

Table des matières

1	Introduction	5
2	Fonctions, ensembles et optimisation convexes	15
2.1	Problème d'optimisation	15
2.2	Ensembles convexes	17
2.3	Fonctions convexes	17
2.4	Exercice : optimisation (quasi) convexe ou pas ?	21
2.5	Problème d'optimisation avec \mathcal{C} non convexe	21
3	Optimisation LMI	23
3.1	Problèmes d'optimisation sous contraintes LMI	23
3.1.1	Problème de Faisabilité	24
3.1.2	Problème de minimisation d'une fonction de coût linéaire	28
3.1.3	Minimisation de la valeur propre généralisée maximale	30
3.2	Problèmes d'optimisation LMI particuliers	32
3.3	Exercice : Optimisation sur des variables de décisions "discrètes"	33
3.4	Notions sur la résolution de problèmes sous contraintes LMI	34
3.4.1	Minimisation d'un coût linéaire sous contraintes LMI	34
3.4.2	Recherche d'un point faisable	36
3.5	Au delà des contraintes LMIs : les contraintes BMIs	36
3.6	A la recherche de la LMI cachée...	37
3.6.1	Règles de transformation si $F(\xi)$ n'est pas une fonction affine en ξ	38
3.6.2	Règle de transformation quand $x \in \mathcal{X}(\xi)$ avec $\mathcal{X}(\xi) \subset \mathbb{R}^n$ et $\mathcal{X}(\xi) \neq \mathbb{R}^n$	41
3.6.3	Exercice : mise sous forme de problèmes d'optimisation LMI	43
4	Formulation de problèmes d'Automatiques sous forme de problèmes d'optimisation LMI : exemple de la synthèse H_∞	45
4.1	Synthèse de correcteur H_∞	45
4.2	Synthèse d'un correcteur H_∞ par retour d'état	47
4.3	Lien (ébauche) avec la solution par équation de Riccati	51
4.4	Synthèse H_∞ par retour de sortie	52
4.5	Exemples	54
4.5.1	Correcteur avec effet intégral	54
4.5.2	Rejet de perturbation mesurée avec une certaine dynamique	55

5	Formulation de problèmes d'Automatiques sous forme de problèmes d'optimisation LMI : au-delà des systèmes linéaires stationnaires	57
5.1	Systèmes à paramètres variant dans le temps	57
5.2	Analyse de la stabilité	57
5.3	Exemple d'application sur un missile	58
6	Annexe A : rappels sur les matrices	63
6.1	Rappels sur les matrices	63
6.1.1	Cas des matrices carrées et réelles	66
7	Annexe B : Ensembles convexes particuliers	69
7.1	Ellipsoïdes	69
7.2	Polyèdres	70
8	Annexe C : TD sur l'analyse et la commande par LMI, Matlab et la LMI control toolbox	73
8.1	Calcul de la norme H_∞ d'une fonction de transfert	73
8.2	LMI et μ -analyse : des critères graphiques à l'optimisation	79

Chapitre 1

Introduction

L'optimisation est la branche des Mathématiques consacrée à l'étude du (ou des) minimum(s)/maximum(s) d'une fonction à plusieurs variables sur un certain domaine de définition, de l'étude de leur existence à leur détermination, en général par la mise en œuvre d'un algorithme sur un ordinateur. On peut s'interroger sur l'intérêt réel d'un tel cours dans un master en Sciences de l'Ingénieur, fût-il un master recherche. S'agit-il d'un cours joliment académique propre à flatter l'ego de l'enseignant-chercheur qui le dispense ? S'agit-il encore d'un cours présentant un domaine prospectif agitant une fraction du milieu de la recherche en Sciences de l'Ingénieur et qui, peut-être, dans un futur proche ou lointain, correspondra à une demande industrielle éclairée éventuelle ? Il ne s'agit de rien de tout cela. Il s'agit d'un cours fondamental pour la formation de l'ingénieur et du chercheur en ingénierie. En effet, la maîtrise d'outils basés sur l'optimisation devient de plus en plus incontournable dans la pratique de l'ingénieur car répondant à des enjeux industriels forts et actuels dans un nombre croissant d'applications. Un ingénieur maîtrisant ces outils possède ainsi un avantage compétitif certain pour répondre à ces enjeux. Parallèlement, au niveau de la recherche, l'optimisation apparaît comme l'une des branches des Mathématiques les plus adaptées au développement d'outils pour l'ingénieur. L'objectif de cette introduction est d'éclairer le lecteur sur l'intérêt de l'optimisation pour l'ingénieur.

Notre quotidien est actuellement envahi par une multitude d'objets (systèmes) technologiques (véhicules, systèmes électroniques, etc.) d'une grande complexité. Pour l'ingénieur, se posent alors plusieurs problèmes importants : comment réussir à appréhender le comportement d'objets de fonctionnement complexe afin de les concevoir, réaliser et/ou commander ? La description du comportement attendu du système constitue le cahier des charges.

