Boîte à outils "Robust Multi-Objective Control" - Version 1

Fondements théoriques de RoMulOC et son utilisation illustrée sur des exemples

Dimitri Peaucelle & Denis Arzelier

LAAS-CNRS
7 avenue du colonel Roche, 31077 Toulouse cedex 4, FRANCE
Tel : (+33) 5-61-33-63-09   Fax : (+33) 5-61-33-69-69
Email : peaucelle@laas.fr

RÉSUMÉ. *Au cours des dix dernières années le formalisme des Inégalités Matricielles Linéaires a permis l'émergence de très nombreux résultats pour l'analyse et la synthèse robuste. Une caractéristique principale de ces résultats est la variété de critères de performances envisagés et de modèles incertains considérés. Même si chaque résultat est individuellement simple à programmer il nous est apparu nécessaire de regrouper tous les résultats dans une même boîte à outils et par là même faire apparaître les complémentarités de résultats produits séparément.*
Dans cet article nous présentons la première version de la boîte à outils "Robust Multi-Objective Control". Cet outil est distribué librement et fonctionne en complément de l'interface de programmation de problèmes d'optimisation YALMIP dans l'environnement Matlab©. La première version actuellement diffusée traite uniquement de problèmes d'analyse robuste. Les performances considérées sont la stabilité, la localisation de pôles, les coûts H_∞, H_2 et impulse-to-peak. Les modèles d'incertitudes sont tous décrits dans l'espace d'état et vont des représentations par matrices intervalles à des représentations rationnelles fortement structurées. Pour chaque problème considéré, l'utilisateur a la possibilité de choisir entre plusieurs fonctions de Lyapunov dépendant ou non des paramètres incertains ce qui conduit à des résultats plus ou moins pessimistes et de coût numérique plus ou moins grand. L'article présente à la fois les fondements théoriques de l'outil et illustre son fonctionnement sur des exemples.

ABSTRACT. *For the past ten years the formalism of Linear Matrix Inequalities has lead to many results for robust analysis and synthesis. One major characteristic of these results is the variety of performances and uncertain models that may be tackled. Although every individual result may be easily coded we felt necessary to gather all the results in a unique toolbox. It allows to put in evidence the complementarity of all these results developed separately.*
This paper is devoted to the first version of the "Robust Multi-Objective Control" toolbox. It is

1re soumission à *RS-JESA*, le 26 avril 2006

freely distributed and works in Matlab©environment along with the parser YALMIP. The version actually distributed tackles stability, pole location, H_∞, H_2 and impulse-to-peak problems. The uncertain models are all in state-space form and range from interval matrices to structured rational models. For each considered problem, the user has the possibility to choose between parameter dependent or independent Lyapunov functions. This gives more or less conservative conditions associated with different numerical burden. The paper exposes the underlying theoretical results and illustrates the use of the tool on some examples.

MOTS-CLÉS : LMI, Commande Robuste, MATLAB, YALMIP, Logiciel, Modèles

KEYWORDS: LMI, Robust Control, MATLAB, YALMIP, Software, Models

1. Introduction

Au cours des dix dernières années le formalisme des Inégalités Matricielles Linéaires (LMI) a permis l'émergence de très nombreux résultats pour l'analyse et la synthèse robustes. Une caractéristique majeure de ces résultats est la variété de critères de performance envisagés et de modèles incertains considérés. Même si chaque résultat est individuellement simple à programmer, il nous est apparu nécessaire de tous les regrouper dans une même boîte à outils. Cela permettra non seulement une meilleure diffusion de ces résultats de recherche récents, mais offrira l'opportunité de les tester massivement sur des exemples numériques d'applications. Finalement, RoMulOC pourra servir de plate-forme coopérative pour tous les chercheurs du domaine. Pour ces raisons la boîte à outils est conçue comme un logiciel distribué librement et nous espérons de nombreux retours et contributions.

A ce stade de développement de l'outil, l'accent est mis sur des fonctions pour la manipulation de modèles de systèmes linéaires invariants dans le temps (LTI) incertains et sur la programmation de méthodes d'analyse robuste.

La modélisation et la manipulation de modèles LTI est très similaire à ce qui est fait dans la Control toolbox de Matlab©. La principale différence est dans la prise en compte explicite des différences entre entrées de commande et entrées de perturbation d'une part et d'autre part entre sorties mesurées et sorties à commander. Tous les modèles considérés sont décrits dans l'espace d'état à temps continu ou à temps discret. Deux grandes classes de modèles incertains sont définis. L'une est telle que les matrices du modèle sont affines en les paramètres incertains et l'ensemble des incertitudes est décrit par un polytope. Pour ces modèles polytopiques, la plupart des outils théoriques connus est décrite dans la littérature (de Oliveira *et al.,* 1999, Geromel *et al.,* 2002, Peaucelle *et al.,* 2000, Xie *et al.,* 2004, Arzelier *et al.,* 2002). L'autre classe de modèles incertains est plus générale et repose sur la transformation Linéaire-Fractionnaire (LFT) utilisée dans le cadre de la théorie de la valeur singulière structurée μ. A ce stade de développement de l'outil logiciel, seules les incertitudes paramétriques constantes dans le temps sont considérées mais les outils de description de ces incertitudes sont très riches allant des incertitudes bornées en norme classiques aux incertitudes de type intervalle en passant par les incertitudes dissipatives (Peaucelle *et al.,* 2001b, Iwasaki *et al.,* 1998, Scorletti *et al.,* 1998). Un grand nombre de fonctions sont fournies pour les manipulations élémentaires des modèles incertains. Pour autant, cet outil ne doit pas être considéré comme un outil de modélisation à part entière. Pour cela nous recommandons l'utilisation de la LFR toolbox (Magni, 2005) qui sera prochainement interfacée avec RoMulOC.

Les critères de performance robuste considérés sont : la stabilité, la localisation des pôles, les normes H_∞ et H_2, et la pseudo-norme "Impulse-to-peak". Pour les deux classes de modèles incertains, deux catégories de résultats LMI sont programmés. La première catégorie est celle de la "stabilité quadratique" (une fonction de Lyapunov unique pour prouver la robustesse pour l'ensemble des incertitudes admissibles). La seconde catégorie fait appel aux fonctions de Lyapunov dépendant des paramètres

(PDLF). Pour la classe des modèles incertains polytopiques, les résultats PDLF sont issus de (Peaucelle *et al.,* 2000). Pour la classe des modèles incertains LFT les résultats PDLF sont ceux issus de la technique de séparation quadratique (Iwasaki *et al.,* 1998, Iwasaki *et al.,* 2001) (voire aussi la "full-block S-procedure" de (Scherer, 2000)).

L'article est organisé comme suit. Premièrement les différents modèles incertains sont présentés. La section 3 résume ensuite les outils théoriques programmés dans l'outil. Enfin, une longue section est dédiée à la description de l'utilisation de l'outil sur un exemple. L'article se clôt sur des conclusions et prospectives.

2. Modélisation des systèmes incertains

2.1. *Modèles LFT*

Pour tenir compte simultanément de lois de commande à synthétiser, de critères de performance sorties/entrées et d'incertitudes, la modélisation LFT (Transformée Linéaire-Fractionnaire) programmée dans RoMulOC distingue explicitement trois paires de sorties/entrées :

– Sorties de mesure/Entrées de commande. Notés y/u, ce sont les vecteurs de signaux accessibles pour la commande.

