

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE MENTOURI DE CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE**

MEMOIRE

Présenté en vue de l'obtention du diplôme
De Magistère en Electronique

OPTION : **Contrôle**

THEME

Calcul et Robustification d'un Controleur

PID Multivariable par Approche LMI

Présenté Par :

HADEF Souhila

Devant le jury :

Président	: Mr Khaled Belarbi	Professeur	Univ. Constantine.
Encadreur	: Mr Salim Filali	Professeur	Univ. Constantine.
Examineurs:	Mr Abdelhak Bennis	Professeur	Univ. Constantine.
	Melle Fella Hachouf	Maître de conférence	Univ. Constantine.

Année 2007

*À toute ma famille.
À tous mes amis.*

SOMMAIRE

<i>Notions</i>	1
<i>CHAPITRE 1 Introduction générale</i>	5
<i>CHAPITRE 2 Optimisation LMI</i>	8
2.1 Introduction	8
2.2 Inégalités matricielles affines	8
2.2.1 Positivité	8
2.2.2 Inégalité matricielle affine ou linéaire	9
2.3 Stabilité de Lyapunov	9
2.3.1 Système non-linéaire	9
2.3.2 Système linéaire	10
2.3.3 Exemple de LMI	11
2.4 Problème d'optimisation sous contraintes LMI	12
2.4.1 Problème de Faisabilité	14
2.4.2 Problème de minimisation d'une fonction de coût linéaire ..	15
2.4.3 Problème de la valeur propre généralisée maximale	15
<i>CHAPITRE 3 Robustesse et performance</i>	16
3.1 Normes H_2 et H_∞ de matrices de transfert	16
3.1.1 Les gains principaux de matrices de transfert	16
3.1.2 Normes de matrices de transfert	16
3.1.2.1 La norme H_2	16
3.1.2.2 La norme H_∞	18
3.4 Critères de robustesse	19
3.4.1 Spécifications fréquentiels	19
3.4.2 Stabilité robuste	21
3.4.2.1 Marges de stabilité	22
3.4.2.2 Théorème de stabilité robuste de modèle incertain.	22

3.4.3	Performance robuste	24
3.4.3.1	Sensibilité en sortie	24
3.4.3.2	Modération de la commande	24
CHAPITRE 4 Théorie de la commande robuste		25
4.1	Introduction	25
4.2	Modélisation incertaine	26
4.3	Analyse et synthèse robuste	28
4.3.1	Analyse robuste	28
4.3.2	Synthèse robuste	29
4.3.2.1	La synthèse par retour de sortie statique	30
4.3.2.2	La synthèse H_∞ par retour de sortie statique	31
	Synthèses de correcteur H_∞	31
	Synthèse H_∞ par retour de sortie	33
	Existence d'un correcteur H_∞	37
CHAPITRE 5 Application de la méthode ILMI		39
5.1	Introduction	39
5.2	Formalisation du problème	39
5.3	Description de la méthode itérative ILMI	40
5.3.1	Initialisation de la matrice P	41
5.3.2	Résolution de problème SOF	43
5.7	Exemples de simulation	45
	<i>Exemple 1</i>	45
	<i>Cas SISO</i>	45
	<i>Cas MIMO</i>	46
	<i>Exemple 2</i> (synthèse H_∞)	47
	<i>Exemple 3</i>	50
	a) contrôleur PID	50
	b) contrôleur PI	53
5.8	Conclusion	55

<i>Conclusion générale</i>	56
----------------------------------	----

Annexes

<i>Annexe A Fonctions de sensibilité et Description des incertitudes.</i>	58
---	----

A.1 Fonctions de sensibilité	58
------------------------------	----

A.2 Description des incertitudes	59
----------------------------------	----

Annexe B Quelques rappels sur les matrices	61
---	-----------

Annexe C régulateurs PID et PI	63
---	-----------

C.1 Régulateur PID	63
--------------------------	----

C.2 Régulateur PI	66
-------------------------	----

<i>Références</i>	67
-------------------------	-----------

Notions

\mathbb{R}	corps des nombres réels
\mathbb{C}	corps des nombres complexes
\mathbb{C}^+	demi-plan complexe droit
\mathbb{C}^-	demi-plan complexe gauche
$\mathbb{R}^{n \times m}$	espace vectoriel de Hilbert des matrices réelles de dimension $n \times m$
$\mathbb{C}^{n \times m}$	espace vectoriel de Hilbert des matrices complexes de dimension $n \times m$
$\ F\ _\infty$	norme H_∞ de la matrice de transfert F
RH_∞	sous-espace de Hardy des matrices de transfert réelles rationnelles stables propres
S_n	sous-espace vectoriel des matrices symétriques de dimension n
S_n^+	cône convexe des matrices symétriques semidéfinies positives de dimension n
S_n^{++}	cône convexe des matrices symétriques définies positives de dimension n
\triangleq	égal par définition
s, p	variable de Laplace
$X_T(X^*)$	transposée (transposée conjuguée) de la matrice X
X_\perp	matrice vérifiant $XX_\perp = 0$ et $[X^T \quad X_\perp]$ de rang plein
BMI	inégalité matricielle bilinéaire (de l'anglais Bilinear Matrix Inequality)
LMI	inégalité matricielle linéaire (de l'anglais Linear Matrix Inequality)
LTI	linéaire stationnaire (de l'anglais Linear Time Invariant)
LFT	représentation linéaire fractionnaire (de l'anglais Linear Fractional transformation)

INTRODUCTION
GÉNÉRALE



INTRODUCTION GÉNÉRALE

La formulation LMI (Linear Matrix Inequality) est aujourd'hui un outil efficace pour la résolution de nombreux problèmes d'automatique, grâce aux avancées du début des années 90 [1]-[3] et des logiciels conviviaux tels que [3].

L'optimisation convexe sous contraintes LMIs a émergé comme un outil incontournable en automatique. Son émergence est en partie le résultat de la prise de conscience au sein des chercheurs en commande robuste de la place centrale de la complexité algorithmique pour la mise au point de méthodes d'analyse et/ou de commande des systèmes. Les problèmes d'optimisation convexe sous contrainte LMI ont le grand intérêt d'avoir des algorithmes de résolution efficace, largement disponibles dans les logiciels de calcul scientifiques généraux.

Un des premiers résultats fut l'obtention d'une approche unifiée et élégante pour la commande des systèmes Linéaires Temps Invariants (LTI). En effet, il est apparu que la très grande majorité des problèmes d'analyse et de commande de systèmes LTI pouvant être résolus efficacement se formulent comme des problèmes d'optimisation convexe sous contraintes LMI.

Pour un nombre significatif de problèmes d'analyse et de commande LTI, la formulation sous forme de problèmes d'optimisation LMI pouvait être (parfois avantageusement) remplacée par des formulations utilisant par exemple des équations de Lyapunov ou de Riccati. Par contre, contrairement à l'optimisation LMI, ces outils alternatifs n'ont pas permis de formuler un large spectre de problèmes sortant du cadre LTI « classique ».

L'approche H_∞ est une méthode d'optimisation et de synthèse de régulateurs qui s'est imposée jusqu'à prendre la tête de la commande robuste, puisqu'elle offre la plus proche et la plus explicite des formulations de la notion de robustesse.

La méthode de synthèse H_∞ est une méthode fréquentielle basée sur les principes des méthodes fréquentielles dites classiques. Les méthodes fréquentielles reposent sur l'hypothèse qu'un cahier des charges peut être traduit par un cahier des charges avec que des spécifications fréquentielles.

Le retour de sortie statique (SOF) joue un rôle très important dans la théorie du contrôle et les applications. Contrairement au cas du retour d'état statique (tâche inscrite dans la suite de nos recherches) un SOF gain qui stabilise le système n'est pas facile à déterminer.



Beaucoup de méthodes de conception basée sur des LMI pour la conception de SOF ont été proposées au cours de la dernière décennie [17]-[11]. Parmi ces méthodes, Une Inégalité Matricielle Linéaire basée sur un processus Itératif (ILMI) a été proposée par Cao et coll. [11] et plus tard employée pour résoudre certains problèmes de conception de contrôleur PID multivariables [13]-[18]. Dans ce contexte, une nouvelle variable a été introduite de telle sorte que la condition de stabilité devient suffisante. L'algorithme itératif dans [11] a essayé de trouver une séquence de variables telles que la condition suffisante est proche de la condition nécessaire et suffisante. Une idée similaire a été utilisée dans ce qu'on appelle la méthode LMI de substitution [11]. Les Deux méthodes ont introduit des matrices supplémentaires à l'étape actuelle avec les matrices dérivées (trouvées) à l'étape précédente. Avec des variables supplémentaires les dimensions de l'LMIs augmentent.

Dans ce travail nous reprenons une méthode où il est possible que les matrices dérivées à l'étape précédente peuvent être utilisées dans la prochaine directement sans introduire de variables supplémentaires ainsi les dimensions des LMIs ne doivent pas être augmenté. On s'intéresse justement à une méthode efficace pour obtenir des valeurs initiales convenables dans la procédure itérative, qui n'ont pas été à l'intention des approches existantes.

Pour cela, un algorithme pour résoudre le problème du contrôle à retour de sortie statique (SOF) est proposé dans ce mémoire, une approche ILMI a été développée pour construire cette méthode. La méthode de stabilisation par retour de sortie statique (SOF) a été étendu pour résoudre aussi le problème du retour de sortie statique cas H_∞ contrôle.

Les algorithmes mis au point sont également appliqués à la conception de contrôleur PID multivariables. Nous avons étendu ces algorithmes pour le cas d'un contrôleur PI ainsi la comparaison des deux algorithmes PI et PID à pu être faite.

La qualité de cette méthode c'est qu'elle évite l'introduction de variables supplémentaires, qui nous le savons diminuent les dimensions des LMIs si nous le comparons avec les résultats des travaux effectués dans les articles [10]-[11]. Cette méthode en particulier, a besoin de bonnes valeurs initiales, un autre algorithme pour calculer les valeurs initiales a été également mis au point.

Le mémoire est organisée comme suit :



Dans le deuxième chapitre nous illustrons quelques notions de base sur l'optimisation convexe, puis les définitions des problèmes d'optimisation sous contrainte LMI.

Le troisième chapitre propose une méthode de commande robuste par synthèse de critères explicite de robustesse vis à vis des incertitudes.

Le quatrième chapitre donne le contexte général de l'analyse et de la synthèse robuste par problèmes ouverts en théorie de la commande pour lesquels la théorie de Lyapunov a été un outil particulièrement intéressant pour définir des approches par l'optimisation.

Dans le cinquième chapitre, on a ré-écrit un Algorithme itérative basé sur l'approche LMI (ILMI) à retours de sortie statique. Cet algorithme a été utilisé pour résoudre le problème H_∞ à retours de sortie. Et enfin il est appliqué au calcul et à la robustification d'un contrôleur PI et PID multivariable. Des exemples numériques ont été faits pour montrer la validité de cet algorithme.

Et en termine notre travail par une conclusion générale.

*OPTIMISATION SOUS
CONTRAINTES LMI*



2.1 Introduction

En automatique, un problème d'optimisation peut correspondre à rechercher un correcteur qui satisfasse les spécifications d'un cahier des charges. Dans le cas général, la résolution par un algorithme est un problème "compliqué". Pourquoi ? Très grossièrement, à partir d'un point initial, les algorithmes efficaces disponibles recherchent un minimum local. Si la fonction admet plusieurs minima, le résultat va dépendre du point initial. Un cas très intéressant est celui où il n'existe qu'un seul minimum local (minimum local = minimum global) : indépendamment du point initial, le minimum global sera alors atteint. Cette propriété est obtenue dans le cas où la fonction est convexe et où l'ensemble des contraintes est convexe: on parle alors d'*optimisation convexe*. Un certain nombre de classes de problèmes d'optimisation convexe de dimension finie admettent des algorithmes de résolution efficaces. Parmi ceux-ci, se trouve l'*optimisation LMI*.

2.2 Inégalités matricielles affines

2.2.1 Positivité

On dit d'une matrice $A \in R^n$ qu'elle est positive et on note $A > 0$ si et seulement si la forme quadratique $x^T Ax$ est positive pour tout vecteur x .

$$\forall x \in R^n, x^T Ax > 0 \Rightarrow A > 0 \quad (2.1)$$

Cette définition se transpose évidemment au cas négatif. On peut toujours écrire une forme quadratique à partir d'une matrice symétrique. Ces matrices ont la particularité de n'avoir que des valeurs propres réelles. En optimisation, on utilise souvent le résultat suivant que, pour A symétrique, $x^T Ax$ est positif pour tout x si et seulement si toutes ses valeurs propres sont positives et on note $A \geq 0$.

On définit aussi la positivité stricte et on dit qu'une matrice est définie positive si toutes ses valeurs propres sont strictement positives. C'est équivalent à dire que la forme quadratique correspondante $x^T Ax$ est strictement positive dès que x est non nul.

Quelques propriétés :

- Soit λ un scalaire, $A - \lambda I > 0$ si et seulement si les valeurs propres de A sont strictement supérieures à λ .
- $P > 0 \Leftrightarrow -P < 0$; on peut donc toujours se ramener à un problème de positivité ou de négativité.

2.2.2 Inégalité matricielle affine ou linéaire

On appelle inégalité matricielle affine (linear matrix inequality) noté LMI le problème suivant : étant donné des matrices réelles, carrées symétriques F_i , $i = 0 \dots n$, existe-t-il un jeu de réels x_i , $i = 1 \dots n$ tels que $F_0 + x_1 F_1 + \dots + x_n F_n > 0$

Ou
$$F(x) = F_0 + \sum_{i=1}^n x_i F_i > 0 \tag{2.2}$$

avec $x = [x_1, \dots, x_n] \in R^n$ appelé vecteur de décision ou vecteur d'optimisation

F_i : sont n matrices symétriques données de $R^{n \times n}$.

Le succès des LMI vient du développement des méthodes dites du point intérieur (interior point methods) qui permettent de résoudre de manière efficace ces problèmes [2].

La mise en évidence de cette « démarche LMI » a ouvert des possibilités telles qu'il est devenu possible d'aborder des problèmes d'analyse et/ou de commande jusque la non envisageables. La mise en oeuvre de la « démarche LMI » sur ces nouveaux problèmes amener à des formulations sous forme de problèmes d'optimisation sous contraintes LMI *dépendant de paramètres* : dans l'expression (2.2), les matrices F_i et les variables de décision x sont maintenant des fonctions rationnelles de paramètres réels, que l'on notera θ et qui appartiennent à un ensemble compact P :

$$\left\{ x(\theta) : P \rightarrow R^n \left| F_0(\theta) + \sum_{i=1}^n x_i(\theta) F_i(\theta) \right. \right\} \tag{2.3}$$

Sous cette forme, malheureusement, on ne retrouve pas l'un des grands intérêts de l'optimisation LMI qui est l'existence d'algorithmes de résolution efficaces. Face à ce problème, l'approche généralement adoptée pour le problème d'Automatique particulier considéré consiste à essayer de définir l'ensemble des fonctions (2.3) ou un sous ensemble de (2.3) par une contrainte LMI indépendant de θ .

2.3 Stabilité de Lyapunov

2.3.1 Système non-linéaire

Soit un système libre de vecteur d'état x et d'équation d'état $\dot{x} = f(x)$. Soit x_0 un point stable candidat. Il doit alors vérifier la condition d'équilibre $f(x_0) = 0$. De plus, il faut aussi que ce point soit attractif, c'est-à-dire que les trajectoires de x convergent vers x_0 .



Définition (Stabilité au sens de Lyapunov)

x_0 est un point stable au sens de Lyapunov s'il existe une fonction scalaire $V(x)$ vérifiant les conditions suivantes :

- $V(x) > V(x_0)$ pour $x \neq x_0$
- $\frac{d}{dt}(V(x)) < 0$ pour $x \neq x_0$

Une telle fonction $V(x)$ est dite fonction d'énergie du système.

A partir d'une condition initiale x_i différente de x_0 , l'énergie interne du système va décroître jusqu'à atteindre son minimum qui correspond à l'unique point x_0 ; l'état du système tendra donc nécessairement vers x_0 .

Pour démontrer la stabilité par cette méthode, la difficulté réside dans le choix d'une bonne fonction d'énergie. Une classe de fonctions souvent utilisées sont les fonction quadratiques $V(x) = (x - x_0)^T Q (x - x_0)$ avec $Q = Q^T > 0$; on parle alors de stabilité quadratique.

Pour la fonction d'énergie choisie, il reste à démontrer que $\frac{d}{dt}(V(x)) = \frac{dV(x)}{dx} f(x) < 0$ pour tout x .