Ces questions sont d'autant plus cruciales que le niveau de complexité des systèmes technologiques a fortement augmenté. Parallèlement, pour répondre à des contraintes de marché, le temps du cycle de conception des systèmes tend à être fortement réduit. Le phénomène est particulièrement impressionnant pour les applications "grand public". On peut citer le cas de l'automobile où le cycle de développement d'un véhicule est passé d'une dizaine d'années à moins de 5 ans alors que de nombreux systèmes électroniques complexes (ABS, ESP, injection électronique, etc..) y ont été massivement introduits, y compris dans les véhicules les plus bas de gamme afin de remplir les nouveaux objectifs pour le comportement désiré d'une véhicule (plus grand confort, sécurité accrue, etc.). Un véhicule apparaît ainsi comme un ensemble de sous systèmes complexes, interagissant les uns avec les autres. Pour répondre à cette évolution, il apparaît donc nécessaire de disposer de méthodes et d'outils de conception, de réalisation et/ou de commande des systèmes

technologiques qui soient particulièrement efficaces, plus efficaces que ceux traditionnellement mis en œuvre.

Afin de concevoir les systèmes, de les réaliser et de les commander, l'ingénieur doit acquérir une certaine compréhension de leur comportement. Pour cela, le mode le plus efficace pour exploiter toute l'information disponible sur le système considéré est de la synthétiser sous la forme d'un modèle mathématique. Les constructeurs automobiles demandent à leurs sous traitants (équipementiers) de développer et de produire des sous ensembles complets pour leurs véhicules. Depuis quelques années, certains constructeurs exigent que leurs équipementiers fournissent les modèles décrivant le comportement des sous ensembles. La "culture du modèle" tend donc à se développer, y compris dans des branches de l'industrie où elle est traditionnellement absente ou peu répandue.

La question est de savoir comment exploiter efficacement ces modèles pour l'étude du comportement des systèmes. Ces modèles se présentent en général comme des systèmes d'équations (différentielles). Une pratique courante pour étudier le comportement du système est de faire la simulation de son modèle par un ordinateur. Le développement formidable de la puissance de calcul et des possibilités de visualisation associées pour un coût modique a popularisé cette pratique parmi les ingénieurs. La simulation se retrouve dans la majorité des logiciels de Conception Assistée par Ordinateur (C.A.O.), parfois au coeur de ceux-ci. Quelles garanties peut-elle apporter ? Un élément de réponse peut être donné à travers un exemple.

Un exemple sur les limites de la simulation On désire mettre au point un métronome constitué d'un axe oscillant avec une période de 4 secondes. Il est constitué un moteur à courant

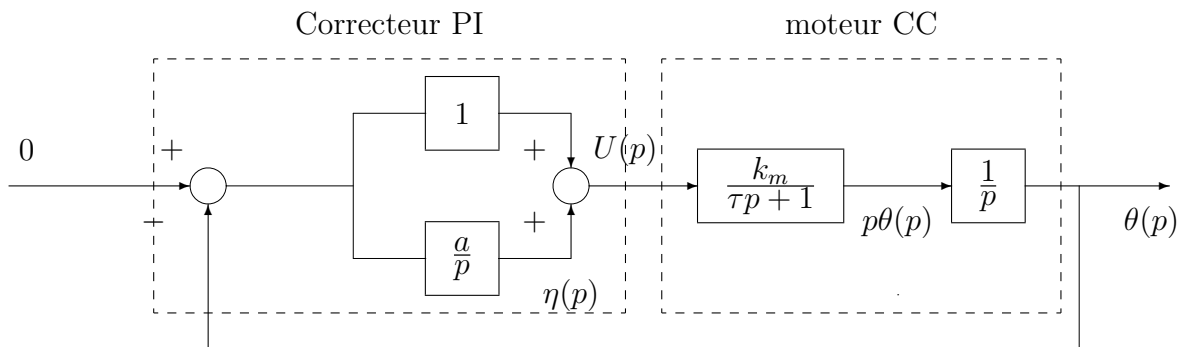


FIG. 1.1 – Commande d'un moteur à courant continu par un correcteur PI

continu commandé en position par un correcteur proportionnel intégral. Le signal $\theta(t)$ représente la position angulaire de l'axe du moteur ; le signal $\dot{\theta}(t)$ représente sa vitesse angulaire ; le signal $\eta(t)$ est un signal du correcteur. Les paramètres ont été choisis égaux à :

- pour le moteur à courant continu : $k_m = 235$ et $\tau = 67$;

– pour le correcteur PI : $a = 0.0149$ et $k^c = 1.647 \times 10^{-4}$.

de façon à ce que le signal $\theta(t)$ soit bien un signal périodique de période 4 secondes. Nous allons essayer de vérifier que le système présente bien le comportement attendu.

A partir de la figure 1.1, il est facile d’obtenir les équations différentielles (le modèle) qui décrivent l’évolution de ce système pour différentes conditions initiales et de les simuler en utilisant par exemple la boîte à outils **Simulink** de **Matlab**. Les conditions initiales correspondent aux valeurs de $\theta(t)$, $\dot{\theta}(t)$ et $\eta(t)$ pour $t = 0$.