– Sorties à contrôler/Entrées de perturbations. Notés z/w, ces vecteurs de signaux permettent de définir des niveaux de performance parmi les critères de norme H_∞, H_2 ou "impulse-to-peak" (amplitude/impulsion).

– Sorties/Entrées exogènes. Notés z_Δ/w_Δ, ce sont des vecteurs de signaux fictifs permettant la modélisation LFT des incertitudes. Les incertitudes regroupées dans une matrices notée Δ interviennent dans le modèle par la rétroaction $w_\Delta = \Delta z_\Delta$. La matrice incertaine Δ regroupe toutes les incertitudes paramétriques et se caractérise par un ensemble de valeurs admissibles $\mathbf{\Delta}$. Les types d'ensembles d'incertitudes prévus dans RoMulOC et leurs structures sont détaillés dans la suite.

Les systèmes LTI (Linéaires invariants dans le temps) peuvent être indifféremment à temps continu ou discret. Dans le cas discret, la notation est $\delta[x(t)] = x(t+T)$ où T est la période d'échantillonnage. Pour les systèmes à temps continu la notation est $\delta[x(t)] = \dot{x}(t)$. Les matrices de la représentation d'état des systèmes se définissent comme suit :

$$\begin{array}{rl} \delta[x(t)] = & Ax(t) + B_\Delta w_\Delta(t) + B_w w(t) + B_u u(t) \\ z_\Delta(t) = & C_\Delta x(t) + D_{\Delta\Delta} w_\Delta(t) + D_{\Delta w} w(t) + D_{\Delta u} u(t) \\ z(t) = & C_z x(t) + D_{z\Delta} w_\Delta(t) + D_{zw} w(t) + D_{zu} u(t) \\ y(t) = & C_y x(t) + D_{y\Delta} w_\Delta(t) + D_{yv} v(t) + D_{yu} u(t) \end{array} \quad [1]$$

où $x \in \mathsf{R}^{\mathsf{n}}$, $w_\Delta \in \mathsf{R}^{\mathsf{m}_\Delta}$, $w \in \mathsf{R}^{\mathsf{m}_\mathsf{w}}$, $u \in \mathsf{R}^{\mathsf{m}_\mathsf{u}}$, $z_\Delta \in \mathsf{R}^{\mathsf{p}_\Delta}$, $z \in \mathsf{R}^{\mathsf{p}_\mathsf{z}}$ et $y \in \mathsf{R}^{\mathsf{p}_\mathsf{y}}$.

En supposant le système bien posé ($(1-D_{\Delta\Delta}\Delta)$ est non singulière pour toutes les incertitudes admissibles $\Delta \in \mathbf{\Delta}$), le système incertain $\Sigma(\Delta)$ est obtenu en appliquant le bouclage $w_\Delta = \Delta z_\Delta$. Sa représentation d'état est notée :

$$\begin{array}{rl} \delta[x(t)] = & A(\Delta)x(t)+ \ B_w(\Delta)w(t)+ \ B_u(\Delta)u(t) \\ z(t) = & C_z(\Delta)x(t)+D_{zw}(\Delta)w(t)+D_{zu}(\Delta)u(t) \\ y(t) = & C_y(\Delta)x(t)+D_{yw}(\Delta)w(t)+D_{yu}(\Delta)u(t) \end{array} \qquad [2]$$

2.2. *Systèmes polytopiques*

Les modèles polytopiques forment un cas particulier des modèles LFT pour lesquels l'incertitude entre de façon affine ($D_{\Delta\Delta} = 0$) et tels que $\mathbf{\Delta}$ est défini comme l'enveloppe convexe d'un nombre fini de sommets. Les propriétés mathématiques de ces systèmes ont conduit à des résultats d'analyse spécifiques impliquant de leur consacrer des fonctionnalités à part.

Pour la modélisation polytopique les signaux z_Δ/w_Δ sont inutiles et sont donc omis (ce qui revient à choisir $m_\Delta = p_\Delta = 0$ dans la suite). Les modèles polytopiques sont définis à la donnée de N sommets qui sont tous des systèmes LTI avec le même nombre de sorties/entrées et d'états. Chaque sommet ($\Sigma^{[i]}$) est défini par :

$$\begin{array}{rl} \delta[x(t)] = & A^{[i]}x(t)+ \ B_w^{[i]}w(t)+ \ B_u^{[i]}u(t) \\ z(t) = & C_z^{[i]}x(t)+D_{zw}^{[i]}w(t)+D_{zu}^{[i]}u(t) \\ y(t) = & C_y^{[i]}x(t)+D_{yw}^{[i]}w(t)+D_{yu}^{[i]}u(t) \end{array}$$

et le système incertain polytopique dont les sommets sont $\Sigma^{[1]}$, $\Sigma^{[2]}$, $\ldots\Sigma^{[N]}$ est décrit par l'ensemble :

$$\left\{ \Sigma(\Delta) \quad : \quad \Delta = \begin{pmatrix} \zeta_1 \geq 0 \\ \vdots \\ \zeta_N \geq 0 \end{pmatrix} \quad , \quad \sum_{i=1}^{N} \zeta_i = 1 \right\}$$

où les matrices du système $\Sigma(\Delta)$ sont telles que :

$$A(\Delta) = \sum_{i=1}^{N} \zeta_i A^{[i]} \quad , \quad B_w(\Delta) = \sum_{i=1}^{N} \zeta_i B_w^{[i]} \quad \ldots$$

Il faut noter que la modélisation polytopique comporte deux importants sous cas : les intervalles et les parallélotopes. Les modèles incertains de type intervalle se définissent à la donnée de deux systèmes LTI extrémaux $\bar{\Sigma}$ et $\underline{\Sigma}$ ayant les mêmes dimensions de vecteurs sorties/entrées et d'état. Les matrices incertaines du modèle $\Sigma(\Delta)$ sont telles que tous leurs éléments sont indépendants les uns des autres et bornés par les valeurs des éléments correspondants dans $\bar{\Sigma}$ et $\underline{\Sigma}$:

$$\underline{A}_{ij} \leq A_{ij}(\Delta) \leq \bar{A}_{ij} \quad \ldots$$

Cette modélisation est trivialement un sous cas de la modélisation polytopique et elle comprend 2^{N_I} sommets où N_I est le nombre d'éléments distincts entre $\bar{\Sigma}$ et $\underline{\Sigma}$.

Les modèles incertains parallélotopiques quant à eux se définissent à la donnée d'un modèle LTI central $\Sigma^{|0|}$ et d'un nombre fini N_P d'"axes" $\Sigma^{|i|}$ qui sont tous des modèles LTI de mêmes dimensions de vecteurs sorties/entrées et d'état. Les matrices du modèle incertain $\Sigma(\Delta)$ sont telles que :

$$A(\Delta) = A^{|0|} + \sum_{i=1}^{N_P} \delta_i A^{|i|} \quad \ldots$$

où chaque $|\delta_i| \leq 1$ est la distance au centre $\Sigma^{|0|}$ le long de l'axe $\Sigma^{|i|}$. La modélisation parallélotopique est trivialement un sous cas des modèles polytopiques avec 2^{N_P} sommets. La fonction `u2poly` de RoMulOC permet la conversion en modèle polytopique depuis les modèles intervalles et parallélotopiques.