2.3.2. Système linéaire

Dans ce cas, $f(x) = Ax$. Le point d'équilibre candidat est $x = 0$. En choisissant $V(x) = x^T Q x$ avec $Q = Q^T > 0$, la condition de stabilité s'écrit alors $x^T (A^T Q + Q A)x < 0$ pour $x \neq 0$, ce qui s'écrit aussi $A^T Q + Q A < 0$ et qui signifie que toutes les valeurs propres (réelles) de la matrice symétrique $A^T Q + Q A$ sont strictement négatives. La stabilité quadratique peut ainsi se caractériser par un système de LMI dont l'inconnue est la matrice symétrique Q :

$$Q > 0 \quad \text{et} \quad A^T Q + Q A < 0 \quad (2.4)$$

Remarque :

- dans ce cas, cette stabilité (quadratique de Lyapunov) est équivalente à la stabilité au sens classique dont le critère est que la matrice A ait ses valeurs propres à partie réelle négative.
- Lorsqu'il y a plusieurs contraintes LMI $F_i(\zeta) > 0$, $i = 1 \dots n$, il est toujours possible de les réécrire sous une seule contrainte LMI :

$$F(\zeta) \triangleq \begin{bmatrix} F_1(\zeta) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & F_n(\zeta) \end{bmatrix} > 0 \quad (2.5)$$

Il n'y a donc pas de perte de généralité à définir les problèmes d'optimisation sous contrainte LMI avec une seule contrainte LMI.

2.3.3 Exemple de LMI

Les LMI ne se présentent pas directement sous la forme de l'inégalité présentée ci-dessus (2.2). Prenons un exemple classique de l'automatique : la stabilité au sens de Lyapunov pour un système linéaire $\dot{x}(t) = Ax(t)$, d'après leur théorie pour toute condition initiale $x(0)$ il existe une fonction $V(x)$ telle que pour $x \neq 0$, $V(x) > 0$ et $\dot{V}(x) < 0$.

On choisira la fonction de Lyapunov quadratique $x^T P x$ il s'agit de trouver une matrice réelle P de même dimension que A telle que :

$$\begin{cases} V(x) > 0 \\ \dot{V}(x) < 0 \end{cases} \Rightarrow \begin{cases} P = P^T > 0 \\ A^T P + P A < 0 \end{cases} \quad (2.6)$$

Qui se réécrit sous une seule inégalité :

$$\begin{bmatrix} P & 0 \\ 0 & -(A^T P + P A) \end{bmatrix} > 0 \quad \text{ou} \quad \begin{bmatrix} -P & 0 \\ 0 & A^T P + P A \end{bmatrix} < 0 \quad (2.7)$$

Ces deux inégalités représentent une caractérisation matricielle de la stabilité sous formulation LMI.

Considérons à titre d'exemple, le cas où A est une matrice 2×2 .

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (2.8)$$

La matrice P dépend alors de 3 paramètres x_i , $i=1, 2, 3$ et peut s'écrire :

$$P = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix} \quad (2.9)$$

La condition de positivité de P ($P > 0$) s'écrit :

$$P = x_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} > 0 \quad (2.10)$$

$$\Leftrightarrow F_1(x) = F_{10} + \sum_{i=1}^3 x_i F_{1i} > 0$$

L'inégalité de Lyapunov, elle se réécrit :

$$x_1 \begin{bmatrix} 2a_1 & a_2 \\ a_2 & 0 \end{bmatrix} + x_2 \begin{bmatrix} a_2 + a_3 & a_1 + a_4 \\ a_1 + a_4 & a_2 + a_3 \end{bmatrix} + x_3 \begin{bmatrix} 0 & a_3 \\ a_3 & 2a_4 \end{bmatrix} < 0 \quad (2.11)$$

$$\Leftrightarrow F_2(x) = F_{20} + \sum_{i=1}^3 x_i F_{2i} < 0$$

Les deux inégalités (2.10) et (2.11) sont de la forme standard d'une contrainte LMI, conclusion que l'inégalité de Lyapunov est une LMI exprimant la stabilité d'un système linéaire stationnaire.

2.4 Problème d'optimisation sous contraintes LMI

Un problème d'optimisation est défini de la manière suivante.

Définition (Problème d'optimisation de dimension finie)

Soit

$$R^m \rightarrow R$$

$$\zeta \mapsto f(\zeta)$$

Alors un problème d'optimisation s'écrit :

$$\min_{\zeta \in C} f(\zeta) \quad (2.12)$$

où

- $C \subset R^m$ est l'ensemble des contraintes;
- $\zeta = [\zeta_1 \dots \zeta_m]^T$ est le vecteur de variables de décision ou variables d'optimisation;
- la valeur de ζ pour laquelle le minimum est atteint est appelée ζ^* . Notation :

$$\zeta^* = \underset{\zeta \in C}{\operatorname{argmin}} f(\zeta)$$

- la fonction f est l'objectif ou la fonction de coût;
- Quand $C = R^m$, on parle de problème d'optimisation sans contrainte sinon de problème d'optimisation sous contraintes.



On dit que :

- $\zeta^* \in C$ est un optimum local s'il existe un voisinage ouvert U de ζ^* tel que, pour tout $\zeta \in U \cap C$, on a $f(\zeta^*) \leq f(\zeta)$;
- $\zeta^* \in C$ est un optimum global si, pour tout $\zeta \in C$, on a $f(\zeta^*) \leq f(\zeta)$.

Résoudre ce problème signifie trouver un optimum global (et la valeur de f en ce point).

On distingue des classes de problèmes d'optimisation en fonction des propriétés de f et de C . Deux classes de problèmes importantes sont celle des problèmes d'optimisation convexes et celle des problèmes d'optimisation quasi-convexes.

Définition

Un problème d'optimisation est dit convexe si

- C est un ensemble convexe: pour tout $\zeta_1 \in C$, pour tout $\zeta_2 \in C$, pour tout $\lambda \in [0,1]$, on a $(\lambda \zeta_1 + (1-\lambda) \zeta_2) \in C$;
- F est une fonction convexe: pour tout $\zeta_1 \in C$, pour tout $\zeta_2 \in C$, pour tout $\lambda \in [0,1]$, on a $f(\lambda \zeta_1 + (1-\lambda) \zeta_2) \leq \lambda f(\zeta_1) + (1-\lambda)f(\zeta_2)$.

Un problème d'optimisation est dit quasi-convexe si

- C est un ensemble convexe: pour tout $\zeta_1 \in C$, pour tout $\zeta_2 \in C$, pour tout $\lambda \in [0,1]$, on a $(\lambda \zeta_1 + (1-\lambda) \zeta_2) \in C$;
- F est une fonction quasi-convexe: pour tout $\zeta_1 \in C$, pour tout $\zeta_2 \in C$, pour tout $\lambda \in [0,1]$, on a $f(\lambda \zeta_1 + (1-\lambda) \zeta_2) \leq \text{Max} \{f(\zeta_1), f(\zeta_2)\}$.

Pour ces deux classes de problèmes d'optimisation, la fonction de coût a la propriété que tout minimum local est un minimum global sur l'ensemble de définition de la fonction. Ceci implique que tout optimum local est un optimum global. Cette propriété est importante car, pour résoudre les problèmes d'optimisation, des algorithmes de résolution numérique sont généralement utilisés. Ces algorithmes nécessitent un point d'initialisation dans C et permettent, à partir de ce point d'initialisation, de trouver un optimum local. Pour les problèmes d'optimisation convexes et quasi-convexes, ces algorithmes permettent donc de trouver un optimum global et donc de les résoudre.

Du fait qu'ils peuvent être résolus efficacement, les problèmes d'optimisation convexes et quasi-convexes sont donc intéressants. Des sous-problèmes importants sont les problèmes d'optimisation sous contraintes LMI.

Un problème d'optimisation sous contrainte s'écrit :

$$\min_{\zeta \in C} f(\zeta)$$

où

$$C = \{ \zeta \in R^m \mid \forall x \in R^n, x^T F(\zeta) x \geq 0 \} \quad \text{ou} \quad C = \{ \zeta \in R^m \mid \forall x \in R^n \setminus \{0\}, x^T F(\zeta) x > 0 \}$$

$$\text{Avec } F(\zeta) \triangleq F_0 + \sum_{i=1}^m \zeta_i F_i,$$

Le vecteur $\zeta \in R^m$ est le vecteur des variable de décision de la contrainte, et où les F_i sont m matrices symétrique données de $R^{n \times n}$, $i=0 \dots m$.

Propriété

Soit les deux ensembles :

$$C_1 = \{ \zeta \in R^m \mid F^1(\zeta) > 0 \} \quad \text{et} \quad C_2 = \{ \zeta \in R^m \mid F^2(\zeta) > 0 \}$$

Alors l'intersection est définie par :

$$C_1 \cap C_2 = \left\{ \zeta \in R^m \mid \begin{bmatrix} F(\zeta)_1 & 0 \\ 0 & F_2(\zeta) \end{bmatrix} > 0 \right\} \quad (2.13)$$

Cette propriété découle du fait que les valeurs propres d'une matrice diagonale par blocs sont constituées par les valeurs propres des matrices sur la diagonale.

Afin de rendre les solvers de LMI facilement utilisables pour les problèmes de l'automatique, des interfaces ont été développées permettant d'écrire les problèmes sous des formes matricielles simples [3]-[4].

Les trois problèmes classiques que ces outils résolvent sont :

- la faisabilité (ou l'existence),
- la minimisation d'une fonction linéaire,
- le problème de valeur propre généralisée : minimiser λ sous les contraintes.

2.4.1 Problème de Faisabilité

Tester s'il existe $\zeta \in C \subset R^m$ tel que $F(\zeta) > 0$, et si oui, déterminer un tel $\zeta \in R^m$:

$$\begin{aligned} &\text{trouver } \zeta \in R^m \\ &\text{tel que } F(\zeta) > 0. \end{aligned} \quad (2.14)$$

Le problème est faisable si $C \neq \emptyset$, existe un ensemble non vide des ζ vérifiant $F(\zeta) > 0$.

2.4.2 Problème de minimisation d'une fonction de coût linéaire

Tester s'il existe $\zeta \in R^m$ tel que $F(\zeta) > 0$, et si oui, déterminer $\zeta \in R^m$ tel que $F(\zeta) > 0$ et qui minimise $c^T \zeta$ où $c \in R^m$ est un vecteur donné :

$$\begin{aligned} \min_{\zeta \in R^m} \quad & c^T \zeta \\ \text{tel que} \quad & F(\zeta) > 0 \end{aligned} \tag{2.15}$$

2.4.3 Problème de minimisation de la valeur propre généralisée maximale

Tester s'il existe $\zeta \in R^m$ tel que $F(\zeta) > 0$ et $H(\zeta) > 0$ où $H(\zeta)$ est de la forme (2.2), et si oui, déterminer $\zeta \in R^m$ tel que $F(\zeta) > 0$ et $H(\zeta) > 0$ et qui minimise la valeur propre généralisée maximale $\lambda_{\max}(G(\zeta), F(\zeta))$ de $F(\zeta)$ et $G(\zeta)$ où $G(\zeta)$ est de la forme (2.2) et où $\lambda_{\max}(G(\zeta), F(\zeta))$ est la valeur minimale de λ pour laquelle $\lambda F(\zeta) - G(\zeta)$ est définie positive :

$$\begin{aligned} \min_{\zeta \in R^m} \quad & \lambda_{\max}(G(\zeta), F(\zeta)) \\ \text{tel que} \quad & F(\zeta) > 0 \text{ et } H(\zeta) > 0. \end{aligned} \tag{2.16}$$

Autre écriture du problème :

$$\begin{aligned} \min_{\zeta \in R^m} \quad & \lambda \\ \text{tel que} \quad & \lambda F(\zeta) - G(\zeta) > 0 \\ & F(\zeta) > 0 \text{ et } H(\zeta) > 0 \end{aligned} \tag{2.17}$$

*ROBUSTESSE ET
PERFORMANCE*



3.1 Normes H_2 et H_∞ de matrices de transfert

On suppose que G est une matrice de transfert stable et propre d'un système d'ordre minimal (observable et commandable).

3.1.1 Les gains principaux de matrices de transfert

Pour chaque valeur de ω , on peut décomposer la matrice de transfert $G(j\omega)$ en valeurs singulières, ordonnées dans le sens décroissant. On peut alors créer 2 fonctions réelles positives et continues :

$$\forall \omega > 0, \overline{\sigma}[G(j\omega)] = \sup_i \{\sigma_i[G(j\omega)]\} \quad (3.1)$$

$$\forall \omega > 0, \underline{\sigma}[G(j\omega)] = \inf_i \{\sigma_i[G(j\omega)]\} \quad (3.2)$$

Ces fonctions s'appellent les gains principaux (supérieur et inférieur) de la matrice de transfert.

Définition 3.1 (Les valeurs singulières de matrices)

Les valeurs singulières d'une matrice A , de dimension $m \times n$, à coefficients complexes et de rang r , sont les racines carrées non négatives des valeurs propres de $A^* A$, où A^* est la matrice adjointe, c'est-à-dire conjuguée et transposée de A .

$$\sigma = \sqrt{\lambda(A^* A)} \quad (3.3)$$

On peut énoncer quelques propriétés des valeurs singulières :

- Les n valeurs singulières de A sont réelles ;
- On peut ordonner ces valeurs singulières :

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_r \geq \dots \geq \sigma_n$$
- Si $r < n$, il y a $n - r$ valeurs singulières nulles. Donc, le rang de A est égal au nombre de valeurs singulières non nulles.

3.1.2 Normes de matrices de transfert

3.1.2.1 La norme H_2

Interprétation de la norme H_2

La norme H_2 pour un système LTI stable et strictement propre $G : \begin{cases} \dot{x} = Ax + Bw \\ y = Cx \end{cases}$



est l'énergie du signal de sortie lorsque qu'on place en entrée une impulsion de Dirac (c à d la réponse impulsionnelle de $g(t)$)

$$\|G\|_2^2 = \frac{1}{2\pi} \int \text{trace}(G(j\omega)^* G(j\omega)) d\omega \quad (3.4)$$

$$\|G\|_2^2 = \max_{W_1(t)=\delta} \|y\|_2 \quad (3.5)$$

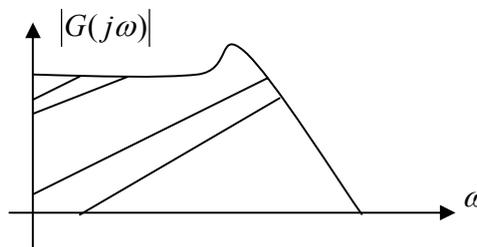


Figure 3.1 Interprétation physique de la norme H_2

Pour des systèmes SISO, la norme H_2 est le gain d'énergie de crête.

Calcul de la norme H_2 :

Le calcul algébrique de la norme H_2 utilise les grammien d'observabilité et de commandabilité du système. On montre que pour un système qui a pour représentation d'état les matrices (A, B, C, D) :

$$\|D + C(zI - AB)\|_2 = \sqrt{\text{trace}(D^T D + B^T L_o B)} = \sqrt{\text{trace}(DD^T + CL_c C^T)} \quad (3.6)$$

Où L_c et L_o sont respectivement les grammien d'observabilité et decommandabilité :

$$L_c = \int_0^{\infty} e^{At} B B^T e^{A^T t} dt \quad \text{et} \quad L_o = \int_0^{\infty} e^{A^T t} C^T C e^{At} dt \quad (3.7)$$

La norme H_2 s'obtient donc par la résolution d'un ensemble d'équations linéaires nécessaires à l'évaluation des grammien, puis par un calcul de trace. On l'évalue donc directement, c'est-à-dire sans itérations.



3.1.2.2 La norme H_∞

Interprétation de la norme H_∞

- **Cas monovariante**

Soit $G(p)$ une matrice de transfert $\|G\|_\infty$ est le gain maximal tel qu'il apparaît sur le diagramme de Bode de $G(p)$:

$$\|G(j\omega)\|_\infty = \sup_{\omega} \{|G(j\omega)|\} \quad (3.8)$$

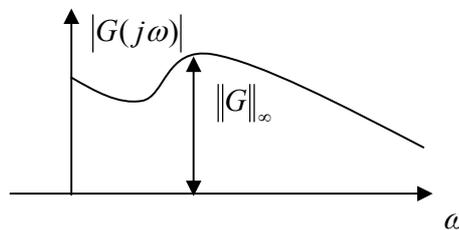


Figure 3.2 Interprétation physique de la norme H_∞

Une propriété fondamentale que vérifie la norme $\|\cdot\|_\infty$ c'est la propriété suivante

$$\|F.G\|_\infty \leq \|F\|_\infty \|G\|_\infty \quad (3.9)$$

- **Cas Multivariable**

Dans la cas Multivariable, on montre que cette norme est le maximum de la plus grande valeur singulière de la matrice de transfert $G(j\omega)$, soit:

$$\|G\|_\infty = \sup_{\omega} \bar{\sigma}(G(j\omega)) \quad (3.10)$$

La norme H_∞ a permis :

- De donner un formalisme unifié pour l'analyse
- La synthèse fréquentielle
- De prendre en compte des objectifs de suivi de référence, stabilité, de marges de stabilités
- Modeler les différents transferts



Calcul de la norme H_∞

Contrairement à la norme H_2 , il n'existe pas une méthode analytique pour calculer la norme H_∞ d'une matrice de transfert.

Pour évaluer la norme H_∞ d'une matrice de transfert, on choisit une valeur arbitraire de γ , on calcule les valeurs propres de la matrice $M\gamma$, si aucune n'est imaginaire pure on diminue γ et on recommence, sinon on augmente γ et on recommence. On ne peut donc pas calculer la valeur de la norme H_∞ , mais seulement en donner une borne supérieure, aussi proche que l'on veut $\|G\|_\infty < \gamma$.