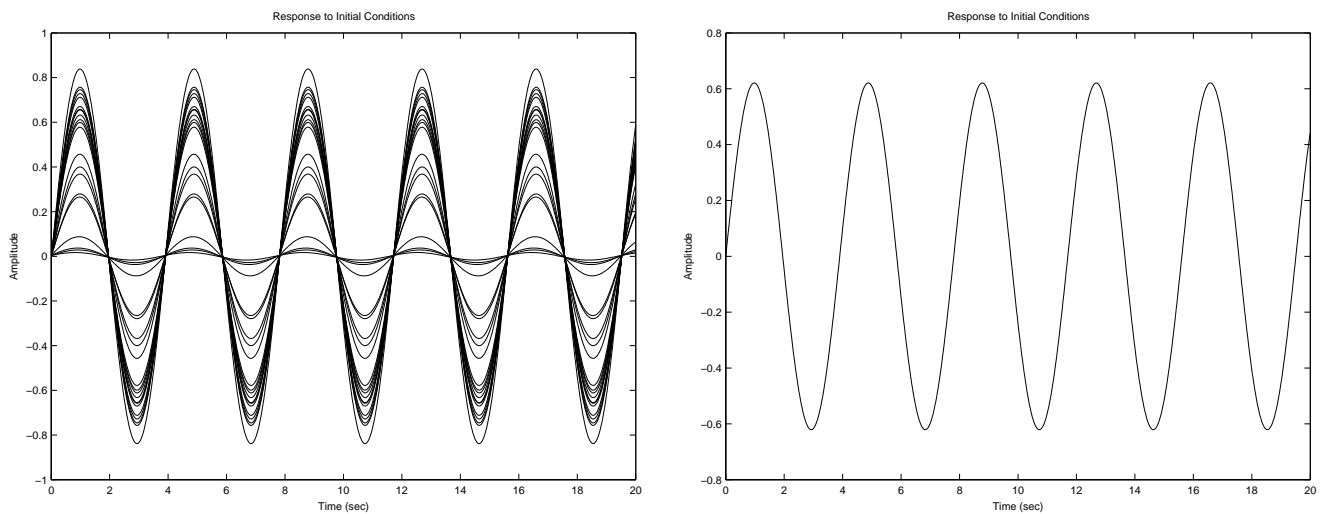


FIG. 1.2 – Simulation du moteur CC avec correcteur PI

L’utilisation de la simulation est une pratique courante pour étudier le comportement temporel (les propriétés) d’un système technologique. Pour cela, dans notre exemple, on fait des simulations correspondant à différentes conditions initiales, chaque jeu de conditions initiales correspondant à une sorte de “scénario”. Sur la figure 1.2, à gauche, sont superposées les évolutions de $\theta(t)$ en fonction de t pour 20 conditions initiales choisies aléatoirement. On constate que dans tous les cas de figure considérés, $\theta(t)$ tend bien vers un signal périodique de période 4 secondes. Cela se voit clairement sur la figure 1.2, à droite, où $\theta(t)$ est représenté pour une seule condition initiale : $\theta(0) = 0$, $\dot{\theta}(0) = 1$ radian/s et $\eta(0) = 0.0043$.

De ces différents essais, il semble donc “naturel” de conclure que, pour tous les scénari (c’est-à-dire pour toutes les conditions initiales), le système a bien le comportement désiré, à savoir $\theta(t)$ tend vers un signal périodique de période 4 secondes. Or, une étude attentive des propriétés mathématiques du modèle montre qu’il en est rien. En effet en choisissant pour condition initiale $\theta(0) = -0.0149\beta$ radians, $\dot{\theta}(0) = \beta$ radians/s et $\eta(0) = -2.5 \times 10^{-6}\beta$, avec $\beta \in \mathbb{R} \neq 0$, le signal $\theta(t)$ tends vers 0 quand t tend vers l’infini, ce qu’illustre la simulation représentée sur la figure 1.3. Est ce que si on avait effectué un nombre suffisant de tirages au sort des conditions initiales, on aurait trouvé une condition initiale qui assure ce comportement ? La réponse est non. On ne peut le mettre en évidence par simulation que si on connaît a priori l’existence de ce comportement.

Il est donc clair que l’utilisation de la simulation du modèle, même de façon intensive, ne permet pas de garantir que l’on va reproduire de façon exhaustive les comportements possibles d’un

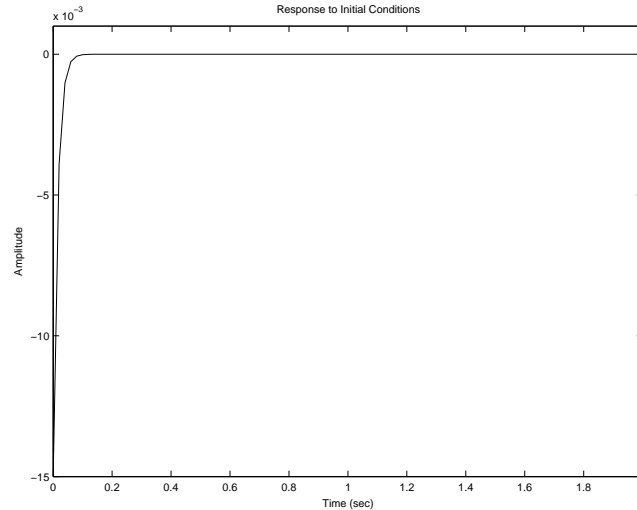


FIG. 1.3 – Simulation du moteur CC avec correcteur PI avec $\theta(0) = -0.0149$ radian, $\dot{\theta}(0) = 1$ radian/s et $\eta(0) = -2.458 \times 10^{-6}$

système. Il est donc impératif de disposer de méthodes permettant de prévoir le comportement d'un système à partir de l'étude directe des propriétés mathématiques du modèle. La simulation montre ainsi clairement ses limitations : elle ne permet que de confirmer l'existence du comportement prévu par l'étude.

Une voie alternative à la simulation est de définir les propriétés mathématiques que doit satisfaire le modèle pour refléter un certain comportement et tester si ce modèle possède effectivement ces propriétés.