2.3. *Manipulation de modèles dans RoMulOC*

De nombreuses fonctions de la boîte à outils sont disponibles pour la manipulation des modèles incertains décrits ci-dessus. Par exemple, la fonction `feedback` calcule le modèle obtenu par bouclage d'un contrôleur sur les sorties/entrées de mesure/commande (y/u) ; pour les modèles LFT, la fonction `certain` calcule le système LTI pour une valeur donnée de Δ ; la fonction `shape` calcule le modèle résultant de la mise en série d'un filtre dynamique sur les sorties de mesure de la performance z ou sur les entrées de perturbation w... Toutes les fonctions de définition et de manipulation des modèles sont précisément décrites dans le guide d'utilisation (Peaucelle, 2005).

Le présent manuscrit est consacré à la description des méthodes d'analyse robuste par formalisme LMI qui sont programmées dans la première version de RoMulOC. Pour cette raison, les signaux de mesure et de commande (y/u) sont omis par la suite. Les systèmes considérés sont soit en boucle ouverte ou bien obtenus par bouclage avec un contrôleur donné.

3. Formules LMI pour l'analyse de performance robuste

3.1. *Une approche LMI pour l'analyse des performances*

Avant de donner les détails des méthodologies employées pour obtenir des conditions robustes, cette première sous-section résume les types de performances considérées dans la boîte à outils pour les systèmes LTI sans incertitudes et en donne les formulations LMI.

Soit le système LTI

$$\begin{aligned} \delta[x(t)] = & \; Ax(t) + \; B_w w(t) \\ z(t) = & \; C_z x(t) + D_{zw} w(t) \end{aligned} \qquad [3]$$

où $x \in \mathsf{R}^{\mathsf{n}}$, $w \in \mathsf{R}^{\mathsf{m_w}}$ et $z \in \mathsf{R}^{\mathsf{p_z}}$. Dans le cadre de travail de la théorie de Lyapunov, cinq critères de performance sont considérés. Toutes les formules sont des réécritures de résultats publiés dans (Boyd *et al.,* 1994, Peaucelle, 2000).

• Stabilité. Soient les deux matrices

$$R_{sc} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \;,\; R_{sd} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad [4]$$

dont la première correspond au cas des systèmes à temps continu et la seconde aux systèmes à temps discret. Classiquement dans le cadre de la théorie de Lyapunov, la stabilité est équivalente à l'existence d'une solution définie positive $P > 0$ au problème LMI

$$\begin{bmatrix} 1 & A^T \end{bmatrix} R_s \otimes P \begin{bmatrix} 1 \\ A \end{bmatrix} < 0 \;. \qquad [5]$$

• Localisation des pôles. Les régions considérées sont des disques et des demi-plans. N'importe quelle région de ce type s'écrit sous la forme suivante

$$\begin{pmatrix} 1 & s^* \end{pmatrix} R \begin{pmatrix} 1 \\ s \end{pmatrix} \leq 0 \qquad [6]$$

où $R = R^* \in \mathsf{C}^{2\times 2}$ est une matrice Hemitienne (symétrique réelle dans le cas des régions symétriques par rapport à l'axe des réels) dont l'élément de la seconde ligne et colonne est non-négatif, $R_{22} \geq 0$. L'analyse de la localisation des pôles dans des régions de ce type permet de tester des bornes supérieures sur le temps de réponse (demi-plan $s + s^* \leq -\alpha$), la fréquence propre ($ss^* \leq \omega_n^2$) ou l'amortissement (demi-plan en dessous d'une droite inclinée de $\arccos \zeta$ par rapport à l'axe imaginaire). Avec le même formalisme que dans (Peaucelle *et al.,* 2000), la localisation des pôles est équivalente à l'existence d'une solution $P > 0$ à

$$\begin{bmatrix} 1 & A^T \end{bmatrix} R \otimes P \begin{bmatrix} 1 \\ A \end{bmatrix} < 0 \;. \qquad [7]$$

• Performance H_∞. Considérant les mêmes notations que précédemment la norme H_∞ de (2) est inférieure à un niveau γ_∞ si et seulement s'il existe une solution $P > 0$ à

$$\begin{aligned} & \begin{bmatrix} 1 & A^T \\ 0 & B_w^T \end{bmatrix} R_s \otimes P \begin{bmatrix} 1 & 0 \\ A & B_w \end{bmatrix} \\ & + \begin{bmatrix} C_z^T C_z & C_z^T D_{zw} \\ D_{zw}^T C_z & D_{zw}^T D_{zw} - \gamma_\infty^2 1 \end{bmatrix} < 0 \;. \end{aligned} \qquad [8]$$

Cette performance garantit que le gain L_2 du transfert $w \to z$ est inférieur à γ_∞ d'une part ; d'autre part que la stabilité asymptotique est préservée quelque soit une perturbation bornée ($\Omega^*(t)\Omega(t) \leq \gamma^2 1$) telle que $w(t) = \Omega(t)z(t)$; et enfin que la valeur singulière maximale de la matrice de transfert pour le couple de sorties/entrés z/w est inférieure à γ_∞ pour toutes les fréquences.

• Performance H_2. La norme H_2 de (2) est inférieure à un niveau γ_2 si et seulement s'il existe une solution $P > 0$ à

$$\begin{array}{c} \left[\begin{array}{cc} 1 & A^T \end{array} \right] R_s \otimes P \left[\begin{array}{c} 1 \\ A \end{array} \right] + C_z^T C_z < 0 \\ \text{Trace}(B_w^T P B_w + D_{zw}^T D_{zw}) \leq \gamma_2^2 \ . \end{array} \qquad [9]$$

Dans le cas des systèmes à temps continu, afin de garantir la finitude de la norme H_2, il est nécessaire d'ajouter $D_{zw} = 0$ à ces contraintes. Ce critère de performance assure que pour un bruit blanc Gaussien w, la variance de réponse z est bornée par γ_2 ; que pour une entrée scalaire w du type impulsion, l'énergie de la sortie z est inférieure à γ_2 ; et que pour une sortie scalaire la déviation de z est bornée en amplitude par γ_2 quelque soit une entrée w d'énergie unitaire.

• Performance "Impulse-to-peak". L'amplitude de la sortie $\|z(t)\|$ est bornée par γ_{i2p} pour toute entrée impulsionnelle w s'il existe une solution $P > 0$ à

$$\begin{array}{cc} \left[\begin{array}{cc} 1 & A^T \end{array} \right] R_s \otimes P \left[\begin{array}{c} 1 \\ A \end{array} \right] < 0 & B_w^T P B_w \leq \gamma_{i2p}^2 1 \\ C_z^T C_z \leq P & D_{zw}^T D_{zw} \leq \gamma_{i2p}^2 1 \ . \end{array} \qquad [10]$$

Toutes ces formulations LMI permettent de résoudre les problèmes sans incertitudes et deviennent inutilisables pour les systèmes incertains. Cela demanderait à rechercher une infinité de matrices $P(\Delta) > 0$ pour une infinité de contraintes. Pour chaque classe de modèles incertains, RoMulOC permet de générer automatiquement des formulations pessimistes, soit en adoptant le cadre de travail de la "stabilité quadratique" qui réduit le problème à la recherche d'une matrice unique $P(\Delta) = P$, soit en faisant appel à des formulations nouvelles qui utilisent des fonctions de Lyapunov dépendant des paramètres $V(x, \Delta) = x^T P(\Delta) x$. Toutes ces méthodes sont décrites dans les sous-sections suivantes.

