

ROMULOC
RObust MULti-Objective Control toolbox

Séminaire OLOCEP - 30 Juin 2004

Plan de la présentation

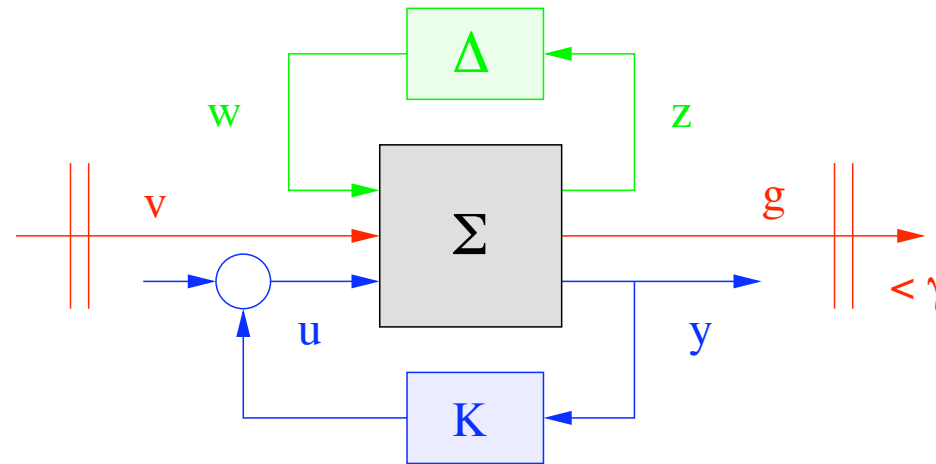
1 – Modélisation : Σ LTI incertains et cahier des charges	2
2 – Résultats LMI d'analyse robuste	16
3 – Résultats LMI et BMI de synthèse	23
4 – Démonstration de ROMULOC 1beta	27

1 – Modélisation : Σ LTI incertains et cahier des charges

- Systèmes LTI incertains, perturbés, contrôlés
- L'objet `ssmodel`, rétroactions et opérateurs en série
- L'objet `uncertainty`, 6 types, incertitudes structurées
- L'objet `ussmodel`, 2 familles, LFT & affines polytopiques

1 - Modélisation : Σ LTI incertains, perturbés, contrôlés³

3 couples Entrées / Sorties



★ $w(t)$: vecteur d'entrées exogènes / $z(t)$ vecteur des sorties exogènes

➔ Transformation Fractionnaire \Rightarrow Linéaire

★ $v(t)$: vecteur d'entrées de perturbation / $g(t)$ vecteur des sorties à commander

➔ Gabarits temporels, fréquentiels, énergie, consommation...

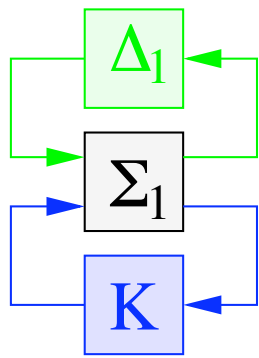
★ $u(t)$: vecteur d'entrées de commande / $y(t)$ vecteur des sorties mesurées

➔ K : paramètres de décision du problème original.

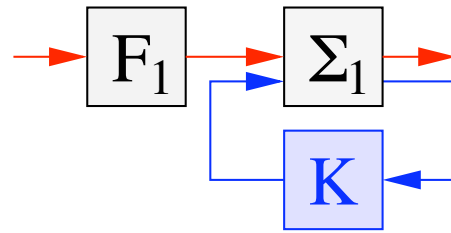
$$\exists K : \forall \Delta, \|\Sigma \star K \star \Delta\| \leq \gamma$$