Retour à l'exemple du moteur CC Au moteur à courant continu commandé par un correcteur proportionnel intégral, on peut associer une matrice A , dite matrice d'évolution :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{k_m k^c}{\tau} & -\frac{1}{\tau} & \frac{k}{\tau} \\ -a k^c & 0 & 0 \end{bmatrix}$$

dont l'étude des valeurs propres et des vecteurs propres permet de prévoir le comportement du système. Deux valeurs propres imaginaires pures correspondent à un phénomène d'oscillation qui peut être observé pour les conditions initiales définies comme des combinaisons linéaires de leurs vecteurs propres associés¹ ; une valeur propre à partie réelle strictement négative implique que pour des conditions initiales proportionnelles au vecteur propre associé, les signaux du système tendent vers 0. Dans l'exemple présent, en utilisant un logiciel de calcul scientifique général comme `Matlab`, les valeurs propres de la matrice A peuvent être calculées :

$$\lambda_1(A) = 1.61j \quad \lambda_2(A) = -1.61j \quad \lambda_3(A) = -67$$

¹Si besoin est, lire le chapitre 6 de rappels sur les matrices, page 63.

avec comme vecteur propre associé pour la troisième : $\begin{bmatrix} -0.0149 & 1 & -2.458 \times 10^{-6} \end{bmatrix}^T$. On a donc $\theta(t)$ qui tend vers 0 pour $\theta(0) = -0.0149\beta$ radians, $\dot{\theta}(0) = \beta$ radians/s et $\eta(0) = -2.458 \times 10^{-6}\beta$, avec $\beta \in \mathbb{R} \neq 0$, comme on peut l'observer sur la simulation reproduite Figure 1.3.

Par suite, on a pu étudier de façon exhaustive le comportement du système à travers la vérification d'un critère mathématique sur un modèle mathématique. On peut ainsi garantir quel sera le comportement du système pour tous les scénari possibles, ce que n'a pas permis la simulation. Un certain nombre de logiciels de CAO incluent² une partie de ces outils.

Il est donc important d'être capable de formaliser les différents types du comportement attendu d'un système technologique par un critère mathématique. La (non) vérification de ce critère mathématique permet de garantir la (non) présence du comportement attendu. Un nombre très important de ces problèmes de vérification de critères mathématiques sur des modèles peut s'écrire comme des *problèmes d'optimisation*, c'est-à-dire comme la recherche du minimum (ou du maximum) d'une fonction à plusieurs variables sur un certain domaine de définition.

Actuellement, tous les différents types de comportements intéressants ne sont pas formalisés par des critères mathématiques. Il s'agit d'un champ de recherche au coeur des Sciences de l'Ingénieur, avec un fort intérêt à la fois fondamental (théorique) et appliqué (pratique), avec des évolutions importantes à court terme et à long terme.

Ce champ de recherche est d'autant plus important qu'actuellement les exigences au niveau du comportement (cahier des charges) sont de plus en plus fortes pour des systèmes décrits par des modèles de plus en plus complexes. L'autre point remarquable est que les outils et les méthodes développés sont suffisamment fondamentaux pour ne pas correspondre à un type particulier de systèmes technologiques. Des problèmes aussi différents que la commande d'engins volants, de procédés industriels chimiques, de canaux d'irrigation ou encore le développement de suspensions actives en automobile peuvent être ainsi abordés en se basant sur un même ensemble de méthodes. Cela est d'autant plus important qu'actuellement de nombreux systèmes technologiques font appel à différents domaines de compétence. Par exemple, un véhicule grand publique était il y a une quinzaine d'année essentiellement un système mécanique. Les véhicules qui sont conçus actuellement sont des systèmes à la fois mécaniques et électroniques tant est grande la place prise par l'électronique.

La vérification du critère mathématique reflétant le comportement se fait en règle générale par calcul numérique, sur un ordinateur. Tout comme pour la simulation, le développement de cette pratique en ingénierie est lié à la forte augmentation de la puissance de calcul pour un coût de plus en plus bas. Dans l'exemple du moteur, son étude repose sur le calcul numérique des valeurs propres et des vecteurs propres d'une matrice, un problème d'analyse numérique bien étudié. Ce calcul prend dans cet exemple une fraction de seconde car il est résolu par un algorithme de faible complexité. Ce point est important car il est impératif que la vérification du critère mathématique par un algorithme se fasse en un temps suffisamment raisonnable pour l'ingénieur. Est-ce toujours le cas ?

La difficulté est de définir un temps "raisonnable". Une mesure usuelle est d'étudier l'évolution du temps de calcul pour la vérification d'un critère en fonction de la taille du problème. Par exemple, pour une matrice carrée (comme la matrice A), ce sera le nombre de lignes. Etudions deux cas de figures :

²ou devrait inclure

Complexité	Taille n du problème				
	10	20	30	40	50
n^3	0,01 s	0,08 s	0,27 s	0,64 s	1,25 s
	10 s	1,33 mn	4,50 mn	10,67 mn	20,83 mn
2^n	0,01 s	10,24 s	2,91 h	124,3 jours	348,7 ans
	10 s	2,84 h	121,4 jours	348,7 ans	$3,49 \times 10^5$ ans

TAB. 1.1 –

1. un premier cas où le temps de calcul est une fonction polynomiale d'ordre 3 de la taille n du problème : dans le tableau 1.1, on considère deux polynômes d'ordre 3, un premier dont l'évaluation pour $n = 10$ donne 0,01s et un second 10s.
2. un second cas où le temps de calcul est une fonction exponentielle de la taille n du problème (2^n) : dans le tableau 1.1, on considère deux fonctions, une première dont l'évaluation pour $n = 10$ donne 0,01s et une seconde 10s.