3.2. *Robustesse : Séparation Quadratique pour les modèles LFT*

Tous les résultats appliqués aux modèles LFT reposent sur le cadre de travail de la théorie de Lyapunov et de la Séparation Quadratique. Des fonctions de Lyapunov dépendant des paramètres et des séparateurs quadratiques sont introduit pour garantir robustement les performances. La fonction de Lyapunov et les séparateurs quadratiques jouent le rôle de variables additionnelles qui transforment le problème en un problème LMI qui peut-être résolu numériquement (nombre fini de variables et nombre fini de contraintes).

A ce stade de programmation de la boîte à outils, deux choix de fonctions de Lyapunov sont mis en oeuvre : une fonction de Lyapunov unique pour toutes les incertitudes $P(\Delta) = P$ et une fonction de Lyapunov dite quadratique-LFT de la forme

$$P(\Delta) = \begin{bmatrix} 1 \\ \Delta_c \end{bmatrix}^T P \begin{bmatrix} 1 \\ \Delta_c \end{bmatrix} \; : \; \Delta_c = \Delta(1 - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \,. \qquad [11]$$

Ce dernier type de fonction de Lyapunov dépendant des paramètres a été introduit dans (Iwasaki *et al.,* 2001, Peaucelle *et al.,* 2001a) et manipulé par des techniques similaires. Dans RoMulOC, la programmation des LMI correspond à la démarche de (Iwasaki *et al.,* 2001) et a l'avantage vis-à-vis des résultats de (Peaucelle *et al.,* 2001a) de présenter le même niveau de pessimisme tout en permettant l'écriture de conditions LMI de plus faibles dimensions et contenant moins de variables de décision. La méthodologie est maintenant résumée. Elle consiste en la réécriture des conditions (5-10) dans lesquelles toutes les matrices A, B_w ... P sont remplacées par leur formulation incertaine $A(\Delta)$, $B_w(\Delta)$, ... $P(\Delta)$, sous la forme générique suivante :

$$\begin{bmatrix} 1 \\ \Delta_{\mathcal{C}} \end{bmatrix}^T \mathcal{P}(P) \begin{bmatrix} 1 \\ \Delta_{\mathcal{C}} \end{bmatrix} < 0 \; : \; \Delta_{\mathcal{C}} = \Delta_k(1 - \mathcal{D}\Delta_k)^{-1}\mathcal{C} \,. \qquad [12]$$

où $\Delta_k = 1_k \otimes \Delta$ et k prend les valeurs 1, 2 ou 3 suivant les performances et les types de fonctions de Lyapunov considérées. Pour chacun des cas, les valeurs de k et les formules donnant $\mathcal{P}$, $\mathcal{C}$ et $\mathcal{D}$ sont listées dans (Peaucelle, 2000). En vue d'obtennir des conditions LMI en dimension finie, la séparation quadratique (aussi connue sous le nom de "full-block S-procedure" (Scherer, 2000)) est appliquée à la condition (12) pour aboutir sans pessimisme à la nouvelle contrainte :

$$\mathcal{P}(P) \leq \begin{bmatrix} \mathcal{C} & \mathcal{D} \\ 0 & 1 \end{bmatrix}^T \Theta \begin{bmatrix} \mathcal{C} & \mathcal{D} \\ 0 & 1 \end{bmatrix} \qquad [13]$$

où Θ doit satisfaire pour toutes les incertitudes $\Delta \in \mathbb{\Delta}$ l'inégalité suivante :

$$\begin{bmatrix} 1 \\ 1_k \otimes \Delta \end{bmatrix}^T \Theta \begin{bmatrix} 1 \\ 1_k \otimes \Delta \end{bmatrix} \leq 0 \qquad [14]$$

Bien que les conditions [13] et [14] soient des conditions convexes (LMI) en les variables de décision (les éléments de P et de Θ), il est nécessaire de proposer un test en dimension finie. Pour cela, le nombre infini de contraintes [14] doit être remplacé par un choix de structure sur Θ. Ce choix de structure particulière pour le séparateur quadratique fait que les conditions dérivées sont uniquement suffisantes (pessimistes) mais numériquement testables. Le choix de la structure du séparateur Θ peut se faire sans pessimisme pour de très rares cas particuliers. Les choix de séparateurs programmés dans RoMulOC conduisent donc à des conditions uniquement suffisantes de performance robuste. Trois classes d'incertitudes sont envisagées à ce stade de programmation de RoMulOC. Chaque classe est maintenant détaillée et y est associé un choix de séparateur.

• Incertitude $\{x, y, z\}$-dissipative scalaire réelle répétée $\delta 1_r$ connue pour appartenir à l'ensemble défini par :

$$x + 2y\delta + z\delta^2 \leq 0 \ , \ z \geq 0 \ .$$

Cette représentation permet de modéliser toute incertitude réelle appartenant à un intervalle, éventuellement infini. Des exemples classiques de cette classe d'incertitudes sont les incertitudes bornées en norme $|\delta| \leq \bar{\delta}$ et les incertitudes positives réelles $\delta \geq 0$. Sur la base des DG-scaling (Iwasaki *et al.,* 1998), les séparateurs suivants sont programmés :

$$\begin{bmatrix} 1 \\ \delta 1_r \end{bmatrix}^T \begin{bmatrix} xD & yD+G \\ yD-G & zD \end{bmatrix} \begin{bmatrix} 1 \\ \delta 1_r \end{bmatrix} \leq 0 \qquad [15]$$

où $D = D^T > 0$ et $G^T = -G$.

• Incertitude complexe ou réelle $\{X, Y, Z\}$-dissipative bloc-plein, répétée $1_r \otimes \Delta$ connue pour appartenir à l'ensemble défini par une inégalité matricielle de la forme :

$$X + Y\Delta + \Delta^* Y^* + \Delta^* Z \Delta \leq 0 \ , \ Z \geq 0.$$

A nouveau, cette modélisation inclue les incertitudes bornées en norme $\Delta^*\Delta \leq \bar{\delta}^2 1$ et les opérateurs passifs $\Delta + \Delta^* \geq 0$. Sur la base des D-scalings, les séparateurs suivants sont programmés :

$$\begin{bmatrix} 1 \\ 1_r \otimes \Delta \end{bmatrix}^* \begin{bmatrix} D \otimes X & D \otimes Y \\ D \otimes Y^* & D \otimes Z \end{bmatrix} \begin{bmatrix} 1 \\ 1_r \otimes \Delta \end{bmatrix} \leq 0 \qquad [16]$$

où $D = D^* > 0$.