Où la matrice $M\gamma$ est une matrice hamiltonienne définie par :

$$M\gamma = \begin{bmatrix} A & \frac{BB^T}{\gamma} \\ -\frac{C^T C}{\gamma} & -A^T \end{bmatrix} \quad (3.11)$$

Le calcul de la norme H_∞ est donc plus complexe que celui de la norme H_2 , et il n'existe pas à notre connaissance de méthode de calcul directe de cette norme. On ne peut en fait que tester si la norme H_∞ est inférieure ou non à un scalaire γ donné. Il est donc nécessaire d'approximer la valeur de cette norme en calculant dans un premier temps des bornes inférieures et supérieures, puis d'avoir recours à une approche itérative.

3.2 Critères de robustesse

Une commande est dite robuste si on peut garantir la satisfaction d'un certain niveau de performances malgré la présence d'erreur de modèle (notion d'incertitude).

IL y'a deux notions plus importantes de la robustesse, stabilité robuste et performance robuste, avant ces deux notions, nous commencerons par la définition des spécifications fréquentielles.

3.2.1 Spécifications fréquentiels

Les méthodes fréquentielles reposent sur l'hypothèse qu'un cahier des charges peut être traduit par un cahier des charges avec que des spécifications fréquentielles. Une spécification



fréquentielle sur une fonction de transfert en boucle fermée est définie par des gabarits (fréquentiels) sur une certaine gamme de pulsations.

Définition 3.2 (Spécification fréquentielle)

Soit $T(s)$ une fonction de transfert en boucle fermée.

Une spécification fréquentielle sur $T(s)$ est définie par n_g fonctions $Gab_j(\omega)$ strictement positives sur $[\underline{\omega}_j, \bar{\omega}_j]$ où $\underline{\omega}_j < \bar{\omega}_j$, où $\underline{\omega}_j$ peut être nulle et où $\bar{\omega}_j$ peut être infini.

La spécification fréquentielle sur $T(s)$ est alors la suivante :

$$\forall j=1, \dots, n_g, \forall \omega \in [\underline{\omega}_j, \bar{\omega}_j], |T(j\omega)| < Gab_j(\omega). \quad (3.12)$$

Les bornes sur les spécifications quantifiables d'un cahier des charges sont reliées à des caractéristiques des gabarits.

Remarque

Une spécification fréquentielle peut aussi porter sur la phase de la fonction de transfert en boucle fermée. Il est, en général, possible de réécrire une spécification sur la phase comme une spécification sur le module.

Exemple

Par exemple, sur la Figure 3.3, il est tracé trois gabarits fréquentiels sur la fonction de transfert en boucle fermée $S(s) = (I + G(s)K(s))^{-1}$. Pour un système $G(s)$ monovarié commandé par un correcteur à un degré de liberté, ces gabarits permettent de prendre en compte plusieurs spécifications.

Le premier gabarit

$$\forall \omega \in \mathbb{R}, |S(j\omega)| < Gab_1(\omega)$$

Permet de prendre en compte que la marge de module en dB est supérieure à $-20 \log(G_\infty)$.

Le deuxième gabarit

$$\forall \omega \in [0.3, 2], |S(j\omega)| < Gab_2(\omega)$$

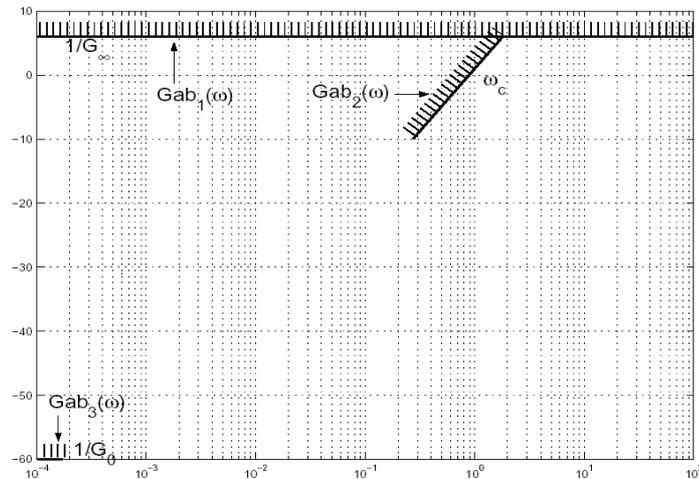


Figure 3.3 Exemples de gabarits

Permet de prendre en compte la rapidité du suivi de trajectoire de référence. Plus ω_c (la pulsation pour laquelle le gabarit vaut 0 dB) est grand, plus la rapidité garantie est importante.

Le troisième gabarit

$$\forall \omega \in [0, 2 \times 10^{-4}], |S(j\omega)| < Gab_3(\omega)$$

Permet de prendre en compte que l'erreur statique en pourcentage est inférieure à l'inverse de G_0 .

3.2.2 Stabilité robuste

S'il s'agit de commande robuste, la première exigence est la robustesse de la stabilité. On dit qu'un système est robuste en stabilité s'il demeure stable en présence d'incertitude telle que les mesures de modélisation, les bruits de mesures ou les perturbations externes. Dès les travaux de Bode et Nyquist (1930-1950) cette exigence est présente à travers les notions de marge de stabilité : marge de gain et marge de phase... Dans l'approche moderne (1980-) les marges de stabilité doivent être spécifiées en fonction des incertitudes, pour garantir de façon certaine la stabilité.

3.4.2.1 Marges de stabilité

La trace de Nyquist donne l'information sur la notion de marge de stabilité. La distance entre le point critique -1 et le point le plus proche sur la trace de Nyquist de $KG(s)$ égale à $1/\|S_\infty\|$:

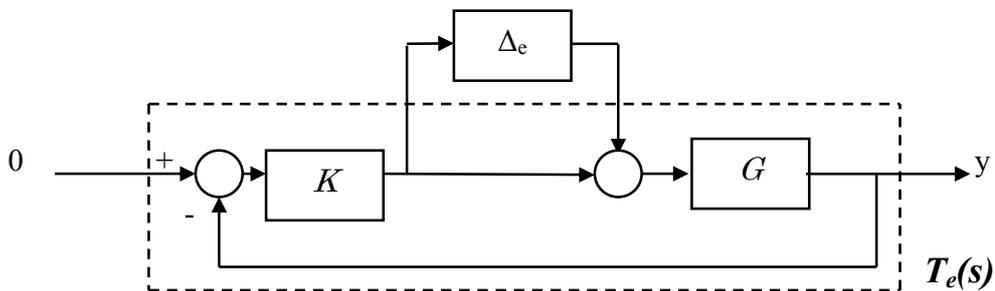


$$\begin{aligned}
 \delta = \text{distance entre } -1 \text{ et la trace de Nyquist} &= \inf_{w \in R} \left\{ \min_i |1 + \lambda_i \{KG(jw)\}| \right\} \\
 &= \inf_{w \in R} \left\{ \lambda_{\min} \{I + KG(jw)\} \right\} \\
 &= \inf_{w \in R} \sigma_{\min} \{I + KG(jw)\} \\
 &= \inf_{w \in R} \sigma_{\min}^{-1} \{I + KG(jw)\}^{-1} \\
 &= \frac{1}{\sup_{w \in R} \bar{\sigma} \{I + KG(jw)\}^{-1}} \\
 \Rightarrow \delta &= \frac{1}{\|S(j\omega)\|_{\infty}} \tag{3.13}
 \end{aligned}$$

où $\bar{\sigma}$ désigne la valeur singulière maximale.

3.4.2.2 Théorème de stabilité robuste de modèle incertain

Soit le système incertain (incertitudes multiplicatives en entrée), bouclé par un régulateur K :



Equivalent :

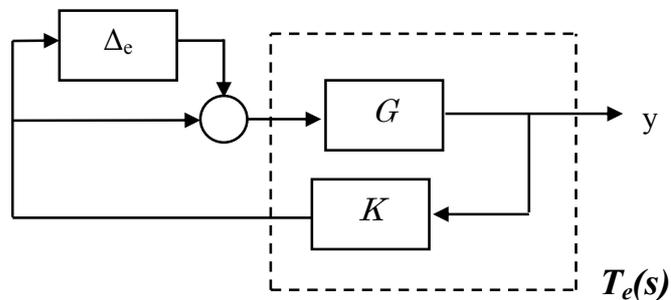


Figure 3.4 système incertain bouclé.



*THÉORIE DE LA
COMMANDE ROBUSTE*

4.1 Introduction

La commande robuste repose sur la structure de commande à contre-réaction et sur les modèles linéaires à temps invariant (LTI) des systèmes dynamiques (figure 4.1 ci-dessous). Le principe de la commande à contre-réaction (feedback control systems) est au coeur des théories classiques et modernes de la commande et définit une structure de commande très générale fondée sur la comparaison entre l'information supportée par un ensemble de signaux de sorties y mesurés par des capteurs et l'information supportée par un ensemble de signaux de référence souhaités. Les différences constatées lors de cette comparaison sont traitées par le système de commande afin d'élaborer un signal de commande « u » affectant le système.

Soit la forme standard d'un système de commande :

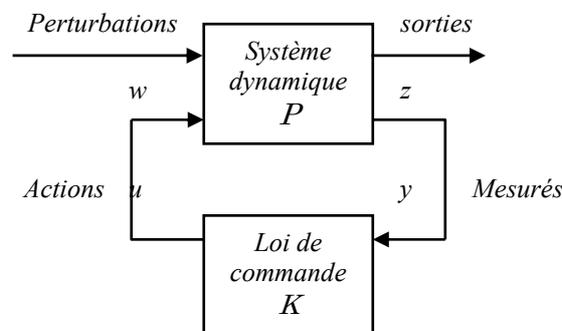


Figure 4.1 La forme standard de la commande robuste

P est appelé le **modèle généralisé** et K est le **correcteur généralisé**. P modélise les interactions dynamiques entre les deux ensembles de signaux d'entrée $w \in R^{n_w}$, $u \in R^m$ et les deux ensembles de signaux de sortie $z \in R^{n_z}$, $y \in R^r$, « u » est le vecteur des commandes délivrées par le correcteur généralisé. w est le vecteur des signaux exogènes d'entrée (signaux exogènes d'entrée pour le modèle généralisé autres que ceux dans u). y est le vecteur des signaux de sortie accessible à la mesure et donc disponibles pour élaborer la commande. Enfin, z est le vecteur des signaux régulés de sortie, Il caractérise le bon fonctionnement de la structure commandée. Le schéma standard prend toute son importance dans un cadre de travail linéariste où les interconnexions entre les signaux d'entrée et de sortie mises en oeuvre par le modèle et le correcteur généralisés sont supposées linéaires.

La matrice de transfert représentant P est partitionnée en 4 blocs :

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \quad (4.1)$$

Où P_{11} (respectivement P_{22}) représente le transfert entre w (resp. u) et z (resp. y).

Hypothèses

- Le modèle généralisé P est un modèle linéaire invariant dans le temps et en dimension finie (LTI).
- Le correcteur généralisé K est un modèle linéaire invariant dans le temps et en dimension finie (LTI).

Ces hypothèses sont évidemment restrictives mais elles sont souvent faites en pratique. En effet, de nombreux systèmes non linéaires (tous les systèmes réels le sont en pratique) peuvent être correctement approximés par des modèles linéaires. De plus, les modèles LTI sont des objets mathématiques bien caractérisés, simples à manipuler et pour l'étude desquels de nombreux outils mathématiques puissants sont disponibles. Cette simplicité n'est toutefois pas sans conséquences sur la précision relative de ce modèle et sur sa représentativité du système physique réel. Il est donc judicieux d'adjoindre au modèle LTI un modèle mathématique de l'incertitude représentant l'écart entre la réalité physique et le modèle mathématique simplifié définissant ainsi une problématique nouvelle en Automatique.

4.2 Modélisation incertaine

Le fait que le modèle LTI soit incertain est généralement formalisé par la dépendance du modèle vis-à-vis d'un opérateur d'incertitude $\Delta \in \Delta$ qui vient compléter le schéma standard précédent pour donner le schéma standard de la figure 4.2.

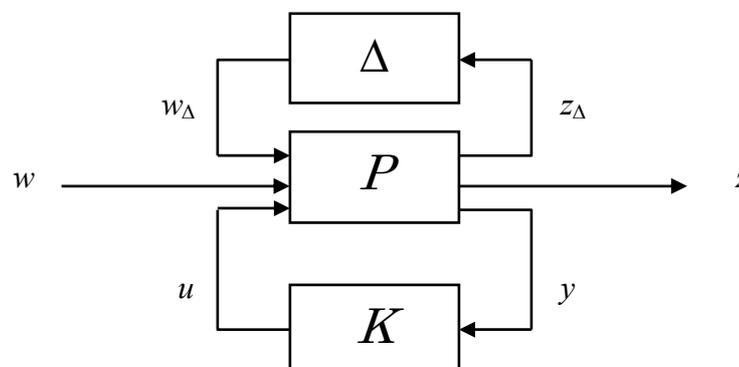


Figure 4.2 Le schéma standard avec incertitude



Le schéma 4.2 représente une structure particulière du modèle d'incertitude où un opérateur d'incertitude Δ opère par bouclage sur un modèle nominal défini par $P(s)$. La nature de ce bouclage caractérise la complexité du modèle incertain complet. Travaillant désormais sous les hypothèses linéarités précédentes, le schéma-bloc (4.2) se traduit par les relations suivantes:

$$\begin{bmatrix} z_\Delta \\ z \\ y \end{bmatrix} = P \begin{bmatrix} w_\Delta \\ w \\ u \end{bmatrix}$$

$$w_\Delta = \Delta z_\Delta \quad \Delta \in \Delta \quad (4.2)$$

$$u = Ky$$

Les objets systèmes P et K sont alors représentés par des matrices de transfert $P(s)$ et $K(s)$. $P(s)$ peut être partitionnée en sous-blocs conformément à la taille des différents vecteurs d'entrée et de sortie.

$$P(s) = \begin{bmatrix} P_{z_\Delta w_\Delta}(s) & P_{z_\Delta w}(s) & P_{z_\Delta u}(s) \\ P_{zw_\Delta}(s) & P_{zw}(s) & P_{zu}(s) \\ P_{yw_\Delta}(s) & P_{yw}(s) & P_{yu}(s) \end{bmatrix} \quad (4.3)$$

En opérant les bouclages supérieurs et inférieurs de Δ et K sur P , le transfert incertain en boucle fermée est ainsi obtenu.

Définition (LFT)

Soient les matrices complexes :

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \in C^{(p_1+p_2) \times (q_1+q_2)} \quad K \in C^{q_2 \times p_2} \quad \Delta \in C^{q_1 \times p_1} \quad (4.4)$$

- Si la matrice $(1 - P_{22}K)^{-1}$ existe alors la **LFT inférieure** est la fonction L_l définie Par :

$$L_l(P, \bullet) : C^{q_2 \times p_2} \rightarrow C^{p_1 \times q_1}$$

$$L_l(P, K) = P_{11} + P_{12}K(1 - P_{22}K)^{-1}P_{21} \quad (4.5)$$

- Si la matrice $(1 - P_{11}\Delta)^{-1}$ existe alors la **LFT supérieure** est la fonction L_u définie par :

$$L_u(P, \bullet) : C^{q_1 \times p_1} \rightarrow C^{p_2 \times q_2}$$

$$L_u(P, \Delta) = P_{22} + P_{21}\Delta(1 - P_{11}\Delta)^{-1}P_{12} \quad (4.6)$$

La matrice P est appelée **la matrice coefficient**.

Par exemple, si le bouclage de l'opérateur d'incertitude Δ sur P est effectué en suivant la partition adéquate, les relations de bouclage deviennent:

$$\begin{bmatrix} z \\ y \end{bmatrix} = P(\Delta) \begin{bmatrix} w \\ u \end{bmatrix} = \left(\begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} + \begin{bmatrix} P_{zw\Delta} \\ P_{yw\Delta} \end{bmatrix} \Delta [1 - P_{z\Delta w\Delta} \Delta]^{-1} \begin{bmatrix} P_{z\Delta w} & P_{z\Delta u} \end{bmatrix} \right) \begin{bmatrix} w \\ u \end{bmatrix} \quad (4.7)$$

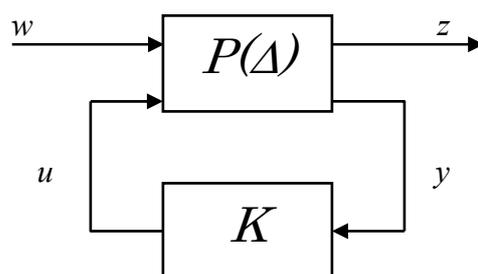


Figure 4.3 Le schéma standard incertain de synthèse

La forme (4.7) schématisée dans la figure (4.3) fait intervenir une transformation linéaire fractionnaire (LFT) sur la matrice Δ qui permet de définir un cadre de travail très général défini comme le paradigme LFT [5]. Plus formellement, nous distinguerons deux LFT différentes correspondant aux deux bouclages possibles du schéma (4.2).

4.3 Analyse et synthèse robuste

Le modèle incertain et sa structure étant fixés, l'objectif principal de la structure de commande est d'assurer la stabilité de la boucle fermée ainsi qu'un certain degré de performance pour l'ensemble des réalisations possibles du modèle incertain. Il s'agit donc de réduire la sensibilité de la structure de commande en présence des variations paramétriques et d'éventuelles perturbations affectant le modèle du système à commander.