Dans le cas où le temps de calcul est une fonction polynomiale, pour $n = 50$ le temps de calcul reste relativement raisonnable : 1,25 secondes et 20,83 minutes. Dans le cas où le temps de calcul est une fonction exponentielle, pour $n = 50$ le temps de calcul devient prohibitif : 349 années³ et 349000 années⁴. Un problème est considéré comme facile quand on est capable de le résoudre par un algorithme en un temps fonction polynomiale de la taille du problème. Dans le cas où le temps est fonction exponentielle, on considère que le problème est difficile⁵. Il ne faut pas confondre difficulté du problème avec difficulté de l'algorithme de résolution : un problème donné peut être résolu par différents algorithmes. Il est considéré comme facile s'il existe au moins un algorithme qui le résout en un temps fonction polynomiale de la taille du problème. Il est considéré comme difficile si tous les algorithmes le résolvant (y compris ceux qu'on ne connaît pas encore) ne sont pas en temps polynomial.

Les problèmes d'optimisation ont été évoqués ci-dessus comme une classe générale de problèmes mathématiques permettant l'étude des propriétés d'un modèle. Les problèmes d'optimisation convexe apparaissent comme une sous classe de problèmes "faciles", c'est-à-dire dotée d'algorithmes de résolution en temps polynomial. L'optimisation convexe sous contraintes "LMI" apparaît actuellement comme une des plus larges classes d'optimisation convexe pour laquelle on dispose d'algorithmes de résolution efficace proposés dans les logiciels de calcul scientifique généraux comme *Matlab* ou *Scilab* et qui a eu d'importantes applications en Sciences de l'Ingénieur. Après des généralités sur l'optimisation convexe, ce document de cours donnera une introduction à cette classe de problèmes d'optimisation.

De nombreux problèmes d'ingénierie se formulent comme des problèmes d'optimisation pour lesquels on ne dispose pas d'algorithme de résolution efficace. Cependant, un certain nombre

³Soit la durée de quatre vies humaines environ.

⁴L'homme préhistorique est apparu il y a environ 1 000 000 ans. Malheureusement l'ordinateur a moins de 100 ans.

⁵En théorie de la complexité, on considère deux classes de problèmes :

1. \mathcal{P} classe des problèmes calculables en temps polynomial
2. \mathcal{NP} classe des problèmes vérifiables en temps polynomial

Une conjecture largement admise par des gens très compétents est que $\mathcal{P} \neq \mathcal{NP}$.

d'entre eux peuvent être réécrits de façon équivalente comme des problèmes d'optimisation d'une autre classe. Si ces problèmes équivalents sont des problèmes d'optimisation convexe sous contrainte "LMI" alors on est ramené à des problèmes qu'il est possible de résoudre efficacement en utilisant les logiciels actuellement disponibles. Transformer un problème d'optimisation en un problème d'optimisation équivalent est un "art" dont nous verrons quelques règles dans le cas où l'on veut se ramener à un problème d'optimisation LMI.

L'intérêt de l'optimisation ne se limite pas à la possibilité d'étudier un nombre important de propriétés de modèles. L'optimisation permet d'envisager l'étude des propriétés non pas d'un modèle mais d'une famille de modèles. Nous allons voir que ce mode de représentation est beaucoup plus adapté à ce qu'attend un ingénieur de l'étude des systèmes par la modélisation. Si l'étude d'un nombre significatif de propriétés d'un modèle peut se faire par des critères d'autre nature basés par exemple sur le calcul des valeurs propres, lorsque l'étude des familles de modèles est envisagée, l'optimisation devient l'outil incontournable.

Que représente une famille de modèles ? Quel est son intérêt pour l'ingénieur ? Dans la discussion précédente, on a supposé que le modèle représentait exactement le système. En pratique, il y a nécessairement des différences (erreurs de modélisation) qui apparaissent entre le système réel et son modèle :

- au niveau de la structure : on a supposé qu'un moteur CC se modélise par une équation différentielle d'ordre 2 : une modélisation plus fine peut mener à un modèle d'ordre plus important ;
- au niveau des paramètres : les paramètres du moteur CC ne sont pas mesurables avec une précision infinie ; d'autre part, ils peuvent évoluer au cours du temps.

Quel peut être l'impact de ces hypothèses sur la qualité de la prédiction du comportement ?

