

Université Paul Sabatier
Cours M2 ISTR & RODECO
Commande linéaire avancée - Commande Robuste
Travaux dirigés - Modélisation incertaine polytopique
Octobre 2016

La totalité des TD et TP se font en salles I1-I3 avec à disposition le logiciel Matlab. Nous utiliserons la version **R2012b** de Matlab fonctionnant dans l'environnement **Windows**. La version Matlab R2012b de l'UPS donne accès aux fonctionnalités de la **Robust Control toolbox** distribuée par Matlab.

Nous allons également travailler avec les logiciels libres suivants : **YALMIP** interface permettant de coder des problèmes d'optimisation tels que les problèmes LMI ; **R-RoMulOC** développé au LAAS dédié aux résultats LMI de commande robuste. De façon à travailler avec des versions à jour vous êtes invités à vous connecter sur les sites^{1,2} pour télécharger et installer sur votre compte les deux logiciels.

Une fois installés, dans Matlab, cliquez sur **Set Path**, faire **Add with subfolders** et sélectionner les répertoires **yalmip** et **romuloc**. Taper ensuite `>> romuloc(2)` dans Matlab pour vérifier que l'installation fonctionne.

Par défaut les problèmes LMI seront résolus par le solveur disponible dans la 'robust control toolbox' de Matlab. Il est également possible d'utiliser des logiciels libres. L'un d'entre eux est disponible en se connectant sur le lecteur réseau **commun Agamemnon**. Faire **Set Path**, **Add with subfolders** et sélectionner le répertoire **M2ASTR/Robuste/SDPT3** pour permettre l'accès.

Soit le système décrit par l'équation différentielle suivante

$$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2y = u$$

comportant deux paramètres incertains :

$$\zeta \in [0.5, 1] \quad , \quad \omega_n \in [1, 10].$$

1.1 Construire un modèle espace d'état du système : $\dot{x} = A(\zeta, \omega_n)x + Bu$, $y = Cx$.

1.2 Proposer deux modèles affines polytopiques englobant de façon plus ou moins proche le modèle incertain initial.

1.3 Définir dans Matlab les modèles avec les fonctionnalités de RoMulOC en utilisant les fonctions **ssmodel** et **upoly**.

1.4 On souhaite analyser l'erreur faite en incluant le système incertain dans les

représentations polytopiques. On le fait par l'analyse des pôles du système. Pour chaque modèle polytopique dans une figure différente (**figure, hold on**) : tracer dans le plan complexe les pôles (**imag, real**) des sommets du polytope (**pole** et **plot**) ; tracer les régions du plan complexe correspondant au fait que la pulsation propre des pôles est bornée, tout comme l'amortissement (**region, plot**) ; tracer les pôles de réalisations aléatoires des systèmes polytopiques (**usample, pole, plot**). Commenter.

1.5 Peut-on conclure sur la stabilité robuste des modèles incertains ?

1.6 D'après les résultats du cours, un test LMI pour s'assurer que les pôles sont à partie réelle inférieure à α est :

$$P > 1 \quad , \quad A^{[v]T}P + PA^{[v]} < 2\alpha P$$

Contraintes à satisfaire simultanément sur tous les sommets du polytope. Définir le problème LMI pour $\alpha = 10$. Résoudre le problème LMI en utilisant la fonction **solvesdp**. Que peut-on en conclure ? Répéter ce processus pour différentes valeurs de α . Peut-on conclure sur la stabilité robuste de la boucle ouverte ?

1.7 De façon à améliorer la stabilité robuste on réalise une commande en boucle fermée

$$u = 50(y_r - y)$$

A l'aide de la commande **feedback** calculer les modèles incertains en boucle fermée. Refaire les tests ci-dessus pour le modèle en boucle fermée. Peut-on conclure sur la stabilité robuste du modèle incertain bouclé ?

1.8 Reproduire le test ci-dessus en utilisant les fonctions **ctrpb** et **dstability**. La stabilité robuste peut-elle être démontrée avec une fonction de Lyapunov dépendant des paramètres incertains ?

1. projects.laas.fr/OLOCEP/rromuloc/download.php
2. users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.Download

Travaux dirigés - Modélisation incertaine polytopique - Corrigé

Octobre 2016

Exercice 1

1.1 La représentation d'état est choisie dans la base telle que $x = (y \quad \dot{y})^T$:

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}}_{A(\zeta, \omega_n)} x + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B u$$

1.2 L'analyse par intervalle des coefficients de $A(\zeta, \omega_n)$ donne que

$$\begin{aligned} \omega_n^2 &\in [1, 100] \\ 2\zeta\omega_n &\in [1, 20]. \end{aligned}$$

On en déduit le modèle polytopique, représenté par toutes les combinaisons convexes de 4 sommets suivants

$$[\omega_n^2 \quad 2\zeta\omega_n] \in \text{co} \left\{ \begin{array}{l} \begin{bmatrix} 1 & 1 \\ 1 & 20 \end{bmatrix}, \\ \begin{bmatrix} 1 & 20 \\ 100 & 20 \end{bmatrix}, \\ \begin{bmatrix} 100 & 20 \\ 100 & 1 \end{bmatrix} \end{array} \right\}.$$

En traçant $2\zeta\omega_n$ comme une fonction de ω_n^2 on se rend compte que quand ζ et ω_n sont dans les intervalles indiqués alors on peut construire des polytopes plus petits que le rectangle précédent tout en incluant toutes les incertitudes. Un tel polytope est par exemple

$$[\omega_n^2 \quad 2\zeta\omega_n] \in \text{co} \left\{ \begin{array}{l} \begin{bmatrix} 1 & 1 \\ 1 & 20 \end{bmatrix}, \\ \begin{bmatrix} 1 & 20 \\ 100 & 20 \end{bmatrix}, \\ \begin{bmatrix} 100 & 20 \\ 100 & 10 \end{bmatrix} \end{array} \right\}.$$

On en déduit donc :

$$\{A(\zeta, \omega_n), \zeta \in [0.5, 1], \omega_n \in [1, 10]\}$$

$$\subset \text{co} \left\{ \begin{array}{l} \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -1 & -20 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -100 & -20 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -100 & -10 \end{bmatrix} \end{array} \right\} \\ \subset \text{co} \left\{ \begin{array}{l} \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -1 & -20 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -100 & -20 \end{bmatrix}, \\ \begin{bmatrix} 0 & 1 \\ -100 & -1 \end{bmatrix} \end{array} \right\}.$$

1.3 Définition d'un tableau de quatre systèmes de mêmes dimensions constitué des sommets du premier polytope, puis définition du polytope :

```
>> sv1=ssmodel;
>> sv1(1).a=[0 1;-1 -1];
>> sv1(2).a=[0 1;-1 -20];
>> sv1(3).a=[0 1;-100 -20];
>> sv1(4).a=[0 1;-100 -1];
>> sv1.bu=[0;1];
>> sv1.cy=[1 0];
>> usp1=upoly(sv1)
Uncertain model : polytope 4 vertices
----- WITH -----
          n=2      mu=1
n=2 dx =  A*x + Bu*u
py=1 y =  Cy*x
continuous time
```