4.3.1 Analyse robuste

La problématique générale de l'analyse robuste peut s'énoncer comme suit. Etant donnée une modélisation incertaine et une loi de commande associée, il s'agit d'établir par une étape d'analyse si cette loi de commande garantit la stabilité et un certain niveau de performance de la boucle de contre-réaction pour toute réalisation du modèle dans son ensemble



d'incertitudes. Suivant les résultats de cette dernière étape, une stratégie de correction peut être nécessaire, résultant alors en la synthèse d'un nouveau compensateur. Une fois celui-ci obtenu, une nouvelle étape d'analyse du système bouclé intervient afin d'attester des propriétés réelles du système corrigé. L'étape d'analyse est donc très importante dans le processus de synthèse puisqu'elle va conditionner en partie les choix effectués pour la synthèse mais également en sanctionner les résultats.

L'analyse est usuellement séparée en deux problèmes présentant des degrés de difficulté différents : l'analyse de stabilité robuste et l'analyse de performance robuste. La notion de valeur singulière structurée (μ -théorie) [6] a permis de donner un cadre général d'étude permettant de ramener l'analyse de performance robuste à celui de l'analyse de la stabilité robuste vis-à-vis d'un ensemble structuré d'incertitudes mettant ainsi en lumière la difficulté théorique de l'étude de la performance.

Ces problèmes d'analyse de la stabilité et de la performance robustes ont été très largement abordés dans la littérature et ont donc atteint un état de maturité important impliquant la grande complexité des problématiques actuelles. Même si une typologie exacte et exhaustive des approches possibles est toujours délicate, il est toutefois possible de dégager trois grandes approches :

- Les approches algébriques ou approches polynomiales [7], [8].
- La théorie de la valeur singulière structurée ou μ -théorie [5].
- La théorie de Lyapunov [9].

4.3.2 Synthèse robuste

Le modèle incertain du système étant défini, il s'agit de calculer une loi de commande compatible avec la structure en contre-réaction de la figure 4.1 et qui confère au système la propriété d'être le plus insensible possible aux incertitudes et perturbations pouvant l'affecter, tout en permettant d'atteindre un niveau de performance satisfaisant. Cela implique le choix d'une structure de commande par la définition de spécifications fonctionnelles et de classes, mais également le choix d'une méthode de synthèse.

Dans notre étude, seule la synthèse robuste sera détaillée, et elle est volontairement limitée à deux problèmes classiques de la théorie de la commande robuste:

La synthèse de loi de commande par retour de sortie statique et la synthèse H_∞ par retour de sortie statique.

4.3.2.1 La synthèse par retour de sortie statique

❖ Formulation du problème et hypothèses

Soit le modèle d'état linéaire temps invariant (LTI):

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{4.8}$$

Où $x \in R^n$, $y \in R^r$ et $u \in R^m$. A , B et C sont des matrices constantes avec de dimensions appropriées.

Il s'agit de résoudre le problème suivant:

Problème : (retour de sortie statique)

Déterminer $K \in R^{r \times m}$ telle que $A + BKC$ soit stable ou en d'autres termes, le spectre de la matrice en boucle fermée $A + BKC$ est inclus dans le demi-plan complexe gauche strict.

$$\Lambda(A + BKC) \subset C^-\tag{4.9}$$

Le spectre Λ : est l'ensemble de solutions complexes de l'équation caractéristique d'une matrice (c à d l'ensemble de valeurs propres de la matrice).

Les hypothèses suivantes sont définies habituellement pour ce problème.

- La paire (A, B) est stabilisable et la paire (A, C) est détectable.
- Les matrices B et C sont de rang plein.

$$\text{rang}(B) = m \quad \text{rang}(C) = r\tag{4.10}$$

La deuxième hypothèse ne restreint en rien la généralité du problème initial puisqu'une simple élimination des entrées et/ou sorties redondantes permet de l'assurer.

La première hypothèse est une condition nécessaire et suffisante de stabilisabilité par retour de sortie (garantit l'existence d'une loi de commande K qui stabilise le système en boucle fermée).



❖ Formulation par la théorie de Lyapunov

Une catégorie d'approches qui proposent des méthodes de construction de la matrice de retour de sortie statique est essentiellement fondée sur la théorie de Lyapunov. Dans le cadre linéariste, il est en effet simple de reformuler le problème d'existence comme un problème conjoint de recherche d'une matrice de Lyapunov P et de la matrice K devant satisfaire une inégalité de Lyapunov en boucle fermée.

Théorème

Le problème de retour de sortie statique a une solution si et seulement si $\exists K \in R^{r \times m}$ et $\exists P \in S_n^{++}$ solutions de :

$$(A + BKC)^T P + P(A + BKC) < 0 \quad (4.11)$$

Le problème de recherche de K est donc transformé en un problème de recherche des matrices (P, K) dans un ensemble non convexe défini par une inégalité matricielle bilinéaire.

4.3.2.2 La synthèse H_∞ par retour de sortie statique

❖ Synthèses de correcteur H_∞

Soit le système P décrit sur le schéma bloc représente figure (4.1) où

- le vecteur des sorties commandées $z(t)$ est de dimension n_z ;
- le vecteur des entrées de critère $w(t)$ est de dimension n_w ;
- le vecteur des sorties mesurées $y(t)$ est de dimension r ;
- le vecteur des entrées de commande $u(t)$ est de dimension m .

Le système P admet pour équations d'état :

$$\begin{cases} \dot{x}(t) = Ax(t) + B_w w(t) + B_u u(t) \\ z(t) = C_z x(t) + D_{zw} w(t) + D_{zu} u(t) \\ y(t) = C_y x(t) + D_{yw} w(t) + D_{yu} u(t) \end{cases} \quad (4.12)$$

L'entier n est l'ordre de représentation d'état. Dans le domaine de Laplace, les équations du système se réécrivent :

$$P(p) = \begin{bmatrix} P_{zw}(p) & P_{zu}(p) \\ P_{yw}(p) & P_{yu}(p) \end{bmatrix} = \begin{bmatrix} D_{zw} & D_{zu} \\ D_{yw} & D_{yu} \end{bmatrix} + \begin{bmatrix} C_z \\ C_y \end{bmatrix} (pI - A)^{-1} [B_w \quad B_u] \quad (4.13)$$

$$\begin{bmatrix} z(p) \\ y(p) \end{bmatrix} = P(p) \begin{bmatrix} w(p) \\ u(p) \end{bmatrix}$$

$$u(p) = K(p)y(p)$$

soit

$$z(p) = \underbrace{(P_{zw}(p) + P_{zu}(p)K(p)(I - P_{yu}(p)K(p))^{-1}P_{yw}(p))}_{\text{noté } P^*K \text{ ou } F_l(P,K)} w(p) \quad (4.14)$$

Problème (H_∞ standard)

1. Etant donné $\gamma > 0$, existe-il une loi de commande K telle que :
 - Le système bouclé $F_l(P, K)$ soit asymptotiquement stable (tous les pôles du système en boucle fermée sont à partie réelle négative). F_l c'est la LFT inférieur.
 - $\|F_l(P^*K)\|_\infty < \gamma$

$$\text{avec } P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}; \quad T_{ZW} = P_{11} + P_{12}K(1 - P_{22})^{-1}P_{21}$$

2. Si oui, construire une loi de commande K assurant pour le système en boucle fermée les deux propriétés précédentes.

Ce problème admet deux résolutions possibles. Elles sont en général mise en oeuvre numériquement sur un ordinateur.

La première approche repose sur la résolution d'une série d'équations de Riccati « Algorithme de Glover-Doyle » [19]. C'est la solution la plus simple et la plus fiable numériquement. Cependant, elle nécessite la vérification d'un certain nombre d'hypothèses qui peuvent être non vérifiées alors que le problème H_∞ standard admet une solution.

Ces hypothèses sont :

Hypothèses solution par équations de Riccati

1. $\text{rang}(D_{zu}) = m$ et $\text{rang}(D_{yw}) = r$: ce sont des conditions suffisantes pour assurer que la loi de commande $K(p)$ est propre. De façon implicite, cela veut dire aussi qu'il y a plus de sorties commandées z que d'entrées de commande u ($n_z \geq m$) et qu'il y a plus d'entrées de critère w que de mesures y ($n_w \geq r$).

2. $\text{rang} \begin{bmatrix} A - j\omega I_n & B_u \\ C_z & D_{zu} \end{bmatrix} = n + m$ garantit que le transfert P_{zu} n'a pas de zéro sur l'axe imaginaire.

3. $\text{rang} \begin{bmatrix} A - j\omega I_n & B_w \\ C_y & D_{yw} \end{bmatrix} = n + r$ garantit que le transfert P_{yw} n'a pas de zéro sur l'axe imaginaire.

Les deux éléments essentiels dans cette approche sont, l'équation de Riccati de la forme générale :

$$A^T X + XA - XRX + Q = 0 \quad (4.15)$$

où A , Q et R sont des matrices réelles de dimension $n \times n$, avec Q et R symétrique.

Et la matrice Hamiltonienne associée à cette équation qui est définie par :

$$H = \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \quad (4.16)$$

La deuxième approche permet de faire l'économie de ces hypothèses, qui est basée sur la résolution d'un problème d'optimisation convexe sous contraintes d'inégalités matricielles linéaires (LMIs). Ici nous allons donner les grandes lignes de la démonstration de cette solution.

❖ Synthèse H_∞ par retour de sortie

Soit le système P d'ordre n défini par (4.12), on recherche une loi de commande par retour de sortie (c-à-d une matrice $\underline{K} \in R^{r \times m}$) telle que $u(t) = \underline{K} y(t)$ assure :

- le système bouclé est stable ;
- la norme H_∞ de la fonction de transfert en boucle fermée entre l'entrée w et la sortie z est inférieure à γ .

Les développements qui suivent sont basés sur le lemme suivant.

Lemme (Réel borné)

Soit un système linéaire stationnaire multivariable stable G décrit par une représentation d'état minimale :

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bw(t) \\ z(t) &= Cx(t) + Dw(t)\end{aligned}\quad (4.17)$$

où $x(t)$ est l'état du système, $w(t)$ l'entrée et $z(t)$ est la sortie. Alors $\|G\|_\infty \leq \gamma$ si et seulement s'il existe une matrice \underline{P} telle que

$$\begin{cases} \underline{P} \geq 0 \\ \begin{bmatrix} A^T \underline{P} + \underline{P}A + C^T C & \underline{P}B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \gamma^2 I \end{bmatrix} \leq 0 \end{cases} \quad (4.18)$$

Alors, le système en boucle fermée admet pour équations d'état :

$$\begin{cases} \dot{x}(t) = (A + B_u \underline{K} C_y) x(t) + (B_w + B_u \underline{K} D_{yw}) w(t) \\ z(t) = (C_z + D_{zu} \underline{K} C_y) x(t) + (D_{zw} + D_{zu} \underline{K} D_{yw}) w(t) \end{cases} \quad (4.19)$$

$$\Leftrightarrow \begin{cases} \dot{x}(t) = A_{bf} x(t) + B_{bf} w(t) \\ z(t) = C_{bf} x(t) + D_{bf} w(t) \end{cases}$$

En appliquant le lemme réel borné au système en boucle fermée, il existe un correcteurs par retour de sortie s'il existe une matrice \underline{K} et une matrice symétrique \underline{P} telle que $\underline{P} > 0$ et :

$$\begin{bmatrix} A_{bf}^T \underline{P} + \underline{P} A_{bf} + C_{bf}^T C_{bf} & \underline{P} B_{bf} + C_{bf}^T D_{bf} \\ B_{bf}^T \underline{P} + D_{bf}^T C_{bf} & D_{bf}^T D_{bf} - \gamma^2 I \end{bmatrix} \leq 0 \quad (4.20)$$

Les variables d'optimisation sont \underline{P} et \underline{K} . Cette inégalité n'étant pas affine en \underline{P} et en \underline{K} , elle ne définit pas une LMI. Par une série de transformations, on va montrer que l'on peut obtenir une contrainte LMI équivalente après un changement de variables adéquat.

L'inégalité (4.20) peut se réécrire :

$$\begin{bmatrix} A_{bf}^T \underline{P} + \underline{P} A_{bf} & \underline{P} B_{bf} \\ B_{bf}^T \underline{P} & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} C_{bf}^T \\ D_{bf}^T \end{bmatrix} \begin{bmatrix} C_{bf} & D_{bf} \end{bmatrix} \leq 0 \quad (4.21)$$

Lemme (complément de Schur)

Soit une matrice symétrique partitionnée :

$$\begin{bmatrix} Q & R \\ R^T & S \end{bmatrix}$$

où Q et S sont des matrices carrées. Cette matrice est définie positive si et seulement si A et $S - R^T Q^{-1} R$ sont définies positives.

Lemme (complément de Schur modifié)

Soit une matrice symétrique partitionnée :

$$\begin{bmatrix} Q & R \\ R^T & S \end{bmatrix}$$

où Q et S sont des matrices carrées, avec Q définie positive. Cette matrice est définie positive si et seulement si $S - R^T Q^{-1} R$ est semi définie positive.

L'inégalité (4.19) peut alors se réécrire par :

$$\underbrace{\begin{bmatrix} A_{bf}^T \underline{P} + \underline{P} A_{bf} & \underline{P} B_{bf} \\ B_{bf}^T \underline{P} & -\gamma^2 I \end{bmatrix}}_S - \underbrace{\begin{bmatrix} C_{bf}^T \\ D_{bf}^T \end{bmatrix}}_{R^T} \underbrace{(-I)}_{Q^{-1}} \underbrace{\begin{bmatrix} C_{bf} & D_{bf} \end{bmatrix}}_R \leq 0 \quad (4.22)$$

Par application du lemme de Schur modifier, on a alors (4.22) qui est équivalente à :

$$\begin{bmatrix} A_{bf}^T \underline{P} + \underline{P} A_{bf} & \underline{P} B_{bf} & C_{bf}^T \\ B_{bf}^T \underline{P} & -\gamma^2 I & D_{bf}^T \\ C_{bf} & D_{bf} & -I \end{bmatrix} \leq 0$$

$$\Leftrightarrow \begin{bmatrix} (A + B_u \underline{K} C_y)^T \underline{P} + \underline{P} (A + B_u \underline{K} C_y) & \underline{P} (B_w + B_u \underline{K} D_{yw}) & (C_z + D_{zu} \underline{K} C_y)^T \\ (B_w + B_u \underline{K} D_{yw})^T \underline{P} & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z + D_{zu} \underline{K} C_y & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0$$

$$\Leftrightarrow \begin{bmatrix} A^T \underline{P} + \underline{P} A + C_y^T \underline{K}^T B_u^T \underline{P} + \underline{P} B_u \underline{K} C_y & \underline{P} B_w + \underline{P} B_u \underline{K} D_{yw} & (C_z + D_{zu} \underline{K} C_y)^T \\ B_w^T \underline{P} + D_{yw}^T \underline{K}^T B_u^T \underline{P} & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z + D_{zu} \underline{K} C_y & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0 \quad (4.23)$$

L'intérêt de cette transformation est que, dans l'inégalité obtenus, il ne reste plus que $C_y^T \underline{K}^T B_u^T \underline{P}$, $\underline{P} B_u \underline{K} C_y$, $\underline{P} B_u \underline{K} D_{yw}$ et $D_{yw}^T \underline{K}^T B_u^T \underline{P}$ comme terme bilinéaire. On va utiliser une

nouvelle transformation puis un changement de variable de façon à se ramener à une inégalité affine. Pour cela, on va se baser sur la propriété suivante :

Propriété

$$\forall T \text{ inversible } \in R^{n \times n}, A \leq 0 \Leftrightarrow T^T A T \leq 0$$

En appliquant cette propriété sur l'inégalité (4.23) avec :

$$T = \begin{bmatrix} \underline{P}^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

On obtient :

$$\begin{bmatrix} \underline{P}^{-1} A^T + A \underline{P}^{-1} + \underline{P}^{-1} C_y^T \underline{K}^T B_u^T + B_u \underline{K} C_y \underline{P}^{-1} & B_w + B_u \underline{K} D_{yw} & \underline{P}^{-1} (C_z + D_{zu} \underline{K} C_y)^T \\ B_w^T + D_{yw}^T \underline{K}^T B_u^T & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z \underline{P}^{-1} + D_{zu} \underline{K} C_y \underline{P}^{-1} & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0 \quad (4.24)$$

En posant $\underline{Q} = \underline{P}^{-1} \in R^{n \times n}$, on obtient :

$$\begin{bmatrix} \underline{Q} A^T + A \underline{Q} + \underline{Q} C_y^T \underline{K}^T B_u^T + B_u \underline{K} C_y \underline{Q} & B_w + B_u \underline{K} D_{yw} & \underline{Q} (C_z + D_{zu} \underline{K} C_y)^T \\ B_w^T + D_{yw}^T \underline{K}^T B_u^T & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z \underline{Q} + D_{zu} \underline{K} C_y \underline{Q} & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0 \quad (4.25)$$

D'autre part, on a $\underline{P} > 0 \Leftrightarrow \underline{Q} > 0$.