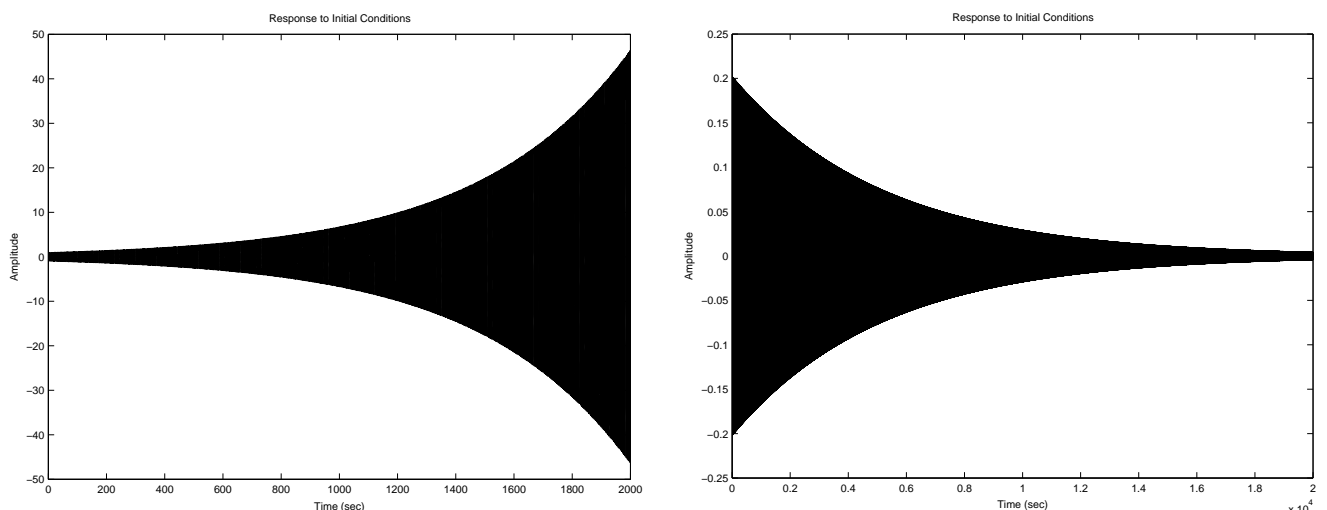


FIG. 1.4 – Simulation du moteur CC avec correcteur PI

De l'impact des erreurs de modélisation à travers un exemple Reprenons l'exemple du moteur à courant continu commandé par un correcteur PI. Supposons qu'une erreur de 1% s'est

glissée dans la mesure de la constante de temps τ du moteur : en réalité, elle est plus importante et vaut $\tau = 0.0151$ s au lieu de $\tau = 0.0149$ s. Pour une condition initiale donnée, on obtient l'évolution de $\theta(t)$ représentée figure 1.4, à gauche. On remarque que $\theta(t)$ tend à diverger : il est possible de démontrer que c'est effectivement le cas. Supposons maintenant que τ est en réalité moins importante (toujours une erreur de 1%) : $\tau = 0.0148$ s au lieu de $\tau = 0.0149$ s. Pour une condition initiale donnée, on obtient l'évolution de $\theta(t)$ représentée figure 1.4, à droite. On remarque que $\theta(t)$ semble tendre vers 0 : il est possible de démontrer que c'est effectivement le cas.

Le comportement du système est donc très *sensible* à de très faibles variations de ces paramètres. Comme il est impossible de réaliser technologiquement un système dont les paramètres sont exactement fixés et stables dans le temps, cela veut dire qu'en pratique, suivant la réalisation effectuée, $\theta(t)$ oscillera, divergera ou tendra vers 0... Concevoir un système technologique dont le comportement n'est pas à priori "prévisible" est évidemment intolérable pour un ingénieur.

Dans le cas de notre système oscillateur, l'utilisation pertinente d'un élément non-linéaire "bien choisi" peut permettre la mise au point d'un système oscillant pour toute condition initiale et malgré des variations de ses paramètres.

L'étude du comportement d'un système à travers son modèle doit donc prendre en compte explicitement les erreurs de modélisation. C'est pour cela que plutôt que représenter un système par un seul modèle : par exemple, le moteur CC est une fonction de transfert définie par :

$$\frac{k_m}{p(\tau p + 1)}, \quad k_m = 235, \quad \tau = 0.0149,$$

il est préférable de le modéliser par une famille de modèles : le moteur CC est une fonction de transfert appartenant à l'ensemble des fonctions de transfert défini par :

$$\left\{ \frac{k_m}{p(\tau p + 1)}, \quad k_m \in [212; 258], \quad \tau \in [0.0134; 0.0164] \right\}.$$

Si on est capable de garantir un même comportement pour toutes les fonctions de transfert de cet ensemble commandées par un correcteur PI, comme le moteur CC est un élément de cet ensemble (on ne sait pas lequel), on garantira ce comportement aussi pour le système réel.

La vérification d'un critère mathématique pour une famille de modèles a l'avantage d'être un problème beaucoup plus pertinent pour l'ingénieur. Par contre, le problème mathématique associé est plus complexe que la vérification d'un critère mathématique pour un seul modèle. Il est l'objet d'intenses recherches dans les Sciences de l'Ingénieur depuis une dizaine d'années, recherches au sein desquelles l'optimisation occupe une place centrale. Les idées sous jacentes seront évoquées à travers différents exemples dans ce cours.

Enfin, la discussion de ce chapitre a porté sur l'analyse du comportement d'un système technologique par optimisation. Qu'en est-il de sa conception, c'est-à-dire, par exemple, du choix de des différents paramètres du système ? De la même façon, on peut montrer que les problèmes de conceptions peuvent être abordés aussi de façon prometteuse par la même démarche. Cela sera illustré dans le cours par différents exemples.

Plan du document La première partie présente les notions de base en optimisation convexe. La seconde partie est consacrée à l'optimisation LMI : définition, exemples de problèmes, notions sur la résolution et comment se ramener à des problèmes d'optimisation LMI (si cela est possible). La dernière partie est consacrée à des exemples de formulation de problèmes comme des problèmes d'optimisation LMI. L'annexe est consacrée à des rappels sur les matrices, les ensembles convexes ainsi que des exemples de programmes `Matlab` d'optimisation LMI.

Chapitre 2

Fonctions, ensembles et optimisation convexes

Si besoin est, lire le chapitre 6 de rappels sur les matrices, page 63.