1 - Modélisation : Σ LTI incertains, perturbés, contrôlés⁴

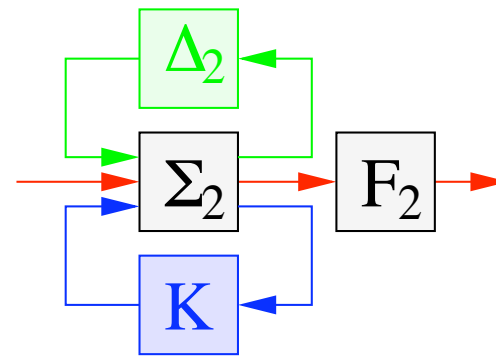
Cahier des charges : 1 spécification = 1 modèle



Rob. Stable



Min Hinf norm



Rob H2 norm

★ Spécifications envisagées (nominales et robustes) :

- ➔ Localisation de pôles : temps de réponse, amortissement, fréquences propres...
- ➔ Coût H_∞ : transfert énergie/énergie, amplification $\forall \omega$, lemme borné réel.
- ➔ Coût H_2 : transfert énergie/b.b.g., valeur max/énergie finie, \sim consommation.
- ➔ Coût I2P : valeur pic/impulsion.

L'objet ssmode1

Modélisation dans l'espace d'état

★ Notations utilisées :

$$\begin{array}{l} x \in \mathbb{R}^n \quad w \in \mathbb{R}^{q_w} \quad v \in \mathbb{R}^{q_v} \quad u \in \mathbb{R}^{q_u} \\ \delta[x(t)] = Ax(t) + B_w w(t) + B_v v(t) + B_u u(t) \\ z \in \mathbb{R}^{p_z} \quad z(t) = C_z x(t) + D_{zw} w(t) + D_{zv} v(t) + D_{zu} u(t) \\ g \in \mathbb{R}^{p_g} \quad g(t) = C_g x(t) + D_{gw} w(t) + D_{gv} v(t) + D_{gu} u(t) \\ y \in \mathbb{R}^{p_y} \quad y(t) = C_y x(t) + D_{yw} w(t) + D_{yv} v(t) + D_{yu} u(t) \end{array}$$

Σ tps discret : $\delta[x(t)] = x(t + T)$

Σ tps continu : $\delta[x(t)] = \dot{x}(t)$ (i.e. $T = 0$)

★ Objet Matlab/Scilab unique : `ssmodel`

Variante des objets `lti/ss` de la Control System Toolbox.

```
sys = ssmodel
```

```
sys.A=A
```

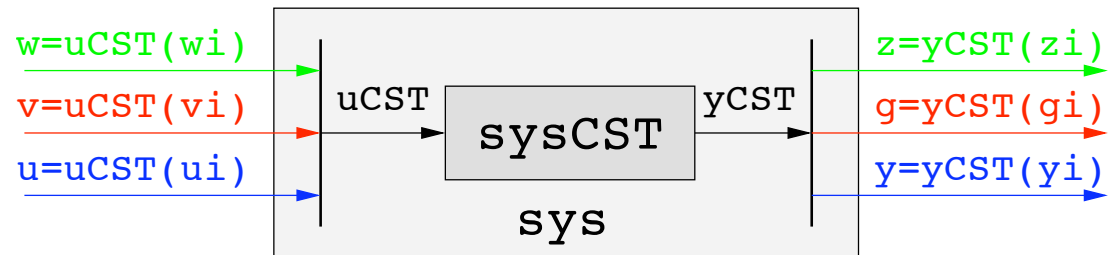
```
sys.Dgu=Dgu
```

```
sys.T=0
```

★ Compatibilité avec Toolbox existantes

↳ Control System Toolbox

```
sys = ssmode1( sysCST , wi,vi,ui, zi,gi,yi )
```



↳ LFR Toolbox de J.F. Magni, ONERA (en cours)

★ Lecture des données :

```
sys.A      sys.Dzy
```

```
sys.T      sys.n      sys.qw
```

★ Fonctionnalités usuelles des systèmes LTI :

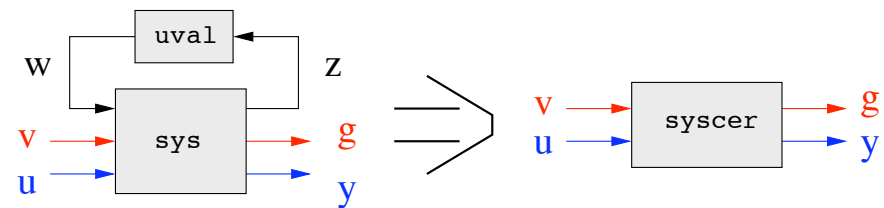
norm, sigma ... (transfert de performance)

pole, transpose...