• Incertitude polytopique bloc-diagonale $\Delta_{pol} = \mathrm{diag}(\Delta_1, \ldots, \Delta_q)$ définie comme l'enveloppe convexe de N sommets par la formule suivante :

$$\Delta_{pol} = \sum_{i=1}^{N} \zeta_i \mathrm{diag}(\Delta_1^{[i]}, \ldots, \Delta_q^{[i]}) = \sum_{i=1}^{N} \zeta_i \Delta_{pol}^{[i]}$$

où $\zeta_i \geq 0$ et $\sum \zeta_i = 1$. A nouveau, comme pour les modèles incertains polytopiques, RoMulOC permet de définir directement des sous-classes d'incertitudes intervalles ou parallélotopiques. Pour toutes ces incertitudes polytopiques les séparateurs se basent sur la formulation des "vertex separators" (Iwasaki *et al.,* 1998), contraints par N inégalités testées sur les sommets :

$$\begin{bmatrix} 1 \\ 1_r \otimes \Delta_{pol}^{[i]} \end{bmatrix}^* \begin{bmatrix} \Theta_1 & \Theta_2 \\ \Theta_2^* & \Theta_3 \end{bmatrix} \begin{bmatrix} 1 \\ 1_r \otimes \Delta_{pol}^{[i]} \end{bmatrix} \leq 0$$

avec la contrainte supplémentaire que les blocs de la diagonale de Θ_3 (découpage en blocs en accord avec la subdivision en $\Delta_1, \ldots, \Delta_q$ de Δ_{pol}) doivent être semi-définis positifs, $\Theta_{3ii} \geq 0$.

• Toute incertitude bloc-diagonale dont les éléments sont des incertitudes appartenant à n'importe quelle de ces trois classes, est admise par RoMulOC. Les séparateurs sont alors construits comme une combinaison des séparateurs adaptés à chaque bloc.

Il faut noter que les incertitudes scalaires réelles $\{x, y, z\}$-dissipatives, lorsqu'elles décrivent un intervalle, peuvent tout aussi bien être décrites à l'aide de la modélisation polytopique. En adoptant cette dernière modélisation (et donc en faisant appel à des "vertex-separators") conduit à des conditions LMI de dimensions plus grandes en termes de nombre de variables mais dont on a prouvé leur plus faible pessimisme (Iwasaki *et al.,* 2001). Cette remarque est illustrée dans la dernière section de l'article sur un exemple numérique.

3.3. *Robustesse : validation aux sommets pour les modèles polytopiques*

Le cas des modèles incertains polytopiques définis en section 2.2 est maintenant considéré. Premièrement, il est possible de montrer que tout modèle de ce type peut toujours être réécrit sous une forme LFT avec une incertitude bloc-diagonale polytopique. Ces manipulations fastidieuses ne sont pas programmées à ce jour dans RoMulOC, d'autant qu'existent des outils d'analyse spécifiques pour les modèles polytopiques. Ceux-ci sont rapidement décrits ci-dessous.

Considérons en premier lieu le cas de la stabilité quadratique, pour laquelle une fonction de Lyapunov unique $P(\Delta) = P$ est recherchée. Dans cette situation il est possible de montrer que les LMI (5-10) sont convexes vis-à-vis des paramètres ζ. Il est dès lors nécessaire et suffisant de tester les LMI sur tous les sommets du polytope pour assurer qu'elles sont satisfaites pour l'ensemble de leur enveloppe convexe.

Ce résultat très connu a été étendu dans (Peaucelle *et al.,* 2000) à la recherche de fonctions de Lyapunov dépendant de façon polytopique des paramètres

$$P(\Delta) = \sum_{i=1}^{N} \zeta_i P^{[i]} \quad . \qquad [17]$$

Le résultat repose sur l'utilisation à rebours du lemme d'élimination (Skelton *et al.,* 1998), il crée des nouvelles variables de relaxation ("slack variables" (de Oliveira *et al.,* 1999)) et conduit à des nouvelles formulations connues sous le nom de "dilated LMIs" (Ebihara *et al.,* 2002). Les nouvelles conditions ainsi produites sont prouvées être toujours moins pessimistes que la stabilité quadratique. La condition obtenue pour la localisation de pôles est donnée pour illustration : trouver une matrice unique F et N matrices $P^{[i]}$ telles que pour les N sommets les LMI suivantes sont satisfaites

$$R \otimes P^{[i]} + F \left[\begin{array}{cc} A^{[i]} & -1 \end{array} \right] + \left[\begin{array}{c} A^{[i]T} \\ -1 \end{array} \right] F^T < 0 \quad .$$

Pour toutes les autres performances, les résulats sont sensiblement identiques et sont donc omis pour privilégier des exemples d'utilisation de RoMulOC.

4. Exemples

L'exemple considéré est pris de (Geromel *et al.,* 2002) et (Xie *et al.,* 2004). Il est utilisé pour illustrer l'analyse H_2 robuste. Sans entrer dans les détails de la syntaxe de RoMulOC, la façon de résoudre le problème est décrite pas à pas.

```
>> sys=ssmodel('Geromel et al. 2002')
name: Geromel et al. 2002
empty SSMODEL
```

ROMULOC a recours à des objets MATLAB notés `ssmodel` pour définir les systèmes. Ceux-ci ne sont pas ceux des boites à outils commerciales mais emprunte fortement aux objets `ss` de la Control Toolbox. Dans la ligne de commande ci-dessus un objet `ssmodel` vide a été déclaré. Lui a été affecté une étiquette qui est utilisée uniquement pour avoir un affichage convivial. Pour affecter des données à la variable `sys` il est possible d'employer les commandes suivantes

```
>> sys.A=[0.9 0.1; 0.01, 0.9];
>> sys.Bw=[ 1 0 0 ; 0 1 0];
>> sys.Cy=[1 0;1 1];
>> sys.Dyw=[0 0 1.414;0 0 0];
>> sys.T=1
name: Geromel et al. 2002
              n=2    mw=3
 n=2    dx  =   A*x +  Bw*w
py=2     y  =  Cy*x + Dyw*w
discrete time  ( dx : advance operator ) period T=1
```

A ce stade `sys` est un système LTI à temps-discret d'ordre 2, avec 3 entrées de perturbation (vecteur w) et 2 sorties de mesure (vecteur y). Dans (Geromel *et al.,* 2002) et (Xie *et al.,* 2004), le système est supposé mal connu du fait de deux incertitudes paramétriques scalaires dans A définies par :

$$A(\Delta) = A + \delta_1 \begin{bmatrix} 0 & 0.06 \\ 0 & 0 \end{bmatrix} + \delta_2 \begin{bmatrix} 0 & 0 \\ 0.05 & 0 \end{bmatrix}$$

où les incertitudes sont bornées $|\delta_i| \leq 1$. Ce système peut être vu comme un système parallélotopique. Les incertitudes δ_i définissent des écarts autour du système central `sys` le long de deux axes définis par des systèmes de même dimension. Commençons par définir les `axes` comme des systèmes vides de données de mêmes dimensions que le système `sys`

```
>> axes=ssmodel(0,sys)
              n=2    mw=3
 n=2    dx  =
```

```
py=2     y  =
discrete time  ( dx : advance operator ) period T=1
```