Si en mettant $\underline{Y} = \underline{P} B_u \underline{K}$ et $\underline{X} = \underline{K} C_y \underline{Q}$, les deux inégalités (4.23) et (4.25) sont équivalentes à :

$$\begin{bmatrix} A^T \underline{P} + \underline{P} A + C_y^T \underline{Y}^T + \underline{Y} C_y & \underline{P} B_w + \underline{Y} D_{yw} & (C_z + D_{zu} \underline{K} C_y)^T \\ B_w^T \underline{P} + D_{yw}^T \underline{Y}^T & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z + D_{zu} \underline{K} C_y & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0 \quad (4.26)$$

$$\begin{bmatrix} \underline{Q} A^T + A \underline{Q} + \underline{X}^T B_u^T + B_u \underline{X} & B_w + B_u \underline{K} D_{yw} & \underline{Q} C_z^T + \underline{X}^T D_{zu}^T \\ B_w^T + D_{yw}^T \underline{K}^T B_u^T & -\gamma^2 I & (D_{zw} + D_{zu} \underline{K} D_{yw})^T \\ C_z \underline{Q} + D_{zu} \underline{X} & D_{zw} + D_{zu} \underline{K} D_{yw} & -I \end{bmatrix} \leq 0 \quad (4.27)$$

Ces deux dernières inégalités sont bien affines en les variables \underline{P} , \underline{Q} , \underline{Y} et \underline{X} . Donc rechercher \underline{P} , \underline{Q} , \underline{Y} et \underline{X} tels que $\underline{P} > 0$, $\underline{Q} > 0$, (4.26) et (4.27) soient satisfaites est un problème de faisabilité LMI.

❖ Existence d'un correcteur H_∞

On peut être intéressé par déterminer s'il existe un correcteur H_∞ pour γ donné, sans le calculer explicitement. Pour cela, on dispose du résultat suivant.

Lemme d'élimination

Soit $G = G^T \in R^{n \times n}$, $U \in R^{r \times n}$ avec $\text{Rang}(U) = r < n$ et $V \in R^{s \times n}$ avec $\text{Rang}(V) = s < n$. Alors, il existe une matrice $K \in R^{r \times s}$ telle que l'inégalité suivante est vérifiée :

$$G + U^T K V + V^T K^T U < 0$$

Si et seulement si :

$$U_\perp^T G U_\perp < 0 \quad \text{et} \quad V_\perp^T G V_\perp < 0,$$

Avec

- U_\perp engendre le noyau de l'application linéaire associée à la matrice U ;
- V_\perp engendre le noyau de l'application linéaire associée à la matrice V .

Ce résultat est remarquable puisqu'il donne une condition nécessaire et suffisante permettant de tester l'existence d'une matrice K telle que $G + U^T K V + V^T K^T U < 0$ est (semi) définie négative, sans rechercher K explicitement.

Si on applique cet lemme, les inégalités (4.26) et (4.27) sont équivalentes à :

$$\begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp^T \begin{bmatrix} \underline{Q} A^T + A \underline{Q} & B_w & \underline{Q} C_z^T \\ B_w^T & -\gamma I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -\gamma I \end{bmatrix} \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp < 0 \quad (4.28)$$

$$\begin{bmatrix} C_y & D_{yw} & 0 \end{bmatrix}_\perp^T \begin{bmatrix} A^T \underline{P} + \underline{P} A & \underline{P} B_w & C_z^T \\ B_w^T \underline{P} & -\gamma I & D_{zw}^T \\ C_z & D_{zw} & -\gamma I \end{bmatrix} \begin{bmatrix} C_y & D_{yw} & 0 \end{bmatrix}_\perp < 0 \quad (4.29)$$

Tester l'existence d'un correcteur qui assure un niveau de performance γ donné est un problème de faisabilité LMI. Tester l'existence d'un correcteur qui assure le plus petit niveau de performance γ peut s'écrire comme un problème de minimisation d'un coût linéaire sous contraintes LMI.

*APPLICATION DE LA
MÉTHODE ILMI*

5.1 Introduction

La technique d'automatique la plus répandue est le contrôle en boucle fermée, de nombreuses méthodes de commande des processus utilisent le principe de retour d'état, qu'est une récupération de l'état du modèle choisi pour élaborer la commande. Cependant il y a des cas où les états ne sont pas disponibles alors, on peut utiliser les sorties des systèmes pour le retour. Dans notre cas le problème est de trouver la matrice de gain du retour de sortie statique K qui satisfait la stabilité du système.

A la base de l'approche de l'inégalité matricielle affine, il existe un nombre important de méthodes de synthèse qui cherchent un gain respectant la stabilité. Il est proposé dans ce chapitre une méthode itérative comportant deux algorithmes. Ces algorithmes sont successifs ; le premier permet de trouver les variables initiales pour le bon démarrage des calculs du second algorithme, qui fait le calcul du gain K de cette méthode (SOF).

De même, cette méthode basée sur le retour de sortie statique peut être généralisée pour résoudre le problème de la commande H^∞ . Cette méthode sera utilisée au calcul de deux régulateur PI et PID.

5.2 Formalisation du problème de la stabilisation :

Soit le modèle d'état linéaire temps-invariant de la figure (4.1 dans le chapitre 4§4.1) donné par :

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \\ z(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t) \\ y(t) = C_2x(t) + D_{21}w(t) \end{cases} \quad (5.1)$$

Où

$x(t) \in R^n$: vecteur d'état;

$w(t) \in R^r$: vecteur des signaux exogènes d'entrée;

$u(t) \in R^m$: vecteur des commandes;

$z(t) \in R^q$: vecteur des signaux régulés de sortie ;

$y(t) \in R^p$: vecteur des sorties mesurés.

$A, B_1, B_2, C_1, C_2, D_{11}, D_{12}$ et D_{21} , sont des matrices constantes de dimensions appropriées.

Les hypothèses suivantes sont établies sur l'expression (5.1)

Hypothèse 5.1

La paire (A, B_2) est stabilisable et la paire (C_2, A) est détectable.

Le problème de stabilisation par retour de sortie statique (SOF) est donc de trouver un contrôleur

$$u(t) = K y(t) \quad (5.2)$$

où $K \in R^{m \times p}$, tel que le système en boucle fermée avec $w(t) = 0$ donné par :

$$\dot{x} = (A + B_2 K C_2) x(t) \quad (5.3)$$

soit stable.

Comme nous avons déjà vu dans le chapitre précédent, le système en boucle fermée (5.3) est stable si et seulement s'il existe une matrice de Lyapunov $P = P^T > 0$ telle que :

$$P(A + B_2 K C_2) + (A + B_2 K C_2)^T P < 0 \quad (5.4)$$

5.3 Description de la méthode itérative ILMI

La condition (5.4) est une BMI qui n'est pas un problème convexe optimal. Une méthode ILMI a été proposée dans [10], où une nouvelle variable X a été introduite, telle que la condition de stabilité devient suffisante lorsque $X \neq P$. L'algorithme présenté dans [10] a essayé de trouver certaines valeurs de X proche de P en utilisant une méthode itérative et la procédure itérative prend entre P et X . D'autre part, une substitution de la formulation LMI a été proposée dans [11], où des nouvelles variables, telles que L et M ont été introduites telles que la condition de stabilité devient aussi suffisante lorsque $L \neq R^{-1} B_2 P$ ou $M \neq K C_2$. L'algorithme présenté dans [11] a également essayé de trouver les valeurs L et M proche de $R^{-1} B_2 P$ et $K C_2$, respectivement, en utilisant la méthode itérative. Il est clair que les tels variables introduites sont données avec des informations sur P et K dans la dernière étape du processus itératif, c'est-à-dire, en trouvant P , ils l'utilisent dans la dernière étape pour fixer les valeurs de ces variables supplémentaires.

En fait, les informations sur P peuvent être utilisés directement sans introduire des variables supplémentaires. Par exemple, lorsque nous déterminons P dans cette étape, nous pouvons l'employer pour déduire K à l'étape suivante. Dans le cas contraire K est également utilisées pour

calculer P , et ainsi de suite. Par conséquent, ces variables dans les procédures du processus itératif dans [10] et [11] sont inutiles et l'itération peut être réalisée entre P et K directement.

Sur la base de cette idée, il est proposé un nouvel algorithme comme suit.

Comme il est mentionné dans [10], si

$$P(A + B_2KC_2) + (A + B_2KC_2)^T P - \alpha P < 0 \quad (5.5)$$

est vérifié alors, la matrice du système en boucle fermée $A + B_2KC_2$ a ses valeurs propres strictement dans la partie gauche de la ligne $\alpha/2$ du plan complexe. Si un $\alpha \leq 0$ satisfaisant (5.5) peut être trouvée, le problème de la stabilisation SOF est résolu.

La clé de notre algorithme est de trouver une valeur initiale P . Le P qui satisfait (5.4) ne peut pas être calculée à partir de LMI dû à la valeur inconnu de K .

5.3.1 Initialisation de la matrice P

En fixant $V_1 = PB_2F$, alors (5.4) devient

$$PA + A^T P + V_1 C_2 + C_2^T V_1^T < 0. \quad (5.6)$$

Cependant, P ne donne pas toute l'information, puisque le terme B_2 n'apparaît pas dans l'inégalité (5.6). D'autre part, (5.4) est transformée à l'inégalité suivante par une pré et post-multiplication par la matrice P^{-1} , et en prenant $P^{-1} = L$:

$$(A + B_2KC_2)L + L(A + B_2KC_2)^T < 0 \quad (5.7)$$

On mettant $V_2 = KC_2L$ alors (5.7) devient :

$$AL + LA^T + B_2V_2 + V_2^T B_2^T < 0 \quad (5.8)$$

Conformément à l'idée de la méthode de linéarisation dans [12], La relation $L=P^{-1}$ doit être aussi exprimé sous forme d'une contrainte LMI. Les problèmes comportant ce type de contraintes peuvent s'exprimer de manière équivalente sous la forme de problème comprenant une condition LMI associée à une condition de rang.

$L = P^{-1}$ donne $PL = I$, qui est LMI suivante:

$$\begin{bmatrix} P & I \\ I & L \end{bmatrix} \geq 0 ; \quad (5.9)$$

avec

$$\text{rang} \begin{bmatrix} P & I \\ I & L \end{bmatrix} = n$$

et la version linéarisée trace (PL) est minimisée. Ensuite, un algorithme du processus itératif pour trouver le P initiale est donné comme suit.

Algorithme 1 :

- Etape 1) Mettre $i=1$ et $P_0=I$ et $L_0=I$.
- Etape 2) Dériver P_i et L_i par la résolution du problème d'optimisation avec les variables P_i, L_i, V_1 et V_2 :

OPI : Minimiser Trace $(P_i * L_{i-1} - L_i * P_{i-1})$ sous les contraintes LMI suivantes :

$$P_i A + A^T P_i + V_1 C_2 + C_2^T V_1^T < 0$$

$$A L_i + L_i A^T + B_2 V_2 + V_2^T B_2^T < 0$$

$$\begin{bmatrix} P_i & I \\ I & L_i \end{bmatrix} \geq 0$$

- Etape 3) Si Trace $(P_i L_i) - n < \varepsilon_1$, est une tolérance prescrite, alors $P=P_i$ initial est trouvé, **s'arrêter**.
- Etape 4) Si la différence entre les deux itérations satisfait
Trace $(P_i L_{i-1}) - \text{Trace} (L_i P_{i-1}) < \varepsilon_2$, est une tolérance prescrite, un P initial
 $P = P_0$ ne peut pas être trouvé, **s'arrêter**.
- Etape 5) mettre $i=i+1$, aller à l'étape 2).

Si le P initial donné par Algorithme 1 ne peut pas être déterminé, alors le control par retour de sortie statique (SOF) pour le système (5.1) avec $w(t)=0$ peut ne pas avoir de solution. D'autre part si P est trouvé, alors un ILMI algorithme qui stabilise le système (5.1) avec $w(t)=0$ utilisant SOF peut être établi comme ci dessous.

5.3.2 Résolution du problème SOF

L'algorithme de calcul du gain statique de retour de sortie est basé sur la recherche du α ($\alpha \leq 0$), tel que l'inégalité (5.5) soit satisfaite.

Premier mot, si P est connu l'inégalité (5.5) est linéaire en les variables α et K , dans ce cas on a un problème de minimisation d'une fonction de coût linéaire.

Par contre, si K est donné et P est inconnu l'inégalité (5.5) s'écrit:

$$(5.5) \Leftrightarrow \begin{cases} P > 0 \\ \alpha P - (P(A + B_2KC_2) + (A + B_2KC_2)^T P) > 0 \end{cases} \quad (5.10)$$

$$\Leftrightarrow \begin{cases} P > 0 \\ \alpha P - G > 0 \end{cases} \quad (5.11)$$

Avec $G = P(A + B_2KC_2) + (A + B_2KC_2)^T P$

En posant

$$P = \sum_{i=1}^{\frac{n(n+1)}{2}} \zeta_i P_i, \quad G(\zeta) = \sum_{i=1}^{\frac{n(n+1)}{2}} \zeta_i G_i, \quad G_i = P_i(A + B_2KC_2) + (A + B_2KC_2)^T P_i$$

Où ζ_i : le vecteur de décision, vecteur de R^m .

P_i : des matrices symétriques données de $R^{n \times n}$.

$$i=1, \dots, \frac{n(n+1)}{2}.$$

La recherche de α s'écrit alors :

$$\begin{aligned} & \text{Minimiser} && \alpha \\ & \text{pour } \alpha \in R, \zeta \in R^m \\ & \text{contraint par} && \alpha P(\zeta) - G(\zeta) > 0 \\ & && P(\zeta) > 0 \end{aligned}$$

Ce qui est un problème de minimisation de la valeur propre généralisée maximale.

Et d'autre part, si ce dernière n'est pas résolvable, c à d, on ne peut pas trouver $\alpha \leq 0$ solution de ce problème, l'inégalité (5.5) peut être reformulé par la minimisation de trace de P qui est considéré comme un problème de minimisation d'une fonction de coût linéaire.

Algorithme 2 :

- Etape 1) Mettre $i=1$ et $P_1=P$ comme il a été obtenu par l'Algorithme 1.
- Etape 2) Résoudre le problème d'optimisation suivant pour K avec P_i donné :

OP1 : minimiser α_i sujet aux contraintes LMI suivantes :

$$P_i(A + B_2KC_2) + (A + B_2KC_2)^T P_i - \alpha_i P_i < 0 \quad (5.12)$$

- Etape 3) Si $\alpha_i \leq 0$, K est le gain stabilisant SOF, **s'arrêter**.
- Etape 4) mettre $i=i+1$. résoudre le problème d'optimisation suivant pour P_i avec K donné :

OP2 : minimiser α_i sujet aux contraintes LMI (5.12).

- Etape 5) Si $\alpha_i \leq 0$, K est le gain stabilisant SOF, **s'arrêter**.
- Etape 6) Résoudre le problème d'optimisation suivant pour P_i avec K et α_i donnés :

OP3 : minimiser Trace (P_i) sujet aux contraintes LMI (5.12).

- Etape 7) Si $\| P_i - P_{i-1} \| / \| P_i \| < \delta$, est une tolérance prescrite, aller à l'étape 8), si non mettre $i=i+1$ et $P_i = P_{i-1}$, puis aller à l'étape 2).
- Etape 8) Le système n'est peut être pas stabilisable par SOF, **s'arrêter**.

Remarque 1: Les discussions sur la procédure itérative et de la convergence de l'Algorithme 2 sont sur [10].

Remarque 2 : Si $C_2 = I$, le problème de stabilisation SOF se réduit à une stabilisation par retour d'état. En fait, le P initial obtenu à partir de l'algorithme 1 sera aussi la solution du problème de stabilisation par retour d'état puisque le gain K de retour d'état peut être dérivé par

$K = V_2 L^{-1} = V_2 P$. D'autre part, le K peut être aussi obtenu directement par **OP1** dans l'Algorithme 2 où le P initial est donné.

Remarque 3 : l'algorithme présenté dans notre travail est différente de celle donné dans [12]. Le critère d'arrêt dans [12] est donné en terme de ε_{eof} qui dépend de α et β sélectionnés, avec α et β sont suffisamment petits. Si le choix de α et β est mauvais, l'Algorithme ne converge pas.

Dans notre procédure, l'algorithme 1 définit un P initial pour ε_1 donné, où la valeur de ε_1 sélectionnée dans l'Algorithme 1 n'est pas cruciale, puisque la matrice P initial obtenue n'est pas la solution finale, mais elle va élaborer dans l'Algorithme 2. Avec le P initial, l'Algorithme 2 nous donne un gain K qui peut nous garantir la stabilité du système en boucle fermée. La condition pour qu'elle s'arrête est $\alpha < 0$, qui n'est pas besoin de spécifié préalablement. Globalement, on peut dire que notre procédure est plus facile à utiliser.

5.4 Exemples de simulation

Pour voir l'intérêt de la méthode proposée dans ce travail, on utilise les applications sur les exemples traités dans les références [10], [14] et [15], afin de pouvoir faire une comparaison entre les différents résultats.