Définition du second polytope comme modification du premier :

```
>> sv2=sv1;
>> sv2(4).a=[0 1;-100 -10];
>> usp2=upoly(sv2)
Uncertain model : polytope 4 vertices
----- WITH -----
          n=2      mu=1
n=2 dx =  A*x + Bu*u
py=1 y =  Cy*x
continuous time
```

1.4 Le nombre de sommets des polytopes est donné par `usp1.nb`. Les sommets du premier polytope sont donnés par `usp1(1)`, `usp1(2)` etc. Pour tracer les pôles des sommets on peut par exemple faire :

```
figure(1); hold on;
for ii=1:usp1.nb
p=pole(usp1(ii));
plot(real(p),imag(p),'r*');
end
```

Le polynôme caractéristique du système originel est

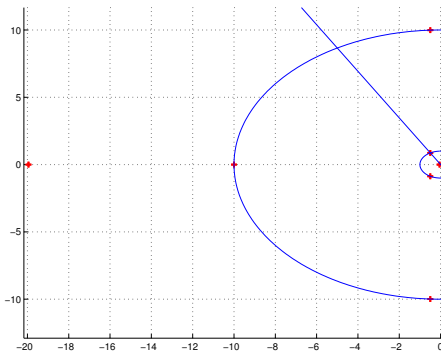
$$s^2 + 2\zeta\omega_n s + \omega_n^2$$

pôles sont de module ω_n et placés sur une droite faisant un angle de $\arcsin(\zeta)$ par rapport à l'axe imaginaire. Comme $\omega_n \in [1, 10]$ les pôles sont tous à l'extérieur du cercle centré en 0 et de

rayon 1 (module des pôles plus grand que 1) et à l'intérieur du cercle centré en 0 et de rayon 10 (module des pôles plus petit que 10). De même les pôles sont en dessous d'une droite faisant un angle de $\arcsin(0.5)$ avec l'axe imaginaire. On peut tracer les limites des régions dans le plan complexe avec les commandes suivantes

```
r1=region('disc',0,1);
plot(r1);
r2=region('disc',0,10);
plot(r2);
r3=region('plane',0,asin(0.5));
plot(r3);
```

Le résultat est donné sur la figure qui suit

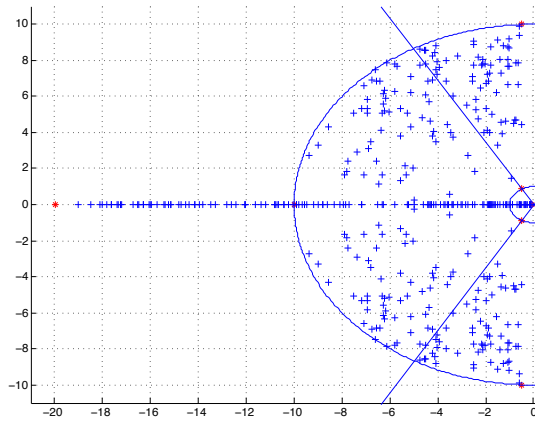


Les croix sont les pôles des sommets du polytope. On remarque que ces sommets ne sont pas tous dans les régions définies ci-dessus. La raison de cela est que le modèle polytopique est une approximation (extérieure) du modèle de départ. Les réalisations du système initial sont garantie d'être parmi des réalisations du modèle polytopique, mais certaines réalisations du modèle polytopique ne sont pas incluses dans les réalisations du système originel. Il en va de même pour les pôles. Les pôles du système polytopique peuvent ne pas correspondre avec les contraintes connues sur les pôles du système originel.

On trace ensuite les pôles de 200 tirages aléatoires de réalisations du modèle polytopique

```
for ii=1:200
p=pole(usample(usp1));
plot(real(p),imag(p),'+');
end
```

Le résultat et tracé sur la figure suivante

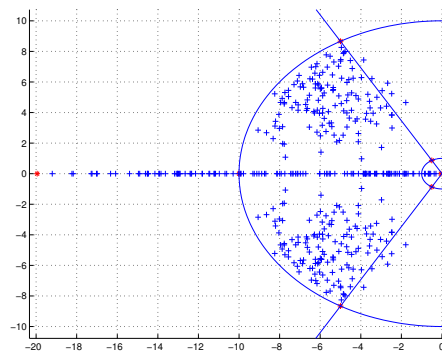


Les pôles des réalisations pour des tirages aléatoires des incertitudes intervalles (et donc polytopiques) sont tracés en bleu. Ils ne sont là aussi pas tous dans la région connue pour les pôles du système incertain originel, pour la même raison que les sommets.

Si on fait sur une autre figure le même exercice pour le second modèle polytopique,

```
figure(2); hold on;
for ii=1:usp2.nb
p=pole(usp2(ii));
plot(real(p),imag(p),'r*');
end
r1=region('disc',0,1);
plot(r1);
r2=region('disc',0,10);
plot(r2);
r3=region('plane',0,asin(0.5));
plot(r3);
for ii=1:200
p=pole(usample(usp2));
plot(real(p),imag(p),'+');
end
```

on trouve la figure suivante :



Certains des sommets collent mieux cette fois-ci au problème originel et par conséquent c'est

également le cas pour les tirages aléatoires. Cependant on constate que sur l'axe réel il y a encore beaucoup de valeurs des pôles qui ne correspondent en rien à la réalité du système. Le second modèle polytopique est meilleur que le premier pour représenter le système tout en continuant d'englober plus de réalisations que nécessaire.

1.5 On ne peut rien conclure à ce stade sur la stabilité du système incertain. Sur 200 tirages aléatoires des incertitudes tous les pôles sont à partie réelle négative (toutes les réalisations sont stables). Mais cela ne permet pas de conclure qu'aucune réalisation n'a de pôles à partie réelle positive (qu'aucune combinaison des paramètres incertains ne conduit à une réalisation instable).

1.6 Code pour $\alpha = 3$ appliqué au modèle `usp1` :

```
>> alpha=3;
>> P=sdpvar(2,2,'sym');
>> quiz=[P>=1];
>> for ii=1:usp1.nb
quiz=quiz+...
[usp1(ii).a'*P+P*usp1(ii).a<=2*alpha*P];
end
>> solvesdp(quiz)
[... affichage sur le calcul ...]
info: 'Successfully solved (SDPT3-4)'
```

On en conclut que les pôles sont garantis d'être à partie réelle plus petite que 3.