Cette variable est maintenant utilisée pour construire un vecteur de systèmes de dimensions identiques. Chaque élément de ce vecteur de système définit un axe.

```
>> axes(1).A=[0 0.06;0 0];
>> axes(2).A=[0 0;0.05 0]
Array of 2 systems
               n=2    mw=3
 n=2    dx  =   A*x
py=2     y  =
discrete time  ( dx : advance operator ) period T=1
```

A la donnée de `sys` et `axes`, le système incertain parallélotopique est déclaré dans RoMulOC comme suit

```
>> usysP =uparal(sys,axes)
Uncertain model : parallélotope 2 param
-------- WITH --------
               n=2    mw=3
 n=2    dx  =   A*x +  Bw*w
py=2     y  =  Cy*x + Dyw*w
discrete time  ( dx : advance operator ) period T=1
```

Dans (Xie *et al.,* 2004) un filtre dynamique de sortie est proposé avec comme objectif l'estimation d'une variable d'état. La performance souhaitée pour ce filtre est du type H_2 calculée sur le transfert de w à z où z est un signal de performance issu du filtre suivant :

```
>> filter=ssmodel('filter Xie et al.2004');
>> filter.A=[0.0705 0.0263;1.2779 0.5492];
>> filter.Bu=[0.9114 0;-0.9972 0];
>> filter.Cz=[-1.2885 -0.2382];
>> filter.Dzu=[0 1];
>> filter.T=1
name: filter Xie et al.2004
               n=2    mu=2
 n=2    dx  =   A*x +  Bu*u
pz=1     z  =  Cz*x + Dzu*u
discrete time  ( dx : advance operator ) period T=1
```

Le système global incluant le filtre est obtenu en mettant en série le système `sys` et le filtre `filter`. Cette opération correspond à la multiplication classique des systèmes et est définie comme telle dans RoMulOC :

```
>> filteredP=filter*usysP;
>> filteredP.name = 'Filtered system'
Uncertain model : parallélotope 2 param
-------- WITH --------
name: Filtered system
               n=4    mw=3
 n=4    dx  =   A*x +  Bw*w
pz=1     z  =  Cz*x
discrete time  ( dx : advance operator ) period T=1
```

Les méthodes d'analyse robuste de la boîte à outils sont programmées pour les systèmes polytopiques et non pas pour les sous-cas de systèmes parallélotopiques. Pour cela il faut recalculer à partir de la donnée du centre et des axes du parallélotope les sommets du polytope correspondant. Ce calcul fastidieux est fait par la commande suivante :

```
>> filteredP=u2poly(filteredP)
Uncertain model : polytope 4 vertices
-------- WITH --------
name: Filtered system
               n=4    mw=3
 n=4    dx  =   A*x +  Bw*w
pz=1     z  =  Cz*x
discrete time  ( dx : advance operator ) period T=1
```

A ce stade le système incertain est défini. Les fonctions de RoMulOC vont maintenant être utilisées pour évaluer le coût H_2 robuste. Ceci se fait en trois étapes. Premièrement, on définit le type de problème à résoudre (dans la version actuelle seule l'analyse est programmée) ainsi que le type de résultat théorique à appliquer.

```
>> quiz=ctrpb
 CHOOSE A CONTROL PROBLEM
(a) Analysis
(b) State-Feedback design
(c) Full-Order Dynamic Output-Feedback
choice > a
 CHOOSE A TYPE OF LYAPUNOV FUNCTION
(a) Unique over all uncertainties
(b) Quadratic w.r.t. del*(I-Ddd*del)^-1*Cd
(c) Polytopic
choice > a

control problem: ANALYSIS
Lyapunov function: UNIQUE (quadratic stability)
No specified performance
```

En second, la variable quiz est modifiée pour spécifier la performance à tester (H_2) et le système (filteredP).

```
>> quiz=h2(quiz,filteredP)
control problem: ANALYSIS
Lyapunov function: UNIQUE (quadratic stability)
Specified performances / systems:
# minimize H2 / Filtered system
```

A ce stade, le problème LMI à résoudre a été construit au format YALMIP et stocké dans la variable quiz. Le problème d'analyse est résolu lors de la troisième étape en faisant appel à n'importe quelle option (sdpsettings) permise par YALMIP.

```
>> solvesdp(quiz ,sdpsettings('solver','sdpt3'))
 num. of constraints = 11
 dim. of sdp    var  = 20,   num. of sdp  blk  =  5
 dim. of linear var  =  4
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 it  pstep dstep p_infeas d_infeas  gap     mean(obj)    cputime
-------------------------------------------------------------------
 0  0.000 0.000 4.4e+01 2.0e+00  2.8e+02 -4.205273e+01  0.0 spchol  1  1
....
 14  0.998 1.000 2.3e-09 2.1e-16  6.9e-07 -2.212779e+01  1.8
 Stop: max(relative gap, infeasibilities) < 1.00e-07
...
-------------------------------------------------------------------
H2 norm = 4.70402 guaranteed for all uncertainties
ans =
    4.7040
```

Il faut remarquer que l'optimisation est faite sur cet exemple en utilisant le solveur SDPT3 et le résultat est satisfaisant. Pour autant, au stade actuel des développements algorithmiques, les résultats ne seront pas nécessairement satisfaisants pour tout problème et tout solveur. Un aspect important de YALMIP (et par conséquence de RoMulOC) est de permettre l'utilisation de plusieurs solveurs sans difficultés. Par exemple, pour le problème considéré l'optimisation à l'aide du solveur SeDuMi donne :

```
>> solvesdp( quiz , sdpsettings('solver','sedumi') )
SeDuMi 1.05R5 by Jos F. Sturm, 1998, 2001-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 11, order n = 25, dim = 85, blocks = 6
nnz(A) = 238 + 0, nnz(ADA) = 111, nnz(L) = 61
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg
  0 :            1.13E+01 0.000
```

```
...
  20 :  -2.21E+01 1.26E-11 0.000 0.0759 0.9900 0.9900   1.00  2  2
iter seconds digits       c*x               b*y
 20      0.8   Inf -2.2127786147e+01 -2.2127786146e+01
|Ax-b| =   6.6e-10, [Ay-c]_+ =   4.1E-11, |x|=  4.6e+02, |y|=  2.8e+01
Max-norms: ||b||=1, ||c|| = 2.577000e+00,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 940.335.

Feasibility is not determined
Worst constraint residual is -4.14989e-11 < 0
4.70402 may be a guaranteed H2 norm
```

Les deux résultats sont très proches l'un de l'autre à la différence que l'algorithme de SeDuMi s'est arrêté sur un point proche de l'optimum mais pas strictement réalisable (ne satisfaisant pas strictement les contraintes LMI). La distance à la réalisabilité, en termes de valeur maximale des valeurs propres, est négligeable `4.14989e-11`.