❖ Exemple 1

Cas SISO [10]

On considère le problème de stabilisation par retour de sortie du système (5.1) avec $w(t)=0$ et les matrices de paramètres suivantes :

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_2 = [1 \quad \beta]$$

Le système en boucle ouverte est instable car les valeurs propres sont (1, -1). Les résultats de calcul d'un correcteur de retour de sortie statique (SOF) pour les deux méthodes (la méthode proposée dans [10], et la méthode que l'on étudie dans ce travail) sont donnés dans le tableau (5.1). Il est noté que le nombre d'itérations sous forme $l+k$, dans le tableau veut exprimer que l est

le nombre d'itérations de l'algorithme 1 pour trouver une valeur initial P et k est le nombre d'itération de l'algorithme 2 pour trouver un gain SOF.

β	Méthode	Gain de retour	Pôles	Nbre d'itéra.	α
15	[10]	$K = -0.7369$	$-0.3684 \pm j3.1492$	15	-0.0377
	proposée	$K = -0.1333$	$-0.0667 \pm j0.9978$	2+1	-7.8046×10^{-4}
100	[10]	$K = -0.1748$	$-0.0874 \pm j4.0586$	366	-0.0017
	proposée	$K = -0.0200$	$-0.0100 \pm j1.0000$	2+1	-0.0012

Tableau 5.1 résultats de l'exemple 1 (cas SISO)

On remarque dans ce tableau que la méthode à laquelle on s'intéresse ici a une meilleure vitesse de convergence que la méthode dans [10].

Cas MIMO [14]

Maintenant nous allons considérer le cas de la stabilisation par retour de sortie dans le cas multivariable, avec les matrices paramètres suivants :

$$A = \begin{bmatrix} -4 & -2 & -8 & 5 & -1 & -8 & 4 \\ -9 & -7 & -6 & -3 & -2 & 2 & 6 \\ -7 & -3 & 7 & 5 & -2 & 2 & -1 \\ -6 & -3 & 8 & 1 & 2 & 3 & -7 \\ 0 & -5 & 6 & -3 & -4 & 6 & 1 \\ 2 & 8 & -4 & 6 & -9 & -2 & -4 \\ 5 & 8 & 3 & 1 & 9 & -6 & 3 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} -3.9 & 2 & 0.1 & -2.5 & -1 & 2.5 & -1 \\ 0.5 & 0.5 & -1 & -0.5 & 1 & 2 & -0.05 \end{bmatrix}^T,$$

$$C_2 = \begin{bmatrix} 3 & 6 & -5 & -2 & -1 & -7 & 5 \\ -1 & -4 & -7 & -1 & -6 & -5 & -3 \end{bmatrix}.$$

Après plus 250 itérations, L'algorithme de [14] converge, par contre l'Algorithme 1 nous donne P initial après 4 itérations. Puis, un gain stabilisant à retours de sortie $K = \begin{bmatrix} -0.8871 & 4.9310 \\ -0.6576 & 0.9869 \end{bmatrix}$ est trouvé en utilisant l'algorithme 2, juste après 2 itérations.

Dans ce cas, $\alpha = -0.7481$ et les valeurs propres de la boucle fermée sont $-7.8846 \pm j36.0334$, $-0.6354 \pm j12.2411$, -0.3742 , $-4.9779 \pm j6.3825$. Donc on peut dire qu'il converge plus rapidement que celui de [14].

❖ Exemple 2 (synthèse H^∞)

Dans cet exemple nous allons utiliser une méthode présentée dans le chapitre précédent pour employer les deux algorithmes précédents afin de résoudre le problème du contrôleur à retour de sortie stabilisant étendu au cas du problème de contrôle H^∞ . Le but de ce contrôleur SOF H^∞ est de trouver un contrôleur SOF ($u(t)=Ky(t)$) de telle façon que la fonction de transfert du système en boucle fermée satisfasse les conditions de contraintes de la norme H^∞ :

$$\|T_{zw}(s)\|_\infty < \gamma, \text{ pour } \gamma > 0 \quad (5.13)$$

L'équation (5.13) peut être représentée comme la matrice inégalité de [10] :

$$\begin{bmatrix} PA_{cl} + A_{cl}^T P & PB_{cl} & C_{cl}^T \\ B_{cl}^T P & -\gamma I & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma I \end{bmatrix} < 0 \quad (5.14)$$

ou

$$A_{cl} = A + B_2 K C_2$$

$$B_{cl} = B_1 + B_2 K D_{21}$$

$$C_{cl} = C_1 + D_{12} K C_2$$

$$D_{cl} = D_{11} + D_{12} K D_{21}$$

De la même façon qu'avec l'algorithme 2, il a été proposé ici un algorithme qui trouve la solution de la matrice inégalité (5.14) pour un $\gamma > 0$ donné. On s'appuie comme pour l'algorithme 1, sur un autre algorithme pour trouvé une valeur initiale de P pour le problème du SOF H^∞ .

Algorithme 3 :

- Etape 1) Mettre $i=1$ et $P_0=I$ et $L_0=I$.
- Etape 2) Dériver P_i et L_i par la résolution du problème d'optimisation avec les variables P_i, L_i, V_1 et V_2 :

OPI : Minimiser Trace ($P_i * L_{i-1} - L_i * P_{i-1}$) sous les contraintes LMI suivantes :

$$\begin{bmatrix} P_i A + A^T P_i + V_1 C_2 + C_2^T V_1 & P_i B_1 + V_1 D_{21} & C_1^T + C_2^T K^T D_{12}^T \\ B_1^T P_i + D_{21}^T V_1^T & -\gamma I & D_{11}^T + D_{21}^T K^T D_{12}^T \\ C_1 + D_{12} K C_2 & D_{11} + D_{12} K D_{21} & -\gamma I \end{bmatrix} < 0 \quad (5.15)$$

$$\begin{bmatrix} A L_i + L_i A^T + B_2 V_2 + V_2^T B_2^T & B_1 + B_2 K D_{21} & L_i C_1^T + V_2^T D_{12}^T \\ B_1^T + D_{21}^T K^T B_2^T & -\gamma I & D_{11}^T + D_{21}^T F^T D_{12}^T \\ C_1 L_i + D_{12} V_2 & D_{11} + D_{12} F D_{21} & -\gamma I \end{bmatrix} < 0$$

(5.16)

$$\begin{bmatrix} P_i & I \\ I & L_i \end{bmatrix} \geq 0 \quad (5.17)$$

- Etape 3) Si Trace ($P_i L_i$) - $n < \varepsilon_1$, est une tolérance prescrite, alors $P=P_i$ initial est trouvé, **s'arrêter**.
- Etape 4) Si la différence entre les deux itérations satisfait Trace ($P_i L_{i-1}$) - Trace ($L_i P_{i-1}$) $< \varepsilon_2$, est une tolérance prescrite, un P initial $P = P_0$ ne peut pas être trouvé, **s'arrêter**.

Etape 5) mettre $i=i+1$, aller à l'étape 2).

Après la détermination de P initial, l'algorithme de calcul du gain statique de retour de sortie pour la commande H^∞ est comme suit :

Algorithme 4:

- Etape 1) Mettre $i=1$ et $P_1=P$ comme il a été obtenu par l'Algorithme 1.
- Etape 2) Résoudre le problème d'optimisation suivant pour K avec P_i donné :

OPI : minimiser α_i sujet aux contraintes LMI suivantes :

$$\begin{bmatrix} P_i(A+B_2KC_2)+(A+B_2KC_2)^T P_i - \alpha_i P_i & P_i B_1 + P_i B_2 K D_{21} & C_1^T + C_2^T K^T D_{12}^T \\ B_1^T P_i + D_{21}^T K^T B_2^T P_i & -\gamma I & D_{11}^T + D_{21}^T K^T D_{12}^T \\ C_1 + D_{12} K C_2 & D_{11} + D_{12} K D_{21} & -\gamma I \end{bmatrix} < 0 \quad (5.18)$$

- Etape 3) Si $\alpha_i \leq 0$, K est le gain stabilisant SOF H_∞ , **s'arrêter**.
- Etape 4) mettre $i=i+1$. résoudre le problème d'optimisation suivant pour P_i avec K donné :

OP2 : minimiser α_i sujet aux contraintes LMI (5.18).

- Etape 5) Si $\alpha_i \leq 0$, K est le gain stabilisant SOF H_∞ , **s'arrêter**.
- Etape 6) Résoudre le problème d'optimisation suivant pour P_i avec K et α_i donnés :

OP3 : minimiser Trace (P_i) sujet aux contraintes LMI (5.18).

- Etape 7) Si $\|P_i - P_{i-1}\| / \|P_i\| < \delta$, est une tolérance prescrite, aller à l'étape 8), si non mettre $i=i+1$ et $P_i = P_{i-1}$, puis aller à l'étape 2).
- Etape 8) Le système n'est peut être pas stabilisable par SOF H_∞ , **s'arrêter**.

Donc, on considère le système (5.1) avec les paramètres suivants : [11]

$$A = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.010 & 0.0024 & -4.0208 \\ 0.1002 & 0.3680 & -0.7070 & 1.4200 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.5200 & 4.4900 \\ 0 & 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D_{11} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad D_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} 0 & 0.1 & 0 \\ 0 & 1 & 0.1 \end{bmatrix}$$

La norme de SOF H_∞ obtenu dans [11] est 1.183. Pour $\gamma = 1.144$, la matrice P initiale est obtenue comme suit :

$$P_1 = \begin{bmatrix} 0.7130 & -0.4054 & -0.3283 & -1.1658 \\ -0.4054 & 2.2769 & 1.4029 & 1.2100 \\ -0.3283 & 1.4029 & 1.5941 & 1.0533 \\ -1.1658 & 1.2100 & 1.0533 & 3.2305 \end{bmatrix}$$

par l'algorithme 3 après 24 itérations. Alors le SOF H_∞ converge vers 1.144 après 2 itérations en utilisant l'algorithme 4, et le gain SOF résultant est :

$$K = \begin{bmatrix} 0.0976 & -3.8054 \\ -0.4191 & 4.6958 \end{bmatrix}$$

Dans ce cas, $\alpha = -8.2776 \times 10^{-7}$, et les pôles sont - 50.1599, - 0.2781, - 0.2431 $\pm j1.3534$.

❖ Exemple 3

a) contrôleur PID

Motivé par sa popularité dans l'industrie on considère ici le contrôleur PID suivant :

$$u(t) = F_1 y(t) + F_2 \int_0^t y(\theta) d\theta + F_3 \dot{y}(t) \quad (5.19)$$

au lieu du contrôleur SOF présenté par l'équation (5.2), où $F_1, F_2, F_3 \in R^{m \times p}$ sont les gains des matrices désignées. Sans perdre de généralité, D_{21} est prise égale à zéro.

Il est noté qu'une nouvelle méthode proposée dans [13] transforme le problème du calcul d'un PID en un problème de contrôle SOF. Afin de pouvoir utiliser la méthode proposée dans [13] on a besoin de faire l'hypothèse suivante :

Hypothèse 5.2

La matrice $I - F_3 C_2 B_2$ est inversible.

$$\text{Dénotons } \bar{x}(t) = \begin{bmatrix} x(t) \\ \int_0^t y(\theta) d\theta \end{bmatrix} \text{ et } \bar{y}(t) = \begin{bmatrix} C_2 x(t) \\ \int_0^t y(\theta) d\theta \\ C_2 A x(t) \end{bmatrix}$$

La combinaison de (5.1) et de (5.19) donnent le système suivant :

$$\begin{cases} \dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}_1 w(t) + \bar{B}_2 u(t) \\ z(t) = \bar{C}_1 \bar{x}(t) + \bar{D}_{11} w(t) + \bar{D}_{12} u(t) \\ \bar{y}(t) = \bar{C}_2 \bar{x}(t) \end{cases} \quad (5.20)$$

avec

$$\bar{A} = \begin{bmatrix} A & 0 \\ C_2 & 0 \end{bmatrix}, \quad \bar{B}_1 = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \quad \bar{B}_2 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix}$$

$$\bar{C}_2 = [\bar{C}_{21}^T \quad \bar{C}_{22}^T \quad \bar{C}_{23}^T]^T, \quad \bar{C}_1 = [C_1 \quad 0]$$

$$\bar{C}_{21} = [C_2 \quad 0], \quad \bar{C}_{22} = [0 \quad I], \quad \bar{C}_{23} = [C_2 A \quad 0]$$

$$\bar{D}_{11} = D_{11}, \quad \bar{D}_{12} = D_{12}$$

Où $\bar{x}(t) \in R^{n+p}$, $u(t) \in R^m$, $z(t) \in R^q$ et $y(t) \in R^{3p}$, et les matrices \bar{A} , \bar{B}_1 , \bar{B}_2 , \bar{C}_1 , \bar{C}_2 , \bar{D}_{11} et \bar{D}_{12} sont des matrices constantes ont des dimensions compatibles.

Et Le contrôleur se réduit à:

$$u(t) = \bar{F} \bar{y}(t) \quad (5.21)$$

avec $\bar{F} = [\bar{F}_1 \quad \bar{F}_2 \quad \bar{F}_3]$, où $\bar{F}_i = (I - F_3 C_2 B_2)^{-1} F_i$ $i=1,2,3$

Une fois que $\bar{F} = [\bar{F}_1 \quad \bar{F}_2 \quad \bar{F}_3]$ déterminer, le contrôleur PID qui nous cherchons peut être retrouver par :

$$\begin{aligned}
F_3 &= \bar{F}_3(I + C_2 B_2 \bar{F}_3)^{-1} \\
F_2 &= (I - F_3 C_2 B_2) \bar{F}_2 \\
F_1 &= (I - F_3 C_2 B_2) \bar{F}_1
\end{aligned} \tag{5.22}$$

Lemme [13]

La matrice $(I + C_2 B_2 \bar{F}_3)$ est inversible si et seulement si l'hypothèse 5.2 est toujours vérifiée, où F_3 et \bar{F}_3 sont reliés entre eux par :

$$\bar{F}_3 = (I - F_3 C_2 B_2)^{-1} F_3 \quad \text{ou} \quad F_3 = \bar{F}_3 (I + C_2 B_2 \bar{F}_3)^{-1}$$

Sous l'hypothèse 5.2 on peut facilement remarquer que les algorithmes 1 et 2 peuvent être employés pour dériver les gains F_1, F_2 et F_3 d'un contrôleur PID. Simultanément, on peut utiliser les algorithmes 3 et 4 pour dériver le contrôleur PID H_∞ pour une performance $\gamma > 0$ donnée.

On considère l'exemple proposé dans [13] avec la représentation (5.1), où :

$$A = \begin{bmatrix} -0.0266 & -36.6170 & -18.8970 & -32.0900 & 3.2509 & -0.7626 \\ 0.0001 & -1.8997 & 0.9831 & -0.0007 & -0.1708 & -0.0050 \\ 0.0123 & 11.7200 & -2.6316 & 0.0009 & -31.6040 & 22.3960 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -30.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30.0000 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 30 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 30 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0], \quad C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D_{11} = [0] \quad D_{12} = [1 \ 1] \quad D_{13} = [1 \ 1]$$

Le tableau (5.2), regroupe tout les résultats obtenus par l'application des trois méthodes de recherche d'un gain statique pour le retour de sortie données dans notre travail, et ces résultats sont comparés avec ceux des références [10] et [13] :

b) contrôleur PI

Dans le cas d'un correcteur PI, la commande est donnée par :

$$u(t) = F_1 y(t) + F_2 \int_0^t y(\theta) d\theta \Rightarrow u(t) = \bar{F} \bar{y}(t) \tag{5.23}$$

De la même façon, et plus simple que le PID, la recherche d'un correcteur PI pour un système décrit par (5.20) se fait avec les paramètres suivants :

$$\begin{aligned} \bar{A} &= \begin{bmatrix} A & 0 \\ C_2 & 0 \end{bmatrix}, & \bar{B}_1 &= \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, & \bar{B}_2 &= \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \\ \bar{C}_2 &= [\bar{C}_{21}^T \quad \bar{C}_{22}^T], & \bar{C}_1 &= [C_1 \quad 0] \\ \bar{C}_{21} &= [C_2 \quad 0], & \bar{C}_{22} &= [0 \quad I] \\ \bar{D}_{11} &= D_{11}, & \bar{D}_{12} &= D_{12} \end{aligned}$$

Où

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ \int_0^t y(\theta) d\theta \end{bmatrix} \text{ et } \bar{y}(t) = \begin{bmatrix} C_2 x(t) \\ \int_0^t y(\theta) d\theta \end{bmatrix}$$