Interconnexions élémentaires

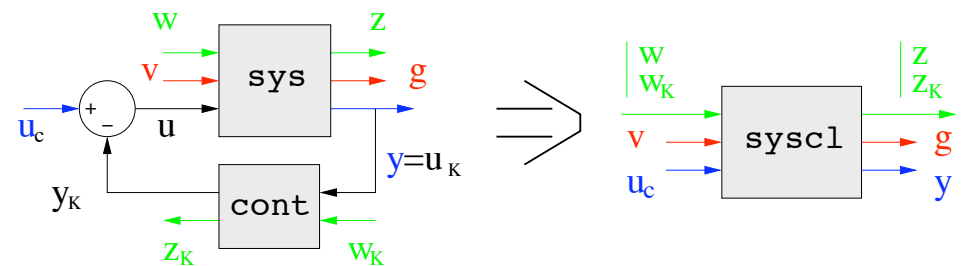
- ★ Figurer les incertitudes :

```
syscer = certain( sys, uval );
```



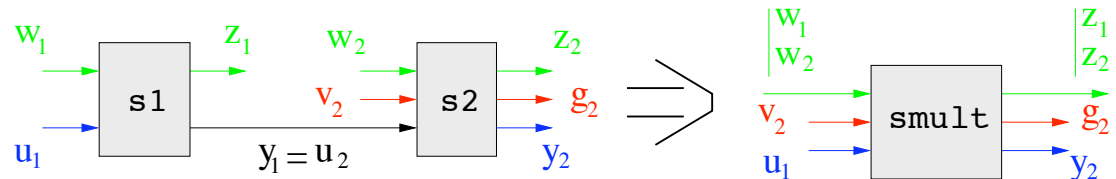
- ★ Appliquer une loi de commande :

```
syscl = feedback( sys, cont );
```



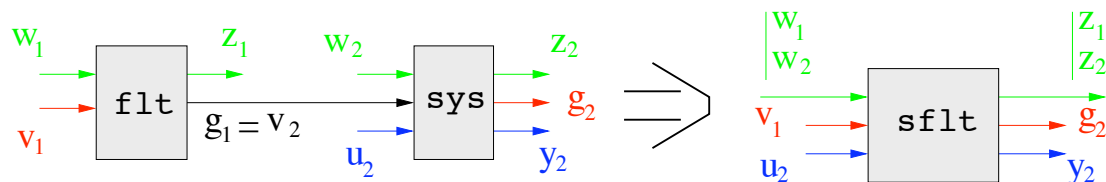
- ★ Deux systèmes en série :

```
smult = s2*s1;
```



- ★ Gabarits de performance :

```
sflt = shape( sys , flt );
```



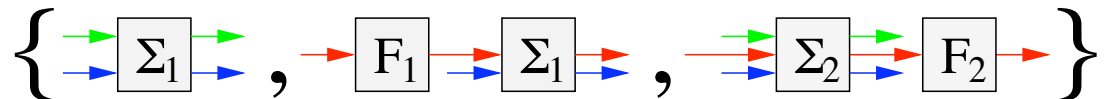
Listes de modèles

- ★ Liste de modèles quelconques (cell array):

```
lsys = { s1, s2 , s3 }
```

```
lsys{1} = s1
```

```
lsys{1}.A = s1.A
```



- ★ Vecteur de modèles de mêmes dimensions (compatible CST).