Le résultat obtenu par l'approche "stabilité quadratique" est que la norme H_2 du système est inférieure à 4.70402 quelles que soient les incertitudes. La même procédure est maintenant appliquée pour tester la seconde méthode à base de fonctions de Lyapunov polytopiques.

```
>> quiz=ctrpb;
 CHOOSE A CONTROL PROBLEM
(a) Analysis
(b) State-Feedback design
(c) Full-Order Dynamic Output-Feedback
choice > a
 CHOOSE A TYPE OF LYAPUNOV FUNCTION
(a) Unique over all uncertainties
(b) Quadratic w.r.t. del*(I-Ddd*del)^-1*Cd
(c) Polytopic
choice > c

>> quiz=h2(quiz,filteredP)

control problem: ANALYSIS
Lyapunov function: POLYTOPIC
Specified performances / systems:
# minimize H2 / Filtered system

>> solvesdp(quiz,sdpsettings('solver','sdpt3'))
 num. of constraints = 73
 dim. of sdp    var  = 36,   num. of sdp  blk  =  5
 dim. of linear var  =  4
*******************************************************************
```

```
   SDPT3: Infeasible path-following algorithms
*******************************************************************
it  pstep dstep p_infeas d_infeas  gap     mean(obj)    cputime
-------------------------------------------------------------------
 0  0.000 0.000 1.4e+02 2.5e+00  1.1e+03 -8.410546e+01  0.1 spchol  2  2
...
 13  0.355 1.000 9.3e-07 1.5e-13  5.4e-06 -1.539477e+01  2.4
 Stop: relative gap < infeasibility.

H2 norm = 3.92362 guaranteed for all uncertainties
```

Comme prouvé théoriquement, l'utilisation des fonctions de Lyapunov dépendant des paramètres conduit à des résultats moins pessimistes que le cadre de travail de la stabilité quadratique. Le pessimisme est réduit en contrepartie d'une augmentation de la taille du problème LMI. Dans le cas présent, la taille des contraintes LMI croît de 11 à 73 et le nombre de variables de décision de 20 à 36. Dans le même temps le temps de calcul (avec un ordinateur SunBlade 150) passe de 1.8s à 2.4s. En termes de pessimisme, la borne supérieure sur le coût H_2 robuste est réduite de 4.70402 à 3.92362.

Pour analyser plus finement ces résultats, il est possible de calculer la norme H_2 du système pour certaines valeurs des incertitudes. Cela donnera donc des bornes inférieures sur le coût H_2 robuste. En premier, il est possible de tester le système dit nominal (il se définit par défaut comme le centre du polytope, c'est à dire le système obtenu pour $\zeta_i = 1/N$) :

```
>> norm(filteredP,2)
Nominal
ans =
    2.9636
```

En second, il est possible de tester sans difficulté n'importe quel sommet du polytope, par exemple le quatrième :

```
>> norm(filteredP(4),2)
System is stable
ans =
    3.9236
```

La norme H_2 du système calculée pour ce sommet est égale à la borne supérieure calculée par les fonctions de Lyapunov dépendant des paramètres. Pour cet exemple on peut donc conclure que l'optimisation LMI a donné sans pessimisme le coût H_2 robuste et que le quatrième sommet correspond au pire des cas. Le même résultat a également été obtenu par maillage des incertitudes dans (Xie *et al.,* 2004).

Pour comparaison, les méthodes basées sur une modélisation LFT sont maintenant testées. Comme mentionné précédemment, tout modèle polytopique peut se réécrire sous forme LFT. Pour l'exemple considéré, cela peut-être fait en écrivant

$$A(\Delta) = A + \underbrace{\begin{bmatrix} 0.06 & 0 \\ 0 & 0.05 \end{bmatrix}}_{B_\Delta} \underbrace{\begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}}_{\Delta} \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{C_\Delta}$$

```
>> sys.Bd=[0.06 0; 0, 0.05];
>> sys.Cd=[ 0 1 ; 1 0]
name: Geromel et al. 2002
               n=2    md=2     mw=3
 n=2    dx  =   A*x +  Bd*wd +  Bw*w
pd=2    zd  =  Cd*x
py=2     y  =  Cy*x          + Dyw*w
discrete time  ( dx : advance operator ) period T=1
```

et les incertitudes peuvent être définies au choix comme des intervalles (donc des polytopes)

```
>> u11 = uinter( -1 , 1 , 'delta1' );
>> u12 = uinter( -1 , 1 , 'delta2' );
>> u1 = diag( u11 , u12 )
diagonal structured uncertainty
size: 2x2 | nb blocks: 2 | independent blocks: 2
 wd = diag( #1 #2 ) * zd
index   size    constraint             name
#1      1x1     interval 1 param       delta1
#2      1x1     interval 1 param       delta2
>> u1 = u2poly( u1 )
diagonal structured uncertainty
size: 2x2 | nb blocks: 2 | independent blocks: 2
 wd = diag( #3 #4 ) * zd
index   size    constraint             name
#3      1x1     polytope 2 vertices    delta1
#4      1x1     polytope 2 vertices    delta2
```

ou comme des incertitudes bornées en norme (donc dissipatives)

```
>> u21 = unb( 1 , 1 , 1 , 'delta1' );
>> u22 = unb( 1 , 1 , 1 , 'delta2' );
>> u2 = diag( u21 , u22 );
```

En fonction du choix du modèle d'incertitudes, deux modèles de systèmes incertains sont obtenus

Δ	$P(\Delta) = P$	$P(\Delta)$ telle que (11)
Polytopique	4.70402	4.33218
Bornée en norme	4.70561	4.35878

Tableau 1. *Résultats pour la modélisation LFT*

```
>> usys1 = ussmodel( sys, u1 );
>> usys2 = ussmodel( sys, u2 );
```

et ils sont associés à deux modèles de systèmes incertains filtrés

```
>> filtered1 = filter*usys1;
>> filtered2 = filter*usys2
Uncertain model : LFT
-------- WITH --------
              n=4    md=2     mw=3
 n=4    dx  =   A*x +  Bd*wd +  Bw*w
pd=2    zd  =  Cd*x
pz=1     z  =  Cz*x
discrete time  ( dx : advance operator ) period T=1
-------- AND  --------
diagonal structured uncertainty
size: 2x2 | nb blocks: 2 | independent blocks: 2
 wd = diag( #5 #6 ) * zd
index   size    constraint              name
#5      1x1     norm-bounded by 1       delta1
#6      1x1     norm-bounded by 1       delta2
```

Pour chacun de ces modèles on définit des problèmes d'analyse H_2 robuste en suivant la procédure en trois étapes précédemment décrite. Les méthodes par fonctions de Lyapunov dépendant des paramètres sont sélectionnées en prenant le choix `b` car il s'agit de modèles LFT. Le Tableau 1 résume les résultats retournés par RoMulOC pour les quatre problèmes. Pour l'exemple considéré, les modèles LFT et les méthodes associées donnent des résultats plus pessimistes que ce qui était obtenu par les méthodes associées aux modèles polytopiques. De plus, on constate que les DG-scalings sont plus pessimistes que les "vertex separators". Les améliorations dues à l'utilisation de fonctions de Lyapunov dépendant des paramètres se confirment également.

5. Conclusions et prospectives

La première version de RoMulOC présentée dans cet article est une base pour de futurs développements. A ce stade, elle illustre les potentialités d'un outil qui combine l'emploi de résultats de recherche contemporains et une plate-forme versatile de résolution des LMI. Beaucoup d'autres développements sont en cours. Parmi les évolutions prévues, voici les principales.