En anticipant la question 1.8, les LMI des tests ci-dessus sont pré-codées dans Romuloc. La commande `ctrpb` permet de dire que l'on cherche à résoudre un problème d'analyse. Avec l'opérateur surchargé `+` (c'est de la programmation objet) on peut y associer à l'aide de la commande `dstability` le problème à analyser, à savoir ici la localisation des pôles du système incertain dans une région du plan complexe. Voir l'utilisation de ces commandes dans les lignes suivantes :

```
>> quiz=ctrpb('analysis')...
+dstability(usp1,region('plane',2.992))
>> solvesdp(quiz);
D-stability assessed for region
Half-plane such that: Re(z)<2.992

>> quiz=ctrpb('analysis')...
+dstability(usp1,region('plane',2.991))
>> solvesdp(quiz);
Infeasible problem
```

De ces deux tests on conclut que les réalisations

du premier polytope sont telles que les pôles sont garantis d'être à partie réelle inférieure à 2.992. Mais les test LMI ne permettent pas de dire mieux que cela (LMI infaisables pour des valeurs plus petites de α). De la même façon avec les tests ci-dessus on montre que les pôles du second polytope ont une partie réelle inférieure à 2.048. Il est normal que l'on arrive à une plus petite borne que pour le premier polytope, car le second polytope est inclus dans le premier.

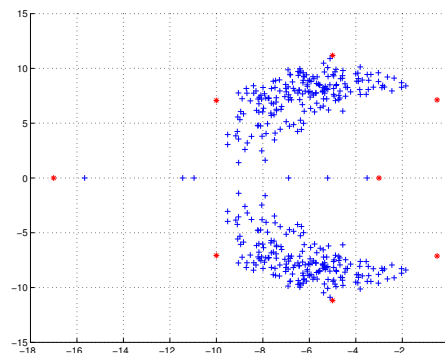
```
>> quiz=ctrpb('analysis')...
+dstability(usp2,region('plane',2.048));
>> solvesdp(quiz);
D-stability assessed for region
Half-plane such that: Re(z)<2.048
```

```
>> quiz=ctrpb('analysis')...
+dstability(usp2,region('plane',2.047));
>> solvesdp(quiz);
Infeasible problem
```

1.7 On calcule la boucle fermée avec la commande `feedback` comme suit (ici uniquement pour le second polytope). Et pour avoir une idée de où se trouvent les pôles on trace les pôles des sommets et de 200 réalisations.

```
>> usp2bf=feedback(usp2*50,1);
>> figure(3); hold on;
>> for ii=1:usp2bf.nb
p=pole(usp2bf(ii));
plot(real(p),imag(p),'r*');
end
>> for ii=1:200
p=pole(usample(usp2bf));
plot(real(p),imag(p),'+');
end
```

Le résultat est donné sur la figure suivante :



Avec les commandes qui suivent on trouve que les LMI utilisées précédemment permettent de

prouver que les poles sont robustement (pour toutes les réalisations du système incertain) à partie réelle inférieure à 0.741. Mais Les LMI sont infaisables pour des valeurs plus faibles de α ce qui ne permet donc pas de conclure quant à la stabilité du système (il faudrait pour cela montrer que les pôles sont à partie réelle inférieure à zéro).

```
>> quiz=ctrpb('analysis')...
+dstability(usp2bf,region('plane',0.741));
>> solvesdp(quiz);
D-stability assessed for region
Half-plane such that: Re(z)<0.741
```

```
>> quiz=ctrpb('analysis')...
+dstability(usp2bf,region('plane',0.740));
>> solvesdp(quiz);
Infeasible problem
```

1.8 On constate un large écart entre le test par tirages aléatoires qui tend à faire penser (de façon optimiste) que les pôles de la boucle fermée sont à partie réelle inférieure à -0.5 (pôle le plus à droite sur la figure), et les tests LMI (pessimistes) qui garantissent uniquement que les pôles sont à partie réelle inférieure à 0.741. Il serait bon de réduire cet écart et en particulier de calculer des bornes supérieures moins pessimistes. Pour cela on peut faire appel à des conditions LMI qui, contrairement à celles exposées plus haut où une seule matrice P était utilisée pour tout le polytope, prouvent l'existence de fonctions de Lyapunov dépendant des paramètres. Un tel résultat est codé dans Romuloc, on l'obtient comme suit :

```
>> quiz=ctrpb('analysis','pdlf')...
+dstability(usp2bf,region('plane',-0.499));
>> solvesdp(quiz);
D-stability assessed for region
Half-plane such that: Re(z)<-0.499
```

Il permet sur cet exemple de prouver exactement la borne optimiste obtenue par tirages aléatoires.

Travaux dirigés - Modélisation incertaine LFT

Octobre 2016

Exercice 1 Construire un modèle LFT du système suivant

$$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = u$$

comportant deux paramètres incertains :

$$\zeta \in [0.5, 1] \quad , \quad \omega_n \in [1, 10].$$

représentés par des expressions normalisées telles que

$$\zeta = \zeta_0 + r_\zeta \delta_\zeta \quad , \quad \omega_n = \omega_0 + r_\omega \delta_\omega$$

avec $|\delta_\zeta| \leq 1$ et $|\delta_\omega| \leq 1$.

Analyser la stabilité robuste de ce système en suivant la démarche appliquée au premier TD.

Exercice 2

Pour chacune des LFT suivantes : donner une représentation sous forme de schémas blocs ; calculer la forme rationnelle correspondante.

$$\delta \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]$$

$$\left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta_1 \mathbf{1}_2)$$

$$\left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] + \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta_2 \mathbf{1}_2)$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] + \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta_1 \mathbf{1}_2)$$

$$(\delta_1 \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]) \cdot \left(\left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star \delta_2 \right)$$

$$\delta \star \left[\begin{array}{c|cc} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right]$$

$$\left[\begin{array}{c|cc} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star \delta$$

$$\left(\left[\begin{array}{c|cc} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star \delta \right)^{-1}$$

$$\delta \star \left[\begin{array}{c|cc} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star k$$

Exercice 3

Donner la matrice de transfert incertaine du système suivant

$$\begin{pmatrix} \dot{x} \\ z_\Delta \\ y \end{pmatrix} = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ w_\Delta \\ u \end{pmatrix}$$

$$w_\Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} z_\Delta$$

Exercice 4

Donner une représentation d'état de la boucle suivante

$$\begin{pmatrix} \dot{x} \\ y \end{pmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix}$$

$$\begin{pmatrix} \dot{x}_K \\ u \end{pmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{pmatrix} x_K \\ y_c - y \end{pmatrix}$$

Exercice 5

Pour chacune des LFT suivantes (Σ , K et W sont des matrices de transfert de dimensions appropriées) : donner une représentation sous forme de schémas blocs ; donner une représentation LFT sous la forme d'un bouclage avec l'incertitude Δ .