2.1 Problème d'optimisation

Definition 2.1.1 (*Problème d'optimisation de dimension finie*) Soit

$$\begin{aligned}\mathbb{R}^m &\rightarrow \mathbb{R} \\ \xi &\mapsto f(\xi)\end{aligned}$$

Alors un problème d'optimisation s'écrit :

$$\min_{\xi \in \mathcal{C}} f(\xi)$$

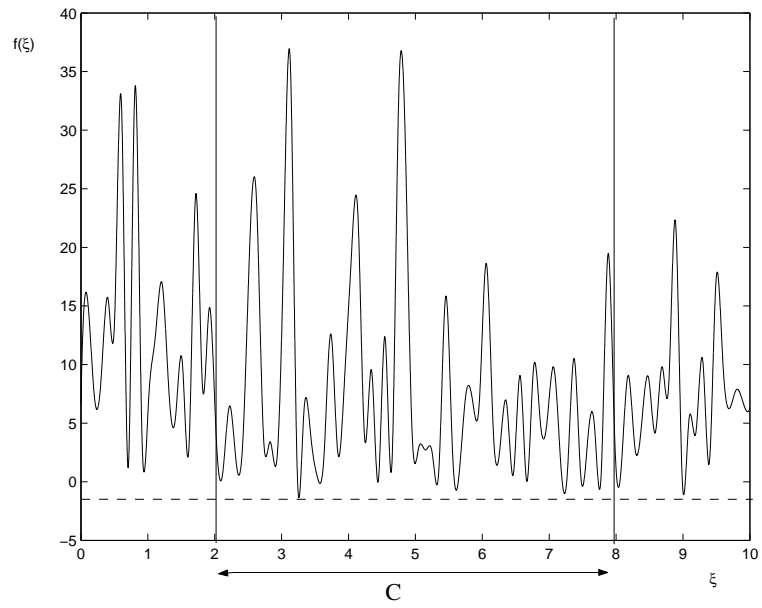
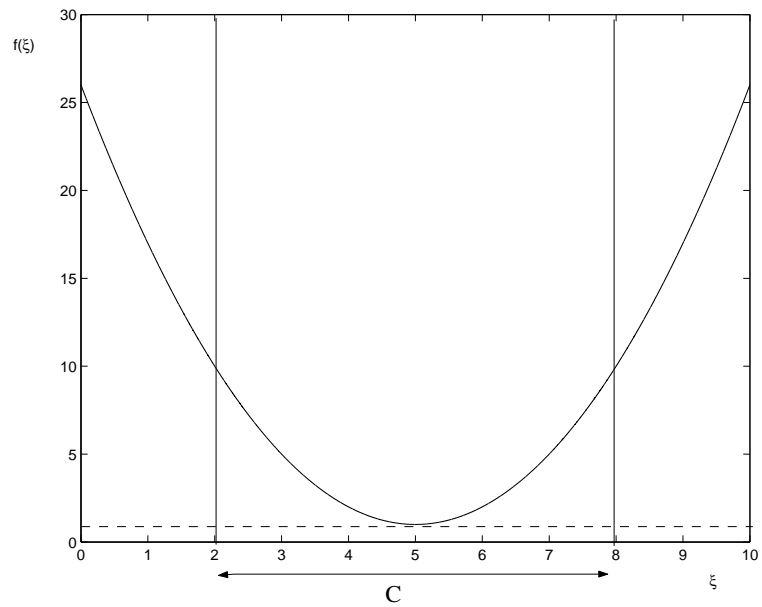
où

- $\mathcal{C} \subset \mathbb{R}^m$ est l'ensemble des contraintes
- ξ est le vecteur de variables de décision ou variables d'optimisation
- la valeur de ξ pour laquelle le minimum est atteint est appelée ξ^* . Notation :

$$\xi^* = \operatorname{argmin}_{\xi \in \mathcal{C}} f(\xi)$$

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

Dans le cas général, la résolution par un algorithme d'un problème d'optimisation est un problème compliqué (voir figure 2.1). Pourquoi? A partir d'un point initial ξ_0 , les algorithmes efficaces recherchent un minimum local. Si la fonction f admet plusieurs minima, le résultat va dépendre du point initial ξ_0 . Dans certains cas sympathiques (minimum local = minimum global), le problème est de faible complexité (voir figure 2.2). Indépendamment du point initial, le minimum global sera alors atteint.

FIG. 2.1 – Exemple d’une fonction sur \mathbb{R} à plusieurs minimaFIG. 2.2 – Exemple d’une fonction sur \mathbb{R} à un seul minimum

Cas sympathiques Problèmes d'optimisation convexe, c'est-à-dire :

1. f est une fonction convexe
2. \mathcal{C} est convexe

Dans ce cas-là, quelque soit le point initial ξ_0 , les algorithmes convergent vers le minimum global.