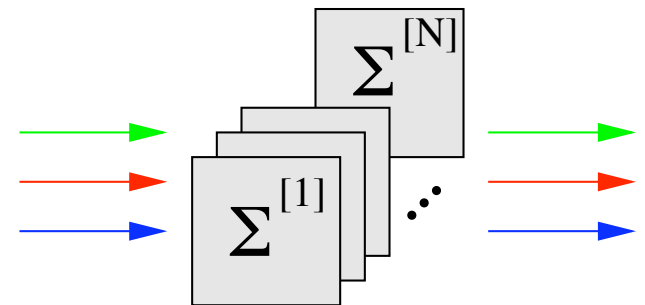
```
asys = ssmode1 ( sysCST , wi,zi,ui, zi,gi,yi )
```

```
asys(1) = s1
```

```
asys(2).A = s2.A
```

```
asys.A(:, :, 5) = s5.A
```

```
asys.T          asys.n          asys.py
```



L'objet uncertainty

Opérateurs incertains : matrices constantes réelles

★ Opérateurs $\{X, Y, Z\}$ -dissipatifs

$$\left\{ \Delta \in \mathbb{R}^{q_w \times p_z} : X + Y\Delta + \Delta^T Y^T + \Delta^T Z\Delta \leq 0, X \leq 0, Z \geq 0 \right\}$$

↳ Opérateurs de norme bornée : $\|\Delta\| \leq \rho \mathbb{1} \rightarrow \{-\rho^2 \mathbb{1}, 0, \mathbb{1}\}$ -dissipatifs

↳ Opérateurs passifs : $\Delta + \Delta^T \geq 0 \rightarrow \{0, -\mathbb{1}, 0\}$ -dissipatifs

★ Opérateurs polytopiques

polytope N sommets

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

↳ Opérateurs parallélotopiques

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

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

↳ Opérateurs intervalles

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

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

Objet Matlab/Scilab unique : uncertainty

```
u1 = udiss( X, Y, Z )
```

```
u4 = upoly( Vertices )
```

Vertices : 3-D array

```
u2 = unb( rho , qw , pz )
```

```
u5 = uparal( Center, Axes )
```

Axes : 3-D array

```
u3 = upos( qw )
```

```
u6 = uinter( Dmin, Dmax )
```

Incertitudes block-diagonales, répétées

$$\Delta = \begin{bmatrix} \Delta_1 & 0 & 0 & 0 & 0 \\ 0 & \Delta_1 & 0 & 0 & 0 \\ 0 & 0 & \Delta_2 & 0 & 0 \\ 0 & 0 & 0 & \Delta_5 & 0 \\ 0 & 0 & 0 & 0 & \Delta_2 \end{bmatrix}$$

```
>> U = diag( u1, u1, u2, u5, u2 )
```

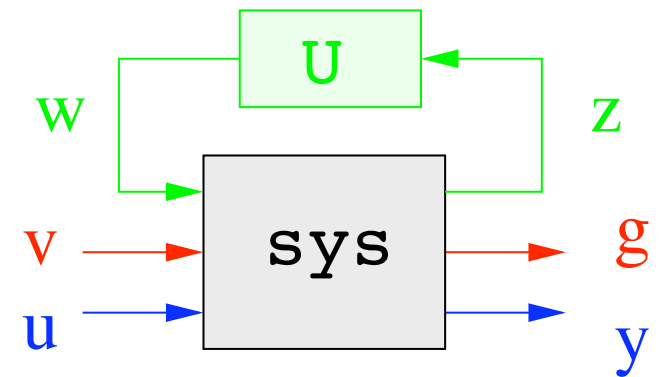
L'objet ussmode1

Objet Matlab/Scilab unique : `ussmodel`

★ Systèmes incertains LFT

```
usys = ussmodel ( sys, U )
```

Toutes les fonctionnalités de `ssmodel`
s'appliquent à `ussmodel`



★ Modèle rationnel / paramètres incertains

$$A(\Delta) = A + B_w \Delta (\mathbb{1} - D_{zw} \Delta)^{-1} C_z \quad , \quad B_v(\Delta) = B_v + B_w \Delta (\mathbb{1} - D_{zw} \Delta)^{-1} D_{zv} \quad \dots$$

