance of satellite attitude & flight control laws)

Future versions

→ PDLF State-feedback design - soon some first results

→ Output-feedback : full-order (LMI) and SOF (BMI) - not planned

→ Uncertain Time-Delay Systems (F. Gouaisbaut) - possible

→ Discrete-time uncertain Periodic Systems (C. Farges) - not planned

→ Time varying uncertainties - work in progress

→ Uncertain Descriptor Systems - work in progress

Uncertain descriptor systems: a method for more PDLF, less conservative, results

→ PDLF stability analysis of $\dot{x} = A(\Delta)x$ with $V(x,\Delta) = x^T P(\Delta) x$ where

$$P(\Delta) = \begin{bmatrix} I & A^T(\Delta) \end{bmatrix} P \begin{bmatrix} I \\ A(\Delta) \end{bmatrix}$$

→ is equivalent to "Lyapunov unique" stability analysis with $V(x,\dot{x}) = \begin{pmatrix} x & \dot{x} \end{pmatrix} P \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$

of the augmented system, in descriptor form:

$$\begin{bmatrix} I & 0 \\ 0 & I \\ I & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{bmatrix} A(\Delta) & 0 \\ 0 & A(\Delta) \\ 0 & I \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$$

↗ Reducing conservatism is as simple as augmenting the model with higher order derivatives

↘ Needs to produce and code LMI results for descriptor systems (work in progress)

Uncertain descriptor systems: extensions for time-vaying uncertainties, delays ...

→ For a time-varying uncertainty, model augmentation by derivation produces:

$$w_\Delta = \Delta z_\Delta$$
$$\dot{w}_\Delta = \dot{\Delta} z_\Delta + \Delta \dot{z}_\Delta$$

→ For delays, similar procedure produces new Lyapunov-Krasovskii results:

$$e^{-sh} = 1 - sh + \frac{(sh)^2}{2} + \dots$$
$$\downarrow$$
$$x(t-h) = x(t) - h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + \dots$$

→ Possible extensions for the case when Δ is a non-linear operator ?