– Pour la modélisation incertaine : programmation d'une interface avec la "LFR toolbox" (Magni, 2005) ; évolution des incertitudes pour tenir compte de variations temporelles ; extension aux systèmes singuliers (implicites).

– Pour l'analyse robuste : inclure les résultats récents de (Oliveira *et al.,* 2005, Chesi *et al.,* 2003, Peaucelle *et al.,* 2006) pour les modèles polytopiques et les résultats de (Bliman, 2004, Peaucelle *et al.,* 2005) pour les modèles LFT.

– Pour la synthèse de correcteurs robustes multi-objectif : programmation des conditions pour les correcteurs par retour d'état et pour les correcteurs par retour de sortie dynamique du même ordre que le système. Ceci se fera à la fois dans le cadre de travail du "Lyapunov Shaping Paradigm" (Scherer *et al.,* 1997) et dans le cadre d'extension plus récentes (Arzelier *et al.,* 2000, Arzelier *et al.,* 2004).

Remerciements

Ce travail est soutenu par un "projet LAAS" intitulé *Outils Logiciels pour l'Optimisation en Commande et Evaluation de Performance*, `http ://www2.laas.fr/OLOCEP`.

6. Bibliographie

Arzelier D., Bernussou J., Peaucelle D., « Fonctions de Lyapunov dépendant des paramètres pour l'analyse et la synthèse robuste », *in* , J. Bernussou, , A. Oustaloup (eds), *Les commandes robustes et inégalités matricielles*, vol. Conception de commandes robustes of *Traité IC2 - Information - Commande - Communication*, Hermes Sciences, p. 189-228, 2002. (in French).

Arzelier D., Peaucelle D., « Robust Multi-Objective State-Feedback Control for Real Parametric Uncertainties via Parameter-Dependent Lyapunov Functions », *IFAC Symposium on Robust Control Design*, vol. 1, Prague, p. 213-218, June, 2000.

Arzelier D., Peaucelle D., « Multiobjective H_2/H_∞/impulse-to-peak synthesis : Application to the control of an aerospace launcher », *IFAC Symposium on Automatic Control in Aerospace*, St Petersburg, Russia, June, 2004.

Bliman P.-A., « A convex approach to robust stability for linear systems with uncertain scalar parameters », *SIAM J. Control and Optimization*, vol. 42, p. 2016-2042, 2004.

Boyd S., Ghaoui L. E., Feron E., Balakrishnan V., *Linear Matrix Inequalities in System and Control Theory*, SIAM Studies in Applied Mathematics, Philadelphia, 1994.

Chesi G., Garulli A., Tesi A., Vicino A., « Robust Stability of Polytopic Systems via Polynomially Parameter-Dependent Lyapunov Functions », *IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December, 2003.

de Oliveira M., Bernussou J., Geromel J., « A New Discrete-Time Stability Condition », *Systems & Control Letters*, vol. 37, n° 4, p. 261-265, July, 1999.

Ebihara Y., Hagiwara T., « Robust Controller Synthesis with Parameter-Dependent Lyapunov Variables : A Dilated LMI Approach », *IEEE Conference on Decision and Control*, Las Vegas, Nevada, p. 4179-4184, December, 2002.

Geromel J., de Oliveira M., Bernussou J., « Robust filtering of discrete-time linear systems with parameter dependent Lyapunov functions », *SIAM J. Control and Optimization*, vol. 41, p. 700-711, 2002.

Iwasaki T., Hara S., « Well-Posedness of Feedback Systems : Insights into Exact Robustness Analysis and Approximate Computations », *IEEE Trans. on Automat. Control*, vol. 43, n° 5, p. 619-630, 1998.

Iwasaki T., Shibata G., « LPV System Analysis via Quadratic Separator for Uncertain Implicit Systems », *IEEE Trans. on Automat. Control*, vol. 46, n° 8, p. 1195-1207, August, 2001.

Magni J., User Manual of the Linear Fractional Representation Toolbox Version 2.0, Technical report, ONERA - Systems Control and Flight Dynamics Depertment, October, 2005. URL : `www.cert.fr/dcsd/idco/perso/Biannic%` verb+/toolboxes.html+.

Oliveira R., Peres P., « Stability of Polytopes of Matrices via Affine Parameter-dependent Lyapunov Functions : Asymptotically exact LMI Conditions », *Linear Algebra and Its Applications*, vol. 405, p. 209-228, 2005.

Peaucelle D., Formulation Générique de Problèmes en Analyse et Commande Robuste par les Fonctions de Lyapunov Dépendant des Paramètres, PhD thesis, Université Toulouse III - Paul Sabatier, France, July, 2000.

Peaucelle D., RoMulOC : a YALMIP-Matlab based Robust Multi-Objective Control Toolbox, Technical Report n° 05377, LAAS-CNRS, Toulouse, June, 2005.

Peaucelle D., Arzelier D., « Robust Performance Analysis with LMI-Based Methods for Real Parametric Uncertainty via Parameter-Dependent Lyapunov Functions », *IEEE Trans. on Automat. Control*, vol. 46, n° 4, p. 624-630, April, 2001a.

Peaucelle D., Arzelier D., « Une formulation générique pour l'analyse et la synthèse robustes en stabilité et performance », *Journal Europeen des Systèmes Automatisés (APII-JESA)*, vol. 35, n° 1-2, p. 49-67, 2001b. Numéro spécial *Commande Robuste des Systèmes Multivariables*.

Peaucelle D., Arzelier D., Bachelier O., Bernussou J., « A New Robust D-Stability Condition for Real Convex Polytopic Uncertainty », *Systems & Control Letters*, vol. 40, n° 1, p. 21-30, May, 2000.

Peaucelle D., Ebihara Y., Arzelier D., Tomomichi H., « General polynomial parameter-dependent Lyapunov functions for polytopic uncertain systems », *International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, July, 2006. Submitted.

Peaucelle D., Henrion D., Arzelier D., « Quadratic Separation for Feedback Connection of an Uncertain Matrix and an Implicit Linear Transformation », *16th IFAC World Congress*, Prague, Czech Republic, July, 2005.

Scherer C., *Advances in Linear Matrix Inequality Methods in Control*, Advances in Design and Control, SIAM, chapter 10 Robust Mixed Control and Linear Parameter-Varying Control with Full Block Scallings, p. 187-207, 2000. edited by L. El Ghaoui and S.-I. Niculescu.

Scherer C., Gahinet P., Chilali M., « Multiobjective Output-Feedback Control via LMI optimization », *IEEE Trans. on Automat. Control*, vol. 42, n° 7, p. 896-911, July, 1997.

Scorletti G., Ghaoui L. E., « Improved LMI Conditions for Gain Scheduling and Related Control Problems », *Int. J. of Robust and Nonlinear Control*, vol. 8, p. 845-877, 1998.

Skelton R., Iwazaki T., Grigoriadis K., *A unified Approach to Linear Control Design*, Taylor and Francis series in Systems and Control, 1998.

Xie L., Lilei L., Zhang D., Zhang H., « Improved H_2 and H_∞ filtering for uncertain discrete-time systems », *Automatica*, vol. 40, p. 873-880, 2004.