$$\left[\begin{array}{c|c} 0 & 1 \\ 1 & (\Sigma(K + W\Delta)) \end{array} \right] \star (-1)$$

$$((1 + \Delta W)\Sigma K) \star (-1)$$

$$\left[\begin{array}{c|c} 0 & 1 \\ ((1 + \Delta W)\Sigma & ((1 + \Delta W)\Sigma K) \end{array} \right] \star (-1)$$

$$((1 + \Delta_m W_m)\Sigma(K + W_p \Delta_p)) \star (-1)$$

Travaux dirigés - Modélisation incertaine LFT - Corrigé

Octobre 2016

Exercice 1

En posant $w_1 = \omega_n z_1$ où $z_1 = 2\zeta\dot{y} + \omega_n y$ l'équation devient

$$\begin{aligned}\ddot{y} + w_1 &= u \\ z_1 &= 2\zeta\dot{y} + \omega_n y\end{aligned}$$

En posant $w_2 = \omega_n z_2$ où $z_2 = y$ le système d'équations devient

$$\begin{aligned}\ddot{y} + w_1 &= u \\ z_1 &= 2\zeta\dot{y} + w_2 \\ z_2 &= y\end{aligned}$$

En posant $w_3 = \zeta z_3$ où $z_3 = \dot{y}$ le système d'équations devient

$$\begin{aligned}\ddot{y} + w_1 &= u \\ z_1 &= 2w_3 + w_2 \\ z_2 &= y \\ z_3 &= \dot{y}\end{aligned}$$

Si on récapitule on posant $z_\Delta = (z_1 \ z_2 \ z_3)^T$ et $w_\Delta = (w_1 \ w_2 \ w_3)^T$ on a

$$w_\Delta = \Delta z_\Delta = \begin{bmatrix} \omega_n & & \\ & \omega_n & \\ & & \zeta \end{bmatrix} z_\Delta$$

et les relations linéaires liant le vecteur d'état x , l'entrée u et les entrées/sorties exogènes w_Δ, z_Δ sont :

$$\begin{pmatrix} \dot{x} \\ z_\Delta \end{pmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} A & B_\Delta & B \\ C_\Delta & D_\Delta & D \end{bmatrix}} \begin{pmatrix} x \\ w_\Delta \\ u \end{pmatrix}$$

On remarque que l'incertitude qui boucle ce modèle n'est pas normalisée (coefficients ne sont pas compris entre -1 et 1). Pour la normaliser on peut au choix, recommencer le calcul avec des incertitudes normalisées, ou faire un changement de repère.

Dans un premier temps on refait le calcul avec des incertitudes normalisées $|\delta_\omega| \leq 1, |\delta_\zeta| \leq 1$ et

$$\omega_n = 5.5 + 4.5\delta_\omega, \quad \zeta = 0.75 + 0.25\delta_\zeta.$$

Le modèle s'écrit alors

$$\ddot{y} + (1.5 + 0.5\delta_\zeta)(5.5 + 4.5\delta_\omega)\dot{y} + (5.5 + 4.5\delta_\omega)^2 y = u.$$

ou encore en développant astucieusement

$$\begin{aligned}\ddot{y} + 5.5(1.5 + 0.5\delta_\zeta)\dot{y} + 5.5^2 y + \hat{w}_1 \\ = \ddot{y} + 8.25\dot{y} + 2.75\delta_\zeta\dot{y} + 30.25y + \hat{w}_1 = u\end{aligned}$$

avec \hat{w}_1 qui contient tous les termes où l'on peut factoriser δ_ω . Plus précisément on a $\hat{w}_1 = \delta_\omega \hat{z}_1$ avec

$$\begin{aligned}\hat{z}_1 &= 4.5(1.5 + 0.5\delta_\zeta)\dot{y} + 2 \cdot 5.5 \cdot 4.5y + 4.5^2 \delta_\omega y \\ &= 6.75\dot{y} + 2.25\delta_\zeta\dot{y} + 49.5y + 20.25\delta_\omega y\end{aligned}$$

A ce stade il ne reste plus qu'à poser $\hat{w}_2 = \delta_\omega \hat{z}_2$ avec $\hat{z}_2 = y$ et $\hat{w}_3 = \delta_\zeta \hat{z}_3$ avec $\hat{z}_3 = \dot{y}$ pour avoir le système d'équations suivant

$$\begin{aligned}\ddot{y} + 8.25\dot{y} + 30.25y &= \hat{w}_1 - 2.75\hat{w}_3 + u \\ \hat{z}_1 &= 6.75\dot{y} + 2.25\hat{w}_3 + 49.5y + 20.25\hat{w}_2 \\ \hat{z}_2 &= y \\ \hat{z}_3 &= \dot{y}\end{aligned}$$

En prenant le même vecteur d'état $x^T = (y \ \dot{y})$ cela correspond à la représentation d'état suivante :

$$\begin{pmatrix} \dot{x} \\ \hat{z}_\Delta \end{pmatrix} = M \begin{pmatrix} x \\ \hat{w}_\Delta \\ u \end{pmatrix}$$

avec $M =$

$$\left[\begin{array}{cc|cc|cc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -30.25 & -8.25 & -1 & 0 & -2.75 & 1 & 1 \\ \hline 49.5 & 6.75 & 0 & 20.25 & 2.25 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right].$$

bouclée par $\hat{w}_\Delta = \hat{\Delta} z_\Delta$ où

$$\hat{\Delta} = \begin{bmatrix} \delta_\omega & & \\ & \delta_\omega & \\ & & \delta_\zeta \end{bmatrix}.$$

L'approche alternative est de partir de la LFT déjà obtenue et de remarquer que

$$w_\Delta = \Delta_0 z_\Delta + R \hat{\Delta} z_\Delta$$

où Δ_0 représente de centre des intervalles et R les variations autour de ce centre :

$$\Delta_0 = \begin{bmatrix} 5.5 & & & \\ & 5.5 & & \\ & & 0.75 & \\ 4.5 & & & \\ & 4.5 & & \\ & & & 0.25 \end{bmatrix},$$