Dans ce cas la, le correcteur est déterminé directement par :

$$\bar{F} = [\bar{F}_1 \quad \bar{F}_2] = F$$

Méthode	Problème	Gain de retour	pôles	Perf.	Nbre d'itéra	\square
[11], [13]	SOF (stab.)	$F = \begin{bmatrix} 7.0158 & -4.3414 \\ 2.1396 & -4.4660 \end{bmatrix}$	-0.0475 ± j0.0853 -0.7576 ± j10.7543 -29.2613, -33.6825	Stab.	20	-1.9×10^{-4}
proposée	SOF (stab.)	$F = \begin{bmatrix} 0.5991 & 0.2703 \\ -0.2074 & -0.0248 \end{bmatrix}$	-1.3352 ± 4.5681i -0.7611, -0.0679 -31.0528, -30.0057	Stab.	3+1	-0.0349
[13]	PID (stab.)	$F_1 = \begin{bmatrix} 10.1359 & -1.7947 \\ 6.9912 & -9.4140 \end{bmatrix}$ $F_2 = \begin{bmatrix} 0.3817 & -0.6939 \\ 0.6528 & -1.1978 \end{bmatrix}$ $F_3 = \begin{bmatrix} 2.6162 & -1.4722 \\ 0.8212 & -1.6284 \end{bmatrix}$	-0.0003, 0.0243 -0.0774, -37.0751 -3.07 ± j0.46 -17.38 ± j34.73	Stab.	1	-4.4×10^{-4}
proposée	PID (stab.)	$F = \begin{bmatrix} 25.9668 & 2.3710 \\ 11.6225 & -0.9431 \end{bmatrix}$ $F = \begin{bmatrix} 0.2729 & -0.5054 \\ 0.2448 & -0.4537 \end{bmatrix}$ $F = \begin{bmatrix} 11.6688 & -10.2690 \\ 6.3776 & -6.9539 \end{bmatrix}$	-84.0459 -17.7468 ± 29.9127i -5.0486, -0.6499 -3.0759 × 10 ⁻⁵ -0.0399, -0.0276	Stab.	2+3	-5.2743×10^{-5}
proposée	PI (stab)	$F_1 = \begin{bmatrix} 0.7020 & 0.1650 \\ -0.0764 & -0.1940 \end{bmatrix}$ $F_2 = \begin{bmatrix} 0.0896 & -0.1653 \\ 0.1272 & -0.2353 \end{bmatrix}$	-31.0804, -29.9843 -1.3173 ± j 4.6457 -0.7595, -0.0589 0.0400, -1.4126 × 10 ⁻⁴	Stab.	2+1	-2.8683×10^{-5}
[11], [13]	SOF (H _∞)	$F = \begin{bmatrix} 0.2838 & 0.0313 \\ -0.8725 & -0.0289 \end{bmatrix}$	-0.1042 ± j0.1536 -1.6525 ± j4.3864 -30.0027, -31.0378	H _∞ < 5	26	-4.5×10^{-2}
proposée	SOF (H _∞)	$F = \begin{bmatrix} 1.5148 & 0.4064 \\ -2.5148 & -0.4064 \end{bmatrix}$	-0.2972 ± 10.2343i -0.0410, -0.3423 -33.5662, -30.0140	H _∞ < 0.323	88+3	-0.0048
[13]	PID (H _∞)	$F_1 = \begin{bmatrix} 442.17 & 221.64 \\ -188.84 & -104.44 \end{bmatrix}$ $F_2 = \begin{bmatrix} 0.3845 & -0.5019 \\ 0.1068 & -0.3373 \end{bmatrix}$ $F_3 = \begin{bmatrix} 48.03 & -31.85 \\ -19.31 & 8.80 \end{bmatrix}$	-0.0001, -0.0020 -0.0050, 0.7200 -19.9100, 191.0000 -48.02 ± j77.79	H _∞ < 5	3	-4.9×10^{-4}
proposée	PID (H _∞)	$F = \begin{bmatrix} 56.7406 & 1.3343 \\ -12.1206 & -2.1600 \end{bmatrix}$ $F = \begin{bmatrix} 0.0148 & -0.0077 \\ 0.0014 & -0.0041 \end{bmatrix}$ $F = \begin{bmatrix} 28.2831 & -27.6218 \\ -4.9638 & 3.5831 \end{bmatrix}$	-173.3549 -14.8136 ± 26.1290i -5.0514 -0.6783 -0.0219 -0.0020 -1.9722*10 ⁻⁴	H _∞ < 1.001	2+9	-3.7637×10^{-8}
proposée	PI (H _∞)	$F_1 = \begin{bmatrix} 1.1959 & 0.3469 \\ -2.1760 & -0.3664 \end{bmatrix}$ $F_2 = \begin{bmatrix} 0.1127 & 0.2372 \\ -0.0835 & -0.2450 \end{bmatrix}$	-33.0813 -30.0127 -0.4361 ± j9.3223 -0.2823 ± j0.4223 -0.0259 -0.0012	H _∞ < 1.001	62+3	-7.0214×10^{-8}

Tableau 5.2 résultats de l'exemple 3
(Contrôleurs SOF, PID et PI)



5.5 Conclusion

L'essentielle de la méthode proposée dans ce chapitre est d'éviter complètement l'introduction des variables supplémentaires comme d'autres méthodes [10], grâce à ça on peut dire que la taille des LMIs ne devient pas très grande. Particulièrement, il y'a une séparation de l'algorithme, premier mot il faut dériver les valeurs initiales, puis établir l'algorithme de la stabilisation par retour de sortie statique (SOF) et la commande H^∞ du système.

Dans ce travail, on fait la transformation d'un problème de recherche d'un PID et d'un PI en un problème de la commande par retour de sortie statique.

On peut avoir la différence entre les deux correcteurs au niveau de la convergence des algorithmes, où le nombre des itérations pour dériver un PID est moins que celle pour un PI, et ce dernier est aussi mieux par rapport au contrôleur SOF.

Le principal objectif de cette nouvelle approche est la diminution des temps de calcul, et augmenter la vitesse de convergence des algorithmes, à cause de la diminution des tailles des matrices de LMI.

CONCLUSION GÉNÉRALE

CONCLUSION GENERALE

Le principe de notre travail est la description d'un algorithme itératif basé sur l'approche LMI qui permet à résoudre le problème de synthèse d'un gain de retour de sortie afin de garantir la stabilité de la boucle fermée d'un système.

La méthode proposée s'appuyant sur les fortes propriétés de l'optimisation convexe, permet en outre de pouvoir conclure à l'existence ou non d'une solution d'un problème. L'utilisation de l'optimisation convexe sous contraintes LMI nécessite une démarche systématique qui utilise les manipulations matricielles :

- Formulation du problème en terme d'optimisation ;
- Mise sous forme d'inégalités matricielles des contraintes d'optimisation ;
- Linéarisation de ces matrices en fonction des paramètres (changement de variable, lemme de Schur, lemme d'élimination...);
- Résolution numérique du problème avec par exemple une méthode de point intérieur ou la méthode des centres ;
- Reconstruction de correcteur.

Le retour de sortie statique (SOF) joue un rôle très important dans la théorie du contrôle et les applications. Un SOF gain qui stabilise le système n'est pas facile à déterminer. Les inégalités linéaires matricielles sont l'un des outils les plus efficaces dans la conception de contrôleurs. Beaucoup de méthodes de conception basée sur des LMI pour la conception de SOF ont été proposées. Dans ce contexte, une nouvelle variable a été introduite de telle sorte que la condition de stabilité devient suffisante. Deux méthodes ont introduit des matrices supplémentaires à l'étape actuelle avec les matrices trouvées à l'étape précédente. Avec des variables supplémentaires les dimensions de l'LMIs augmentent. Un nouvel algorithme pour résoudre le problème du contrôle à retour de sortie statique (SOF) est proposé dans le présent mémoire, une approche ILMI a été développée pour construire cette méthode. La méthode de stabilisation par retour de sortie statique (SOF) a été étendue pour résoudre aussi le problème du retour de sortie statique cas H_∞ contrôle. La qualité de cette méthode c'est qu'elle évite l'introduction de variables supplémentaires, qui nous le savons entraînent à la baisse les dimensions des LMIs si nous le comparons avec les résultats des travaux effectués dans les articles [13] et [14].

Cette méthode en particulier, a besoin de bonnes valeurs initiales, un autre algorithme pour calculer les valeurs initiales a été également mis au point. Les algorithmes sont également appliqués à la conception des contrôleurs PI et PID multivariables.

Des exemples Numériques montrent que les algorithmes proposés donne de meilleurs résultats que ceux existants et aussi que la convergence est plus rapide que celle des algorithmes déjà existant.

Dans ce travail nous avons étendu la méthode proposée au cas du contrôleur PI. Ainsi La comparaison des deux algorithmes PI et PID à pue être faites en même que les autres algorithmes.

Pour cela, Nous commençons dans ce document par la représentation des problèmes d'optimisation convexe, et comment les résoudre sous des contraintes du forme LMI, les trois solutionneurs appropriées de LMI sont : solution de faisabilité, solution par minimisation d'une fonction de coût linéaire et solution par minimisation des valeurs propres généralisées.

Quelques critères de robustesse sont illustrés, puisque la synthèse par optimisation H_∞ est repose sur les principes des méthodes fréquentielles, ces critères peuvent être formulé par des spécifications fréquentielles sur des fonctions de transferts en boucle fermée.

Pour but de garantie la stabilité d'un système en boucle fermée, la résolution d'un problème par retour de sortie statique a été traitée par les deux méthodes, formulation du problème par la théorie de Lyapunov, et la méthode de synthèse H_∞ .

Perspectives

Dans un prochain travail en pensé élaboré une autre méthode mise au point de la conception de SOF pour le cas du retour d'état statique et bien sur avec une amélioration de l'algorithme de recherche des valeurs initiales.

ANNEXE A

Fonctions de sensibilité et Description des incertitudes

A.1 Fonctions de sensibilité

On considère le système général bouclé (multivariable) suivant :

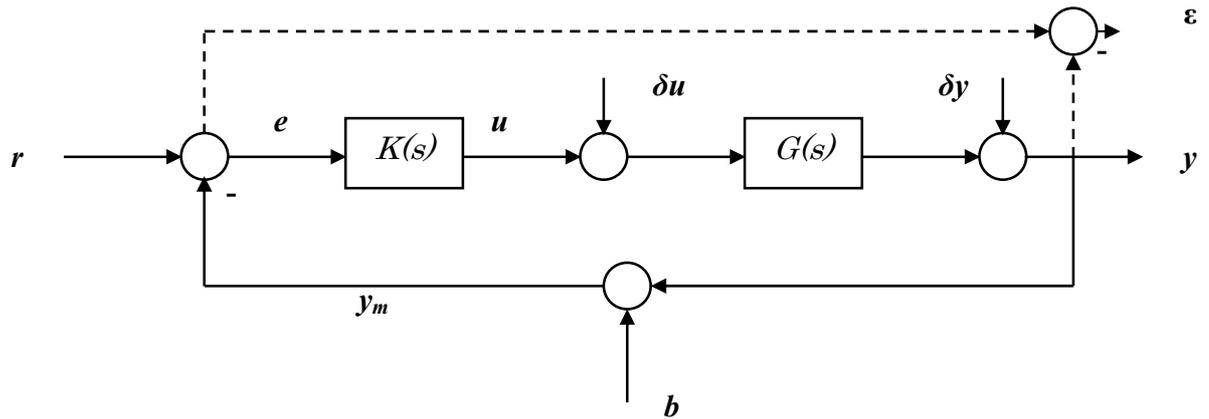


Figure A.1 Représentation générale d'un Système bouclé.

$G(s)$ Fonction de transfert du système

$K(s)$ Fonction de transfert du correcteur

r Signal de consigne ou de référence

u Signal de commande

y Signal de sortie (grandeur à réguler)

y_m Mesure de la sortie

δu Perturbation d'entrée

δy Perturbation de sortie

b Bruit de mesure

e Erreur d'asservissement

ε Erreur de suivi (non mesurable)

$$e = r - y_m = r - (y + b)$$

$$y = \delta y + G \cdot \delta u + GK \cdot e = \delta y + G \cdot \delta u + GK \cdot (r - y - b)$$

$$\Rightarrow y = S_s \cdot \delta y + S_s G \cdot \delta u + T_s \cdot (r - b)$$

$$\text{avec } S_s(s) = (I + GK)^{-1} \quad \text{La fonction de sensibilité en sortie} \quad (\text{A.1})$$

$$\text{et } T_s(s) = (I + GK)^{-1} GK = I - S_s \quad \text{La fonction de sensibilité complémentaire en sortie} \quad (\text{A.2})$$

$$e = r - (y + b) = r - \delta y - b - G \cdot \delta u - GK \cdot e$$

$$\Rightarrow e = S_s[r - \delta y - b] - S_s G \cdot \delta u$$



$$\begin{aligned}\varepsilon &= r - y = S_s.(r - \delta y + T_s.b - S_s G.\delta u \\ u &= K.(r - \delta y - b) - KG.\delta u - KG.u \\ \Rightarrow u &= -T_e.\delta u + S_e K.(r - \delta y - b)\end{aligned}$$

$$\text{avec } S_e(s) = (I + KG)^{-1} \quad \text{La fonction de sensibilité en entrée} \quad (\text{A.3})$$

$$\text{et } T_e(s) = (I + KG)^{-1}KG = I - S_e \quad \text{La fonction de sensibilité complémentaire en entrée} \quad (\text{A.4})$$

$$\begin{aligned}KS_s &= K(I + GK)^{-1} = S_e K = (I + KG)^{-1}K \\ KS_s G &= K(I + GK)^{-1}G = T_e = (I + KG)^{-1}KG \neq T_s\end{aligned}$$

A.2 Description des incertitudes

Dans le cadre de la commande robuste, il est nécessaire de préciser l'ensemble des modèles possibles représentant le système à commander. Un tel ensemble est défini par un modèle nominal et un domaine d'incertitude.

2.1 Structuration d'incertitudes

On distingue 2 types d'incertitudes

- Les incertitudes structurées : ce sont les incertitudes sur les paramètres de modèle.
- Les incertitudes non structurées : ce sont les incertitudes sur les réponses de systèmes.

- **Incertitudes structurées**

Il y a 2 types d'incertitudes structurées :

- Les incertitudes structurées à structuration forte : Les incertitudes sont données sur les paramètres physiques ($m = m_0 \pm \Delta m$).
- Les incertitudes structurées à structuration faible : Les incertitudes sont données sur les paramètres de modèle.

Exemples:

- Fonction de transfert :

$$G(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + 1} \quad (\text{A.5})$$



pour ($i = 1, 2, 3, \dots, n$), $a_i = a_i(\theta)$ où θ est un vecteur de paramètres physiques. Chaque a_i est de la forme $a_i = a_{i0} \pm \Delta a_i$.

- Modèle d'état :

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ z(t) &= Cx(t)\end{aligned}\tag{A.6}$$

$$A = A_0 + \Delta A$$

$$B = B_0 + \Delta B$$

$$C = C_0 + \Delta C$$

Cette description d'incertitude est moins précise que la structuration forte.

- **Incertitudes non structurées**

Ces incertitudes définissent des domaines d'appartenance d'une réponse de type :

- Réponse transitoire à un échelon.
- Réponse transitoire à une impulsion.
- Réponse fréquentielle.

Ces incertitudes sont facilement observables et quantifiables. De plus elles contiennent les incertitudes sur les grandeurs physiques (incertitudes structurées) et elles peuvent tenir compte des dynamiques négligées.

2.2 Représentations des incertitudes

On utilise les formes additives et les formes multiplicatives :

- **forme additive :**

$$G(s) = G_0(s) + \Delta_a(s)$$

avec

$$\bar{\sigma}[\Delta_a(jw)] \leq \delta_a(w), \quad \forall w > 0$$

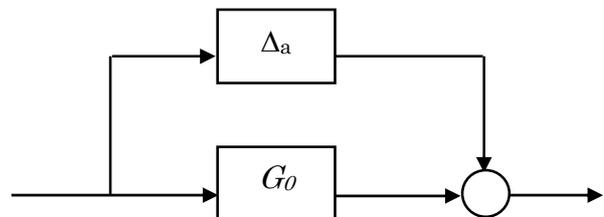


Figure A.2 Incertitudes additives



- forme multiplicative en entrée :

$$G(s) = G_0(s) [I + \Delta_e(s)]$$

$$\bar{\sigma} [\Delta_e(j\omega)] \leq \delta_e(\omega), \quad \forall \omega > 0$$

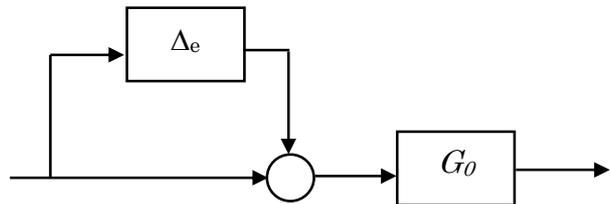


Figure A.3 Incertitudes multiplicatives en entrée

- forme multiplicative en sortie :

$$G(s) = [I + \Delta_s(s)] G_0(s)$$

$$\bar{\sigma} [\Delta_s(j\omega)] \leq \delta_s(\omega), \quad \forall \omega > 0$$

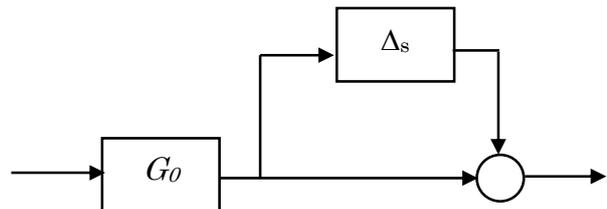


Figure A.4 Incertitudes multiplicatives en sortie

ANNEXE B



Quelques rappels sur les matrices

Définition B.1 soit $A \in C^{n \times n}$. Le nombre complexe λ est valeur propre de A si :

$$\exists v \in C^n, v \neq 0, Av = \lambda v$$

Le vecteur v est un vecteur propre associé à la valeur propre λ , A possède alors n vecteurs propres notées λ_i

(A), $i \in \{1, \dots, n\}$

Caractérisation λ est valeur propre de A si et seulement si $\det(\lambda I_n - A) = 0$ où I_n représente la matrice identité.