➔ Si $D_{zw} = \mathbb{0}$: modèle incertain affine

★ Systèmes affines polytopiques, N sommets

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

↳ Systèmes parallélotopiques, N_P axes \rightarrow polytope $N = 2^{N_P}$ sommets

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

↳ Systèmes intervalles, N_I coef. \neq \rightarrow polytope $N = 2^{N_I}$ sommets

$$A^{[1]} \preceq A(\Delta) \preceq A^{[2]} , \quad B_w^{[1]} \preceq B_w(\Delta) \preceq B_w^{[2]} \quad \dots$$

★ Objet ussmodel

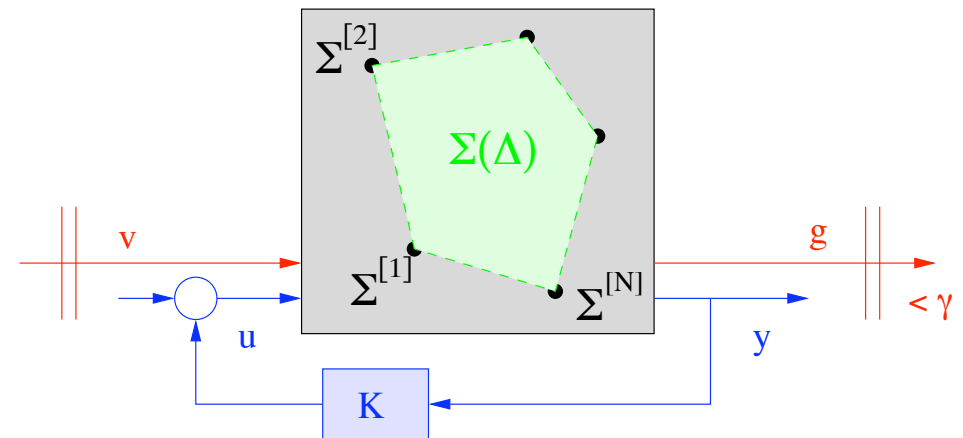
```
usys = upoly( SVertices )
```

```
SVertices : array of ssmodel
```

```
usys = uparal( SCenter, SAXes )
```

```
SAXes : array of ssmodel
```

```
usys = uinter( Smin, Smax )
```



2 – Résultats LMI d'analyse robuste

- Objectifs et outils
- Critères SDP représentables
- Fonctionnalités de ROMULOC

- ✓ Objectifs d'analyse en stabilité et performance
 - Critères SDP représentables
 - Relaxations SDP de conditions NP-difficiles
- ✓ Outils théoriques de relaxation
 - Théorie de Lyapunov
 - Lemme de projection, S-procédure, lemme de Finsler
 - Séparation quadratique
- ✓ Outils numériques
 - Programmation semidéfinie positive
 - Méthodes de linéarisation (gradient conditionnel, projections alternées)

★ Stabilité

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

★ D-Stabilité

$$\begin{bmatrix} 1 & s^* \end{bmatrix} D \begin{bmatrix} 1 \\ s \end{bmatrix} \leq 0$$

$$\Leftrightarrow \begin{bmatrix} \mathbb{1} & \mathbb{1} \otimes A^T \end{bmatrix} D \otimes P \begin{bmatrix} \mathbb{1} \\ \mathbb{1} \otimes A \end{bmatrix} < 0$$

★ Norme H_∞

$$\begin{bmatrix} A^T P + PA + C_g^T C_g & PB_v + C_g^T D_{gv} \\ B_v^T P + D_{gv}^T C_g & -\gamma^2 \mathbf{1} + D_{gv}^T D_{gv} \end{bmatrix} < 0$$