En posant $w_0 = \Delta_0 z_\Delta$ et $\hat{w}_\Delta = \hat{\Delta} z_\Delta$ le système s'écrit

$$\begin{pmatrix} \dot{x} \\ z_\Delta \\ z_\Delta \end{pmatrix} = \begin{bmatrix} A & B_\Delta R & B & B_\Delta \\ C_\Delta & D_\Delta R & D & D_\Delta \\ C_\Delta & D_\Delta R & D & D_\Delta \end{bmatrix} \begin{pmatrix} x \\ \hat{w}_\Delta \\ u \\ w_0 \end{pmatrix}$$

avec un bouclage "haut" $\hat{w}_\Delta = \hat{\Delta} z_\Delta$ et un bouclage "bas" $w_0 = \Delta_0 z_\Delta$. En fermant la boucle basse les propriétés des LFT donnent

$$\begin{pmatrix} \dot{x} \\ z_\Delta \end{pmatrix} = \begin{bmatrix} \hat{A} & \hat{B}_\Delta & \hat{B} \\ \hat{C}_\Delta & \hat{D}_\Delta & \hat{D} \end{bmatrix} \begin{pmatrix} x \\ \hat{w}_\Delta \\ u \end{pmatrix}$$

où les matrices sont données par la formule suivante

$$\begin{bmatrix} \hat{A} & \hat{B}_\Delta & \hat{B} \\ \hat{C}_\Delta & \hat{D}_\Delta & \hat{D} \end{bmatrix} = \begin{bmatrix} A & B_\Delta R & B \\ C_\Delta & D_\Delta R & D \end{bmatrix} + \begin{bmatrix} B_\Delta \\ D_\Delta \end{bmatrix} \Delta_0 (1 - D_\Delta \Delta_0)^{-1} \begin{bmatrix} C_\Delta & D_\Delta R & D \end{bmatrix}$$

Le résultat est équivalent à celui de la première approche :

$$\begin{bmatrix} \hat{A} & \hat{B}_\Delta & \hat{B} \\ \hat{C}_\Delta & \hat{D}_\Delta & \hat{D} \end{bmatrix} \equiv M.$$

Ces deux matrices sont égales à un changement de base près (changement de base tel que $T \hat{\Delta} T^{-1} = \hat{\Delta}$).

En complément, voici comment on peut définir le système LFT dans Romuloc. On commence par définir le système dans l'espace d'état :

```
>> s = ssmodel;
>> s.A=[0 1; -30.25 -8.25];
>> s.Bd=[0 0 0;-1 0 -5.5];
>> s.Bu=[0 ; 1];
>> s.Cd=[49.5 6.75;1 0;0 0.5];
>> s.Ddd=[0 20.25 4.5;0 0 0;0 0 0];
      n=2      md=3      mu=1
      n=2 dx = A*x + Bd*wd + Bu*u
      pd=3 zd = Cd*x + Ddd*wd
continuous time
```

puis les deux incertitudes (ici des incertitudes intervalles comprises entre -1 et 1) et la matrice incertaine composée de celles-ci :

```
>> do = uinter(-1,1);
>> dx = uinter(-1,1);
>> delta = diag(do,do,dx)
diagonal structured uncertainty
size: 3x3 | nb blocks: 3
| independent blocks: 2
wd = diag( #1 #1 #2 ) * zd
index size repeat. RorC dist. nature
#1 1x1 2 real det. LTI interval 1 param
#2 1x1 1 real det. LTI interval 1 param
```

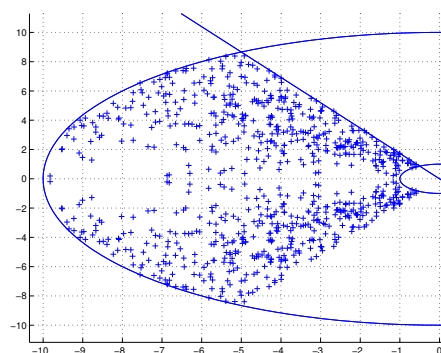
finalement on déclare le modèle LFT en associant ces deux objets :

```
>> us = ussmodel(s,delta);
```

Par curiosité on peut pour ce système incertain reprendre l'analyse faite lors du TD précédent. Les commandes qui suivent permettent de tracer les pôles pour des tirages aléatoires des incertitudes, et de les comparer avec les régions dans lesquels ils sont supposés être

```
>> figure(4); hold on;
>> for ii=1:200
      p=pole(usample(us));
      plot(real(p),imag(p),'+');
    end
>> plot(region('disc',0,1));
>> plot(region('disc',0,10));
>> plot(region('plane',0,asin(0.5)));
```

Le résultat est dans la figure qui suit. Les pôles sont exactement dans la région attendue. C'est normal car le modèle LFT représente exactement le modèle incertain initial.



Si l'on veut analyser la stabilité robuste de ce système on peut par exemple appliquer le théorème du petit gain. Pour cela on fait la manipulation suivante dans Romuloc qui consiste

à re-déclarer les entrées/sorties $\hat{w}_\Delta/\hat{z}_\delta$ en tant qu'entrées/sorties de performance w/z :

```
>> s2=ssmodel;
>> s2.A=s.A;
>> s2.Bw=s.Bd;
>> s2.Cz=s.Cd;
>> s2.Dzw=s.Ddd;
>> norm(s2,inf)
ans =
    21.1644
```

Le calcul de la norme H_∞ donne une valeur $\mu = 21.1644$. Ce système, bouclé avec une incertitude pleine, complexe, bornée en norme Δ , est donc, d'après le théorème du petit gain, stable pour toute incertitude telle que $\|\Delta\|_\infty \leq \frac{1}{\mu} = 0.0472$. Dans notre cas l'incertitude est diagonale structurée avec des coefficients réels, ce qui est un sous cas d'une incertitude pleine non-structurée. La norme H_∞ de la matrice diagonale est égale à la valeur absolue maximale de chacun des coefficients de la diagonale. Le théorème du petit gain nous permet donc de conclure que le système est stable si $|\delta_\omega| \leq 0.0472$ et $|\delta_\zeta| \leq 0.0472$. C'est très loin des véritables bornes sur les incertitudes. Le théorème du petit-gain donne un résultat pessimiste dès lors que l'incertitude est structurée (ici le résultat est très pessimiste).

Une autre façon d'analyser la stabilité robuste c'est de faire appel aux outils LMI de Romuloc, come nous l'avions fait pour les systèmes polytopiques dans le TD précédent. Sans entrer dans les détails de comment ces résultats sont obtenus (cette partie du cours a été retirée du programme cette année du fait de la réduction du volume horaire), on trouve avec une fonction de Lyapunov unique pour toutes les incertitudes :

```
>> quiz=ctrpb('analysis')...
+dstability(us,region('plane',2.048));
>> solvesdp(quiz);
D-stability assessed for region
Half-plane such that: Re(z)<2.048
```

```
>> quiz=ctrpb('analysis')...
+dstability(us,region('plane',2.047));
>> solvesdp(quiz);
Infeasible problem
```

et avec une fonction de Lyapunov dépendant des paramètres

```
quiz=ctrpb('analysis','pdlf')...
```

```
+dstability(us,region('plane',-0.499));
solvesdp(quiz);
```

```
D-stability assessed for region
Half-plane such that: Re(z)<-0.499
```

Le test avec fonctions de Lyapunov dépendant des paramètres donne cette fois encore un résultat non-pessimiste (on sait qu'une des réalisations du systèmes a ses pôles à partie réelle égale à -0.5).