Définition B.2 Une matrice $A \in C^{n \times n}$ est inversible (ou non singulière) s'il existe une matrice, notée $A^{-1} \in C^{n \times n}$, telle que :

$$AA^{-1} = I_n$$

Propriétés

- Une matrice A est inversible si et seulement si $\forall i \in \{1, \dots, n\}, \lambda_i(A) \neq 0$. A est inversible si et seulement si $\det(A) \neq 0$ (car $\det(A) = \prod_{i=1}^n \lambda_i(A)$).
- $(AB)^{-1} = B^{-1}A^{-1}$

Définition B.3 La trace d'une matrice A (pas forcément symétrique) est la somme des éléments de sa diagonale. Notation : Trace (A).

Propriétés

- $\text{Trace}(A) = \sum_{i=1}^n \lambda_i(A)$
- $\text{Trace}(AB) = \text{Trace}(BA)$

Définition B.4 Matrice bloc diagonale :

$$A = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_r \end{bmatrix}$$

Où 0 correspond à la matrice nulle de dimension adéquate.



Propriété

$$\bullet \quad \{\lambda_i(A)\} = \bigcup_{j=1}^r \{\lambda_i(A_j)\}$$

Définition B.5 (valeurs propres généralisé) Soit $A \in R^{n \times n}$ et soit $B \in R^{n \times n}$, $B=B^T$ et $B > 0$. Les valeurs généralisées de (A, B) sont les valeurs propres de la matrice $M=B^{-1/2}A B^{-1/2}$. De façon équivalente, le nombre complexe λ est valeur propre généralisée de (A, B) si :

$$\exists v \in C^n, v \neq 0, Av = \lambda Bv$$

Caractérisation de la plus grande valeur généralisée de (A, B) :

$$\lambda_{max}(A, B) = \min_{\lambda \in R} \lambda$$

$$\lambda B - A \geq 0$$

Définition B.6 Si A est une matrice symétrique ($A = A^T$) alors on peut lui associer une forme quadratique, c'est-à-dire une fonction de R^n dans R qui à x associe $q_A(x) = x^T A x$.

Si $\forall x \neq 0, q_A(x) > 0$ ($q_A(x) \geq 0$) alors A est dite (semi) définie positive. Notation : $A > 0$ ($A \geq 0$).

Caractérisation des matrices définies positives

- A est symétrique (semi) définie positive si et seulement si ses valeurs propres sont positives (ou positives ou nulles).

$$\forall i \in \{1, \dots, n\}, \lambda_i(A) > 0 \quad (\lambda_i(A) \geq 0)$$

- Soit A_i la sous matrice obtenue à partir de la matrice A en ne conservant que les i premières lignes et les i premières colonnes. A est symétrique définie positive si et seulement si :

$$\forall i \in \{1, \dots, n\}, \det(A_i) > 0$$

Propriétés

- L'ensemble des matrices définies positives est un cône convexe.
- Si $A > 0$ alors $\exists T$ inversible $\in R^{n \times n}$, $A = T^T T$. Notation : $A^{1/2}$
- $\forall T$ inversible $\in R^{n \times n}$, $A > 0 \Leftrightarrow T^T A T > 0$.

**BMI (Inégalité Matricielle Bilinéaire) :**

La forme d'une contrainte BMI est :

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i + \sum_{k=1}^m \sum_{l=1}^m x_k x_l F_{kl} < 0$$

ANNEXE C

Régulateur PI et PID

C.1 Régulateur PID Un régulateur Proportionnel Intégral Dérivé (PID) est un organe de contrôle permettant d'effectuer une régulation en boucle fermée d'un système industriel. C'est le régulateur le plus utilisé dans l'industrie et permet de contrôler la grande majorité des procédés.

C.1.1 Principe général

Le PID permet 3 actions simultanées sur l'erreur consigne/mesure :

Une action Proportionnelle : l'erreur est multipliée par un gain G_r

Une action Intégrale : l'erreur est intégrée sur un intervalle de temps T_i

Une action Dérivée : l'erreur est dérivée suivant un temps T_d

Il existe plusieurs architectures possibles pour combiner les 3 effets (série, parallèle ou mixte), on prendra par exemple une architecture parallèle ici :

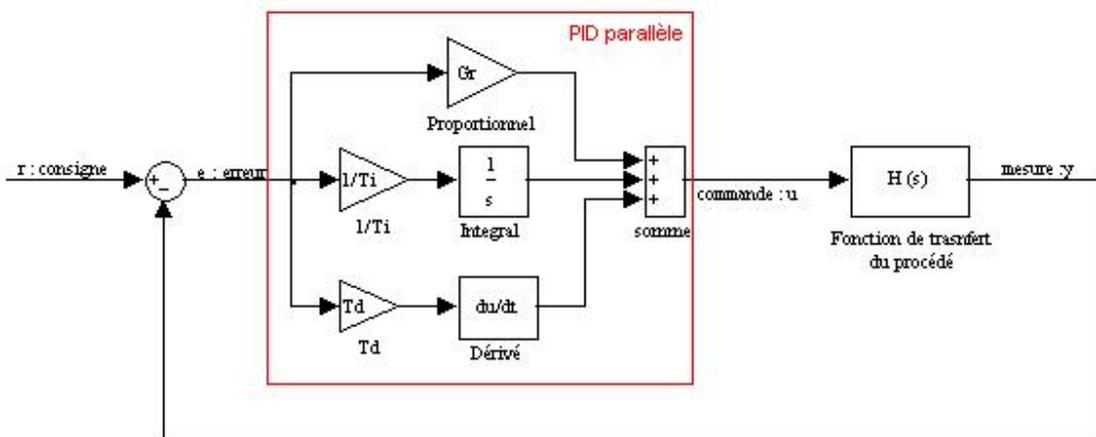


Figure C.1 schéma d'un PID parallèle

La fonction de transfert avec la transformée de Laplace du régulateur PID parallèle est la somme des 3 actions:

$$C(s) = G_r + T_d \cdot s + \frac{1}{T_i} \cdot \frac{1}{s} = \frac{G_r \cdot T_i \cdot s + T_d \cdot T_i \cdot s^2 + 1}{T_i \cdot s}$$



Les différents paramètres à trouver sont Gr , Td et Ti pour réguler le procédé ayant pour fonction de transfert $H(s)$. Il existe de nombreuses méthodes pour trouver ces paramètres. Cette recherche de paramètre est communément appelée réglage.

La fonction de transfert du contrôleur PID idéale est irréalisable car l'ordre du numérateur est supérieur à l'ordre du dénominateur. Dans la réalité, on rajoute toujours un paramètre alpha sur l'action dérivée de manière à obtenir un ordre 2 au numérateur et au dénominateur : Td

$$T_d \cdot s \rightarrow \frac{T_d \cdot s}{1 + \alpha \cdot T_d \cdot s}$$

avec $\alpha \ll 1$ On obtient alors une nouvelle fonction de transfert réalisable pour notre régulateur:

$$C(s) = \frac{(G_r \cdot T_i \cdot T_d \cdot \alpha + T_d \cdot T_i) \cdot s^2 + (G_r \cdot T_i + \alpha \cdot T_d) \cdot s + 1}{\alpha \cdot T_i \cdot T_d \cdot s^2 + T_i \cdot s}$$

Néanmoins, pour limiter les effets néfastes du bruit du signal de mesure sur le signal de commande u , on modère l'effet de l'action dérivée dans les régulateurs industriels en adoptant généralement $\alpha \sim 0,1$.

C.1.2 Réglage d'un PID

Le réglage d'un PID consiste à trouver les coefficients Gr , Td et Ti dans le but d'obtenir une réponse adéquate du procédé et de la régulation. L'objectif est d'être robuste, rapide, précis et de limiter les dépassements. La robustesse est sans doute le paramètre le plus important et délicat. On dit qu'un système est robuste si la régulation fonctionne toujours même si le modèle change un peu. Par exemple, les fonctions de transfert de certains procédés peuvent varier en fonction de la température ambiante ou de l'hygrométrie ambiante relativement à la loi de Pascal. Notre régulateur doit être capable d'assurer sa tâche même avec ces changements pour éviter de redévelopper un nouveau régulateur toutes les semaines.

La réponse type d'un procédé stable est la suivante :

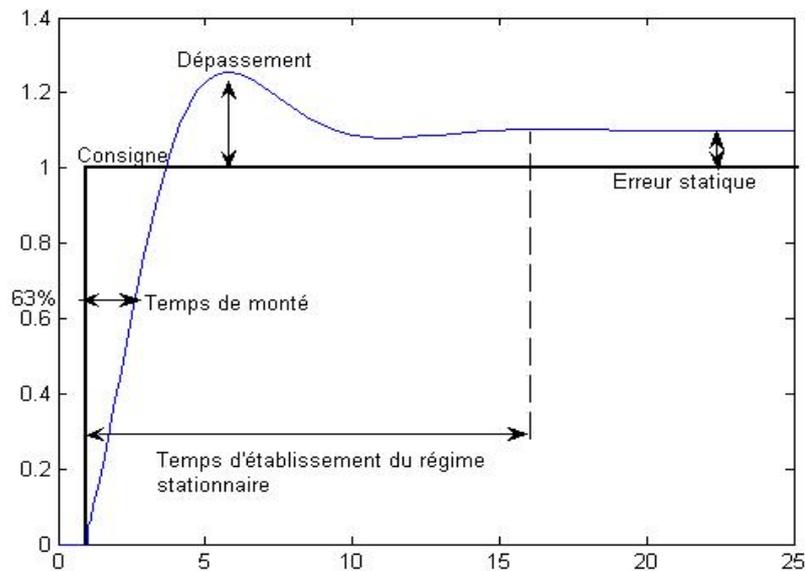


Figure C.2 Réponse d'un procédé stable

Les paramètres du PID influencent la réponse du système de la manière suivante :

- G_r : Lorsque G_r augmente, le temps de montée est plus court mais il y a un dépassement plus important. Le temps d'établissement varie peu et l'erreur statique se trouve améliorée.
- T_i : Lorsque $1/T_i$ augmente, le temps de montée est plus court mais il y a un dépassement plus important. Le temps d'établissement au régime stationnaire s'allonge mais dans ce cas on assure une erreur statique nulle.
- T_d : Lorsque T_d augmente, le temps de montée change peu mais le dépassement diminue. Le temps d'établissement au régime stationnaire est meilleur. Pas d'influences sur l'erreur statique.

L'analyse du système avec PID est très simple mais sa conception peut être délicate, voire difficile, car n'existe pas de manière unique pour résoudre ce problème. Il faut trouver des compromis, le régulateur idéal n'existe pas. En général on se fixe un cahier des charges à respecter sur la robustesse, le dépassement et le temps d'établissement du régime stationnaire. Les méthodes de réglage les plus utilisées sont la méthode Ziegler-Nichols, la méthode de P.Naslin (polynôme mormaux à amortissement réglage), la méthode du lieu de Nyquist inverse (utilise le diagramme de Nyquist).



C.2 Régulateur PI

Le régulateur PI (action proportionnelle intégrale) est une combinaison d'un régulateur P et d'un régulateur I.

Commande proportionnelle intégrale continue

$$u(t) = K_p \cdot \left(\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau \right)$$

Fonction de transfert en s :

$$C_{PI}(s) = K_p \left(1 + \frac{1}{T_i s} \right)$$

C.2.1 Structure d'un PI

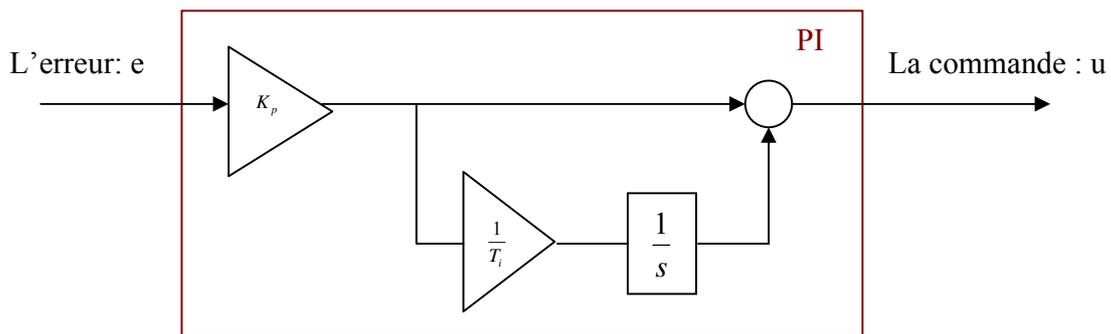


Figure C.3 schéma d'un PI

RÉFÉRENCES

RÉFÉRENCES

- [1] Boyd, El Ghaoui, Method of centers for minimizing generalized eigenvalue, *Linear Algebra and Applications*, pp. 63-111, 1993.
- [2] Nesterov, Nemirovskii, *Interior point polynomial methods in convex programming: theory and applications*, SIAM Publications, 1994.
- [3] Gahinet, Nemirovskii, Laub, Chilali, *The LMI Control Toolbox Use with Matlab*, The Mathworks Inc., 1995.
- [4] L. El Ghaoui, R. Nikoukhah, and F. Delebecque, *LMITOOL: A Front- End for LMI Optimization*, User's Guide, Feb. 1995; ftp to ftp.ensta.fr, under /pub/elghaoui/lmitool.
- [5] K. Zhou, J. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall, New Jersey, 1996.
- [6] A. Packard, J. Doyle, "The complex structured singular value", *Automatica*, Vol. 21, No. 1, pp. 71-109, 1993.
- [7] B.R. Barmish, *New Tools for Robustness of Linear Systems*, MacMillan Publishing Company, 1994.
- [8] S.P. Bhattacharyya, H. Chapellat, L.H. Keel, *Robust Control: The Parametric approach*, Prentice Hall Information and Sciences Series, 1995.
- [9] P. Colaneri, J.C. Geromel, A. Locatelli, *Control Theory and design: An RH_2 and RH_∞ perspective*, Academic Press, 1997.
- [10] Y. Y. Cao, J. Lam, and Y. X. Sun, *Static output feedback stabilization: An ILMI approach*, *Automatica*, vol. 34, pp. 1641–1645, 1998.
- [11] A. Fujimori, *Optimization of static output feedback using substitutive LMI formulation*, *IEEE Trans. Autom. Control*, vol. 49, no. 6, pp. 995–999, Jun. 2004.
- [12] L. E. Ghaoui, F. Oustry, and M. AitRami, *A cone complementarity linearization algorithm for static output-feedback and related problems*, *IEEE Transactions on Automatic Control*, Vol. 42, No. 8, Août 1997.
- [13] F. Zheng, Q. G. Wang, and T. H. Lee, *On the design of multivariable PID controllers via LMI approach*, *Automatica*, vol. 38, pp. 517–526, 2002.
- [14] J. T. Yu, *A convergent algorithm for computing stabilizing static output feedback gains*, *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 2271–2275, Dec. 2004.
- [15] D. S. Bernstein, *Some open problems in matrix theory arising in linear systems and control*, *Linear Alg. Appl.*, vol. 162-164, pp.409-432, 1992.

-
- 
- [16] S. Boyd, L E Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994, vol.15 SIAM Studies in Applied Mathematics.
- [17] J. C. Geromel, C. C. de Souza, and R. E. Skelton, *LMI numerical solution for output feedback stabilization*, in Proc. Amer. Control Conf., 1994, pp. 40–44.
- [18] C. Lin, Q. G. Wang, and T. H. Lee, “An improvement on multivariable PID controller design via iterative LMI approach,” *Automatica*, vol. 40, pp. 519–525, 2004.
- [19] J.C. Doyle, K. Glover, P.P. Khargonekar, B.A. Francis, *State-space solutions to standard H_2 and H_∞ control problems*, IEEE Transaction on Automatic Control, Vol. 34, August, 1989.

Résumé

Un nouvel algorithme pour résoudre le problème du contrôle à retour de sortie statique (SOF) est proposé dans le présent mémoire, une approche ILMI a été développée pour construire cette méthode. Cette méthode de stabilisation a été étendue pour résoudre aussi le problème du retour de sortie statique dans le cas de contrôle H^∞ . Les algorithmes mis au point sont également appliqués à la conception des contrôleurs PI et PID multivariables. Des exemples numériques montrent que les algorithmes proposés donnent des meilleurs résultats et aussi que la convergence est plus rapide que celle des algorithmes déjà existants.

ملخص

هذه المذكرة تعرض خوارزمي جديد لحل مشكل مراقبة العودة من المخرج الساكن، لبناء هذه الطريقة وضعت تقنية ILMI. وهذه الطريقة للاستقرار تعدت أيضا لحل المشكل من أجل المراقبة H_{∞} . إضافة إلى ذلك طبقت الخوارزميات لإيجاد المراقب PI أو PID متعدد المتغيرات. أمثلة تطبيقية تبرهن أن الخوارزميات المقترحة تعطي نتائج أحسن و أيضا تقارب أسرع من الخوارزميات المتواجدة.

Abstract

An improved iterative linear matrix inequality (ILMI) algorithm for static output feedback (SOF) stabilization problem without introducing any additional variables is proposed in this note. The proposed ILMI algorithm is also extended to solve the SOF H_{∞} controller design problem. They are applied to the multivariable PI and PID controllers. Numerical examples show that the proposed algorithms yield better results and faster convergence than the existing ones.