★ Norme H_2

$$A^T P + PA + C_g^T C_g < 0 \quad \text{trace}(B_v^T P B_v) < \gamma^2$$

★ Crête/impulsion

$$A^T P + PA < 0 \quad B_v^T P B_v < \gamma^2 \mathbb{1} \quad C_g^T C_g < P$$

★ Analyse nominale (LMI) → Analyse robuste (NP-dur)

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

➔ Nombre fini de variables - paramétrisation de $P(\Delta)$:

➔ “Stabilité quadratique” : $P(\Delta) = P$

➔ FLDP polytopique : $P(\Delta) = \sum \lambda_i P^{[i]}$

➔ FLDP Q-LFT : $V(\Delta, x) = \begin{pmatrix} x^T & w^T \end{pmatrix} P \begin{pmatrix} x \\ w \end{pmatrix} : w = \Delta C_z x + \Delta D_{zw} w$

➔ $P(\Delta)$ polynômiale en les coefficients de Δ

➔ Nombre fini de contraintes : $\rightarrow \exists P, D, G... : \mathcal{L}_{\Sigma}(P, D, G...) < \mathbb{0}$

➔ Sprocédure, séparation quadratique, DG-scaling, P-separator

➔ Multi-convexité, somme de carrés, slack variables, lemme de projection...

Déterminer la stabilité de $\dot{x} = Ax$ avec $A \in \mathcal{A} = \left\{ A = \sum_{i=1}^N \lambda_i A^{[i]}, \lambda_i \geq 0, \sum_{i=1}^N \lambda_i = 1 \right\}$

→ Paramétrisation polytopique de la FLDP : $P(\lambda) = \sum_{i=1}^N \lambda_i P_i$

→ LMIs paramétrées à résoudre :

$$A^T(\lambda)P(\lambda) + P(\lambda)A(\lambda) < 0, \quad P(\lambda) > 0 \quad \forall \lambda \in \Lambda$$
$$\Leftrightarrow \exists x : F(x, \lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & -A'(\lambda)P(\lambda) - P(\lambda)A(\lambda) \end{bmatrix} > 0 \quad \forall \lambda \in \Lambda$$

→ Une condition suffisante :

$$\exists x \quad F(x, \lambda) > \begin{bmatrix} 0 & 0 \\ 0 & h(\lambda)\mathbb{1} \end{bmatrix} \quad h(\lambda) \geq 0 \quad \forall \lambda \in \Lambda \quad \text{et} \quad h(\lambda) \in \mathbb{R}[\lambda_1, \dots, \lambda_N]$$

★ 5 fonctions pour l'utilisateur

Stabilité: `quiz = stability(usys, method)`

Localisation des pôles: `quiz = dstability(usys, method, region)`

Norme H_∞ : `quiz = hinf(usys, method , value)`

Norme H_2 : `quiz = htwo(usys, method, value)`

Valeur crête/impulsion: `quiz = i2peak(usys, method, value)`

★ Résolution YALMIP + choix de solveur et des paramètres du solveur :

`result = solvesdp(quiz, sdpoptions)`

★ Bibliothèque de 30 fonctions à développer en coopération

↳ Normes simples de programmation

quiz est un objet contenant :

- `quiz.vars` : cell array contenant les variables `sdpvar` (YALMIP)
- `quiz.lmi` : objet `lmi` (YALMIP)
- `quiz.obj` : objet `sdpvar` (YALMIP) définissant l'objectif à minimiser