Exercice 2

$$\delta \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] = \left[\begin{array}{cc} 1-\delta & -2\delta \\ 1 & 3 \end{array} \right]$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] = \frac{3-3\delta+\delta^2}{1-\delta+\delta^2}$$

$$\left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta_1 \mathbf{1}_2) = \frac{-\delta+\delta^2}{1-4\delta+3\delta^2}$$

$$\left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]$$

Calcul impossible : la LFT est mal posée car la matrice suivante n'est pas inversible

$$1 - \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 3 & -3 \end{bmatrix}$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] + \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta_2 \mathbf{1}_2)$$

$$(\delta_1 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] + (\delta_2 \mathbf{1}_2) \star \left[\begin{array}{c|cc} 1 & 0 & -1 \\ 1 & 3 & 0 \\ 1 & 2 & 0 \end{array} \right]$$

$$\left[\begin{array}{cc} \delta_1 \mathbf{1}_2 & \\ & \delta_2 \mathbf{1}_2 \end{array} \right] \star \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 2 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 1 & 1 & 2 & 3 \end{array} \right]$$

$$\begin{aligned}
(\delta 1_2) \star \left[\begin{array}{cc|c} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] + \left[\begin{array}{cc|c} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star (\delta 1_2) \\
= \\
(\delta 1_4) \star \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 2 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 1 & 1 & 2 & 3 \end{array} \right]
\end{aligned}$$

qui est minimale car il s'agit bien d'une fraction rationnelle de polynômes d'ordre 4

$$\begin{aligned}
&= \frac{3 - 3\delta + \delta^2}{1 - \delta + \delta^2} + \frac{-\delta + \delta^2}{1 - 4\delta + 3\delta^2} \\
&= \frac{3(1 - 3.36\delta)(1 - \delta)(1 - 0.9731\delta + 0.3968\delta^2)}{(1 - 3\delta)(1 - \delta)(1 - \delta + \delta^2)}
\end{aligned}$$

Cependant elle contient une simplification si $\delta \neq 1$. Donc si $\delta \neq 1$, il existe une représentation où δ n'est répétée que 3 fois.

$$\begin{aligned}
(\delta 1 \star \left[\begin{array}{cc|c} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]) \cdot \left(\left[\begin{array}{cc|c} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right] \star \delta_2 \right) \\
= \\
(\delta 1 \star \left[\begin{array}{cc|c} 0 & 1 & 2 \\ -1 & 1 & 0 \\ 0 & 1 & 3 \end{array} \right]) \cdot (\delta_2 \star \left[\begin{array}{c|cc} 3 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & -1 & 1 \end{array} \right]) \\
= \\
\left[\begin{array}{cc} \delta_1 & \delta_2 \end{array} \right] \star \left[\begin{array}{cc|cc} 0 & 2 & -2 & 3 \\ 0 & 3 & 0 & 1 \\ -1 & 2 & 0 & 1 \\ 0 & 2 & -3 & 4 \end{array} \right] \\
= \\
\left[\begin{array}{c} 2\delta_1 \\ -3 \end{array} \quad \frac{1-2\delta_1-\delta_2+7\delta_1\delta_2}{1-3\delta_2} \quad \frac{4-10\delta_2}{1-3\delta_2} \right]
\end{aligned}$$

$$\begin{aligned}
\delta \star \left[\begin{array}{cc|c} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \\
= \\
\left[\begin{array}{c} \frac{s+2-\delta}{s+2} \\ 1 \end{array} \quad \frac{-2\delta}{(s+1)\frac{3}{s}(s+2)} \right]
\end{aligned}$$

$$\begin{aligned}
\left[\begin{array}{cc|c} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star \delta \\
= \\
\left[\begin{array}{c} 0 \\ \frac{-1}{s+2} \end{array} \quad \frac{s^2+(1-\delta)s-3\delta}{s^2+(1-3\delta)s-3\delta} \quad 1 \right]
\end{aligned}$$

$$\left(\left[\begin{array}{cc|c} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star \delta \right)^{-1}$$

$$\left(\delta \star \left[\begin{array}{cc|c} \frac{3}{s+1} & 0 & 1 \\ \frac{3}{s+1} & 0 & 1 \\ 0 & -\frac{1}{s+2} & 1 \end{array} \right] \right)^{-1}$$

$$\delta \star \left[\begin{array}{cc|c} \frac{s+3}{s(s+1)} & 1 & 0 \\ \frac{2(s+2)}{s+1} & s+2 & -(s+2) \\ \frac{2}{s+1} & 1 & 0 \end{array} \right]$$

$$\left[\begin{array}{cc} \frac{(s+2)(s^2+(1+\delta)s-3\delta)}{s^2+(1-\delta)s-3\delta} & -s-2 \\ \frac{s^2+(1+\delta)s-3\delta}{s^2+(1-\delta)s-3\delta} & 0 \end{array} \right]$$

$$\delta \star \left[\begin{array}{cc|c} 0 & 1 & \frac{2}{s+1} \\ -\frac{1}{s+2} & 1 & 0 \\ 0 & 1 & \frac{3}{s} \end{array} \right] \star k$$

$$\frac{s^3 + (3 - \delta - 3k)s^2 + (\delta k - \delta - 9k)s + 3\delta k}{(s+1)(s+2)(s-3k)}$$

Exercice 3

$$\begin{pmatrix} \dot{x} \\ z_\Delta \\ y \end{pmatrix} = \left[\begin{array}{ccc|ccc} -2 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array} \right] \begin{pmatrix} x \\ w_\Delta \\ u \end{pmatrix}$$

$$w_\Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} z_\Delta$$

Si on réalise le bouclage LFT avec $x = s^{-1}\dot{x}$ on trouve la représentation relativement complexe suivante (inexploitable pour des calculs "à la main")

$$\begin{pmatrix} z_\Delta \\ y \end{pmatrix} = \frac{1}{(s+1)(s+2)} M(s) \begin{pmatrix} w_\Delta \\ u \end{pmatrix}$$

avec $M(s) =$

$$\left[\begin{array}{cc|c} (s+1)(s+2) & 2(s+2) & (s+2) \\ (s+1) & 2 & s^2+3s+3 \\ (s+1)(s+3) & 2 & 1 \\ 0 & 2(s+1) & (s+2)^2 \end{array} \right]$$

Si par contre on réaliste le bouclage avec l'incertitude on a :

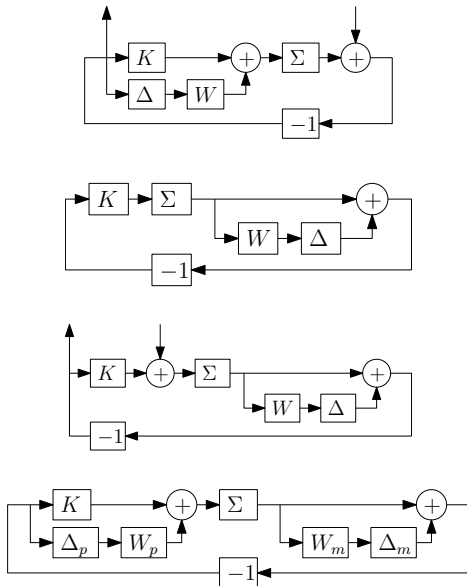
$$\begin{pmatrix} \dot{x} \\ y \end{pmatrix} = \left[\begin{array}{cc|c} -2 & \frac{1}{1-\delta_1} & 0 \\ 2\delta_2 & -1 & 1+2\delta_2 \\ \hline 1 & \frac{\delta_1}{1-\delta_1} & 0 \\ 0 & 1 & 0 \end{array} \right] \begin{pmatrix} x \\ u \end{pmatrix}$$

On fait alors le bouclage avec $x = s^{-1}\dot{x}$ sur cette dernière forme et on trouve

$$y = P(s, \delta_1, \delta_2) \begin{bmatrix} s+3 \\ (s+2)(1-\delta_1) \end{bmatrix} u$$

avec $P(s, \delta_1, \delta_2) = \frac{1+2\delta_2}{(1-\delta_1)s^2+(1-\delta_1)s+2-2\delta_1-2\delta_2}$.

Exercice 5



Travaux dirigés - Synthèse H_∞
 Octobre 2016

Soit le système

$$G(s, \delta) = \frac{1 + 0.2\delta}{s^3 + 2(1 + \delta)s^2 + 2s + 1}$$

où δ est une incertitude scalaire réelle $|\delta| \leq 1$.
 On souhaite trouver un correcteur qui stabilise robustement le système tout en satisfaisant la contrainte suivante :

Erreur faible (inférieure à 1 en norme L_2) pour des signaux de perturbation basse fréquence sur l'entrée du système ($W_i(s) = \frac{1}{10s+1}$, signal de perturbation de norme L_2 inférieure à 1).

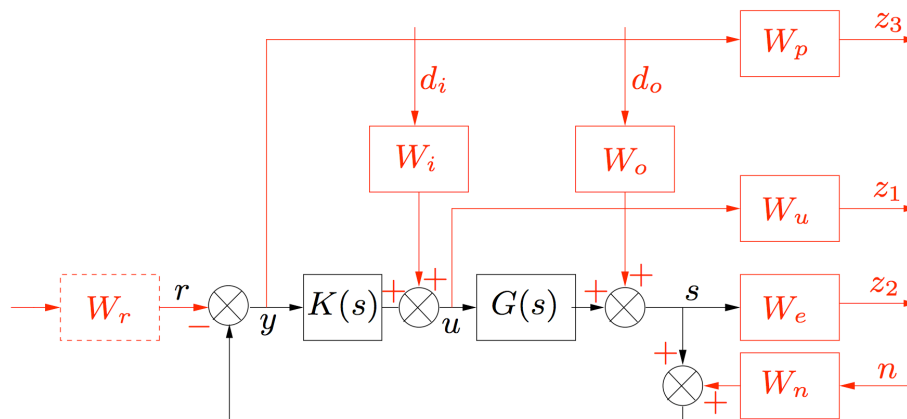
1. Construire une représentation LFT telle que

$$G(s, \delta) = \delta \star \Sigma(s)$$

avec $\Sigma(s)$ représentée dans l'espace d'état. Définir $\Sigma(s)$ dans Matlab avec le formalisme de la Control Toolbox (`ss`).

2. Inclure au modèle précédent les entrées/sorties de performance et réécrire à l'aide du théorème du petit gain la contrainte comme un problème de performance vis-à-vis d'une seconde incertitude. Définir le nouveau système dans Matlab avec le formalisme de la Control Toolbox et chercher une représentation espace d'état minimale (`tf ss minreal`).

3. Faire le calcul d'un correcteur H_∞ à l'aide de la commande `hinfscn`. Le correcteur obtenu répond-il à la question posée (`lft norm`) ?



Travaux dirigés - Synthèse H_∞ - Corrigé
Octobre 2016

Le système s'écrit :

$$s^3y + 2(\delta + 1)s^2y + 2sy + y = u + 0.2\delta u$$

On pose $w_\delta = \delta z_\delta$ avec $z_\delta = -s^2y + 0.1u$ ce qui donne

$$(s^3 + 2s^2 + 2s + 1)y = 2w_\delta + u$$

et on en déduit

$$y = \frac{2}{s^3+2s^2+2s+1}w_\delta + \frac{1}{s^3+2s^2+2s+1}u$$

$$z_\delta = \frac{-2s^2}{s^3+2s^2+2s+1}w_\delta - \frac{s^2}{s^3+2s^2+2s+1}u + 0.1u$$

c'est à dire :

$$\begin{pmatrix} z_\delta \\ y \end{pmatrix} = \underbrace{\begin{bmatrix} \Sigma_{\delta\delta}(s) & \Sigma_{\delta u}(s) \\ \Sigma_{y\delta}(s) & \Sigma_{yu}(s) \end{bmatrix}}_{\Sigma(s)} \begin{pmatrix} w_\delta \\ u \end{pmatrix}$$

avec

$$\begin{aligned} \Sigma_{\delta\delta}(s) &= \frac{-2s^2}{s^3+2s^2+2s+1} \\ \Sigma_{\delta u}(s) &= \frac{0.1s^3-0.8s^2+0.2s+0.1}{s^3+2s^2+2s+1} \\ \Sigma_{y\delta}(s) &= \frac{2}{s^3+2s^2+2s+1} \\ \Sigma_{yu}(s) &= \frac{1}{s^3+2s^2+2s+1} \end{aligned}$$

Dans Matlab ceci s'écrit par exemple de la façon suivante

```
num={[-2 0 0] , [0.1 -0.8 0.2 0.1] ;
      2 , 1}
den = [1 2 2 1];
Sigma = tf( num , den )
```

Pour avoir une représentation d'état on applique la commande `ss` dans Matlab. Ce qui donne

`ss(Sigma)`

```
a =
      x1      x2      x3      x4      x5      x6
x1      -2      -1     -0.5      0      0      0
x2       2       0       0       0      0      0
x3       0       1       0       0      0      0
x4       0       0       0      -2     -1     -0.5
x5       0       0       0       2      0      0
x6       0       0       0       0      1      0

b =
      u1      u2
x1       2      0
x2       0      0
x3       0      0
x4       0      1
```

```
x5      0      0
x6      0      0
c =
      x1      x2      x3      x4      x5      x6
y1     -1      0      0     -1      0      0
y2      0      0     0.5      0      0     0.5

d =
      u1      u2
y1      0     0.1
y2      0      0
```

qui est un système d'ordre 6 alors que le système initial est d'ordre 3. On peut le réduire dans Matlab comme ceci

```
minreal(ss(Sigma))
3 states removed.
a =
      x1      x2      x3
x1     -1.824   -0.6503   0.4679
x2      1.33   -0.006137  -0.8946
x3      1.17    1.521    -0.1697

b =
      u1      u2
x1     1.215   0.6074
x2     0.2674  0.1337
x3    -0.6728 -0.3364

c =
      x1      x2      x3
y1     -1.215   -0.2674   0.6728
y2     0.3554  -0.09546   0.6038

d =
      u1      u2
y1      0     0.1
y2      0      0
```