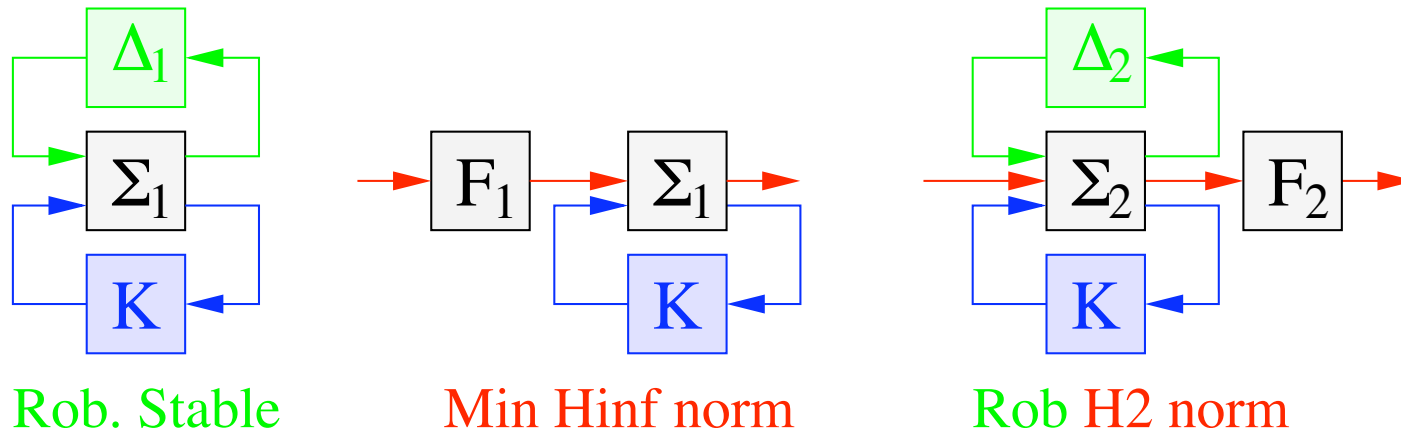
↳ Codification des fonctions

↳ Ajout aisé de méthodes

⇒ ROMULOC 1, Septembre 2004

3 – Résultats LMI et BMI de synthèse

- Formulation générale, robuste, multi-objectif, résilient...
- Hypothèses pour des représentations SDP
- Résultats non-SDP en cours de développement



★ Analyse robuste (LMI plus ou moins pessimiste) → Synthèse robuste (BMI)

$$\exists P, D, G : \mathcal{L}_\Sigma(P, D, G) < 0 \rightarrow \exists K : \{\exists P, D, G : \mathcal{L}_{\Sigma \star K}(P, D, G) < 0\}$$

Exemple de la synthèse H_2 :

$$A_{bf}^T(K)P + PA_{bf}(K) + C_{gbf}^T(K)C_{gbf}(K) < 0 \quad \text{trace}(B_{vbf}^T(K)PB_{vbf}(K)) < \gamma^2$$

➔ Relaxations LMI exactes : (changement de variables linéarisant)

- ➔ Retour d'état (stabilisation, minimisation H_2, H_∞)
- ➔ Modèles LFT, Δ non-structurée
- ➔ Retour de sortie d'ordre plein ($\text{ordre}(K)=\text{ordre}(\Sigma)$)

➔ Relaxations LMI pessimistes :

- ➔ Synthèse paramétrique robuste (Shaping Paradigm $P_i = P \rightarrow G_i = G$)
- ➔ Synthèse avec contrainte de structure (paramétrisation pessimiste)
- ➔ Synthèse multiobjectifs (Shaping Paradigm $P_i = P \rightarrow G_i = G$)

⇒ ROMULOC 2, printemps 2005

- ★ Synthèse avec structure de contrainte → retour de sortie statique
- ★ Synthèse multiobjectifs
- ★ Synthèse robuste

↳ Formules BMI issues de l'analyse

↳ Technique d'élimination de K

$$\Rightarrow \text{LMI} + \text{contrainte } PQ = \mathbb{1}$$

↳ Synthèse de séparateur quadratique vis-à-vis de K

$$\Rightarrow \text{LMI} + \text{contrainte NL} : X - YZ^{-1}Y^T \leq \mathbb{0}$$

★ Solveurs BMI, décomposition croisée, gradient conditionnel

⇒ ROMULOC 3, printemps 2006

4 – Démonstration de ROMULOC 1beta