Une alternative est de travailler directement dans l'espace d'état. On pose $x = (y \ \dot{y} \ \ddot{y})$ ce qui donne les équations suivantes

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_3, \\ \dot{x}_3 &= -2x_2 - 2x_3 - x_1 + 2w_\delta + u, \\ z_\delta &= -x_3 + 0.1u, \\ y &= x_1 \end{aligned}$$

c'est à dire la représentation d'état :

$$\begin{pmatrix} z_\delta \\ y \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -2 & -1 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2 & 1 \end{bmatrix} \begin{pmatrix} w_\delta \\ u \end{pmatrix}$$

$$\begin{pmatrix} z_\delta \\ y \end{pmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} w_\delta \\ u \end{pmatrix}$$

Les deux représentations d'état sont différentes mais équivalentes (à un changement de base près).

Une troisième façon d'obtenir une représentation d'état en utilisant la Robust control toolbox est la suivante :

```
d=ureal('delta',1);
G = tf(1+0.2*d, [1 2*(1+d) 2 1])
Uncertain continuous-time model
with 1 outputs, 1 inputs, 3 states.
The model uncertainty consists of:
delta: Uncertain real, nominal = 1,
variability = [-1,1], 2 occurrences
M=lftdata(G)
a =
    x1  x2  x3
x1  -4  -2  -1
x2   1   0   0
x3   0   1   0
b =
    u1  u2  u3
x1  -2   0   1
x2   0   0   0
x3   0   0   0
c =
    x1  x2  x3
y1   1   0   0
y2   0   0   1
y3   0   0  1.2
d =
    u1  u2  u3
y1   0   0   0
y2   0   0   0
y3   0  0.2   0
```

Cette dernière représentation a le désavantage de faire apparaître une boucle vis-à-vis d'une incertitude répétée deux fois.

2. On ajoute au système les signaux de performance à savoir

- pour la mesure de l'erreur

$$z_1 = y_c - y$$

- où y_c est la consigne (nulle par la suite)
- pour les perturbations en entrée du système

$$u = u_c + W_i(s)w_1$$

où u_c est l'entrée de commande (sortie du correcteur) et w_1 sont les perturbations avant filtrage

- pour la mesure des efforts d'actionnement

$$z_2 = W_u u_c$$

- pour les bruits sur les capteurs

$$y_m = y + 0.04w_2$$

où y_m est le signal disponible pour la commande et w_2 sont les bruits normalisés ($0.1w_2$ est de norme 0.1 si w_2 est de norme 1)..

Le modèle complet s'écrit donc

$$\begin{pmatrix} z_\delta \\ z_1 \\ z_2 \\ y_m \end{pmatrix} = M(s) \begin{pmatrix} w_\delta \\ w_1 \\ w_2 \\ u_c \end{pmatrix}$$

où $M(s)$

$$\begin{aligned} &= \begin{bmatrix} \Sigma_{\delta\delta}(s) & \Sigma_{\delta u}(s)W_i(s) & 0 & \Sigma_{\delta u}(s) \\ -\Sigma_{y\delta}(s) & -\Sigma_{yu}(s)W_i(s) & 0 & -\Sigma_{yu}(s) \\ 0 & 0 & 0 & W_u(s) \\ \Sigma_{y\delta}(s) & \Sigma_{yu}(s)W_i(s) & 0.04 & \Sigma_{yu}(s) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{\delta\delta}(s) & \Sigma_{\delta u}(s) \\ \Sigma_{y\delta}(s) & \Sigma_{yu}(s) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & W_i(s) & 0 & 1 \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0.04 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & W_u(s) \end{bmatrix} \end{aligned}$$

Dans Matlab ce modèle se construit comme suit (par exemple)

```
Wi=tf(1,[10 1]);
Wu=tf([1 0],[1 10]);
H=[1 0;0 -1;0 0;0 1]*Sigma*[1 0 0 0;0 Wi 0 1]...
+[0;0;0;1]*[0 0 0.1 0]+[0;0;Wu;0]*[0 0 0 1];
```

un modèle d'état minimal de ce système est obtenu avec

`Hss=minreal(ss(H))`

C'est un modèle d'ordre 5 avec 4 entrées 4 sorties. Les dernières entrées/sorties correspondent aux signaux de mesure/commande disponibles pour réaliser un correcteur. Les autres signaux sont tels que :

- Robustesse à l'incertitude

$$w_\delta = \delta z_\delta : |\delta| \leq 1$$

On sait (théorème du petit gain) que la robustesse vis à vis de cette incertitude est garantie si la norme H_∞ du transfert de w_δ à z_δ est strictement inférieure à 1.

— Erreur faible

On souhaite que la norme de la sortie z_1 soit bornée par 1 quand la norme de l'entrée w_1 est elle-même bornée par 1. Ce critère correspond exactement à chercher à ce que la norme induite L_2 du transfert w_1 vers z_1 soit inférieure à 1, ou encore (car nous sommes dans le cas de systèmes linéaires) à ce que la norme H_∞ du transfert de w_1 vers z_1 soit inférieure à 1.

— Faible sollicitation des actionneurs

Pour la même raison que ci-dessus cette performance est obtenue si la norme H_∞ du transfert de w_2 à z_2 est de norme H_∞ inférieure à 1.

Une condition suffisante pour que chacun de ces transferts soient individuellement de norme H_∞ plus petite que 1 est que le transfert entre le vecteur $w = (w_\delta \ w_1 \ w_2)^T$ et $z = (z_\delta \ z_1 \ z_2)^T$ soit de norme inférieure à 1.

On cherche donc un correcteur qui utilise pour la commande les signaux u et y (les derniers signaux du modèle **Hss**) et qui minimise la norme H_∞ entre w et z (trois premiers signaux du modèle **Hss**).

```
[kopt,N,g]=hinfsyn(Hss,1,1);  
g =  
    0.9724
```

Le correcteur obtenu **kopt** est un système dynamique du même ordre que **Hss** (d'où l'importance d'avoir réduit l'ordre de **Hss**). La boucle fermée **N** est un modèle d'ordre 10. Elle peut aussi être obtenue comme suit. Et sa norme H_∞ est exactement **g** :

```
Hcl=lft(Hss,kopt);  
norm(Hcl,inf)  
ans =  
    0.9724
```

Cette norme est inférieure à 1. C'est ce qui était souhaité. Ce correcteur stabilise robustement le système (pour tout $|\delta| \leq 1$) et garanti les critères d'erreur faible et de faible sollicitation des actionneurs.