2.2 Ensembles convexes

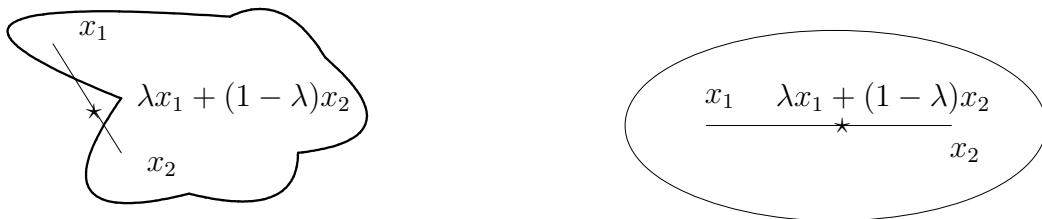


FIG. 2.3 – Ensemble non convexe (gauche), convexe (droite)

Definition 2.2.1

Ensemble convexe \mathcal{C} :

$$\forall \lambda \in [0, 1], \quad x_1 \in \mathcal{C} \text{ et } x_2 \in \mathcal{C} \implies \lambda x_1 + (1 - \lambda)x_2 \in \mathcal{C}$$

Cône convexe \mathcal{C} : $\forall \lambda_1 \geq 0, \forall \lambda_2 \geq 0, x_1 \in \mathcal{C} \text{ et } x_2 \in \mathcal{C} \implies \lambda_1 x_1 + \lambda_2 x_2 \in \mathcal{C}$

Ellipsoïde : $\varepsilon = \{x \in \mathbb{R}^n \mid (x - x_c)^T P^{-1}(x - x_c) \leq 1\}$ avec P matrice définie positive

Coquille convexe d'un ensemble S : $\{\lambda_1 x_1 + \dots + \lambda_k x_k, x_i \in S, \lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1\}$

Propriétés L'intersection de deux ensembles convexes est convexe.

Deux classes importantes d'ensembles convexes (les ellipsoïdes et les polytopes) sont étudiés en Annexe, page 69.

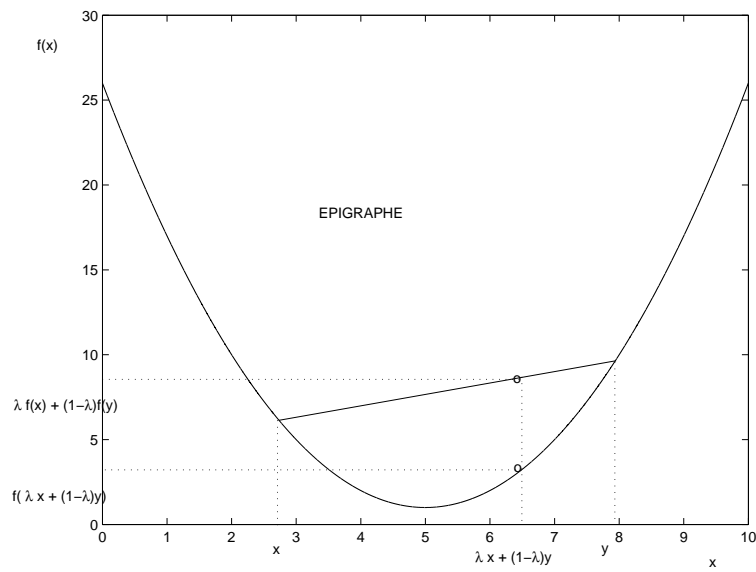
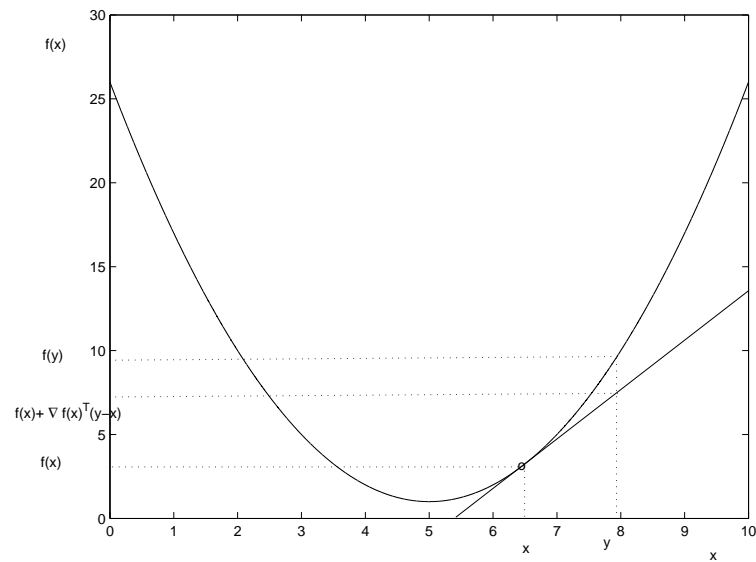
2.3 Fonctions convexes

Definition 2.3.1 (*Fonction convexe*) Soit une fonction f

$$\begin{aligned} \mathcal{D} \subset \mathbb{R}^m &\rightarrow \mathbb{R} \\ x &\mapsto f(x) \end{aligned}$$

La fonction f est convexe si :

1. le support (ensemble de définition) \mathcal{D} de f est convexe

FIG. 2.4 – Exemple d’une fonction sur \mathbb{R} à un seul minimumFIG. 2.5 – Exemple d’une fonction sur \mathbb{R} à un seul minimum

2. $\forall x_1 \in \mathcal{D}, \forall x_2 \in \mathcal{D}, \forall \lambda \in [0, 1], f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$
convexité stricte : \leq remplacé par $<$

Si f est différentiable alors la seconde condition est équivalente à

$$\forall x_1 \in \mathcal{D}, \forall x_2 \in \mathcal{D}, f(x_2) \geq f(x_1) + \nabla f(x_1)^T(x_2 - x_1)$$

où $\nabla f(x)$ est le gradient¹ de la fonction f défini par :

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

Pour une fonction convexe, d'une information de nature locale (la valeur du gradient de f en x_1), on peut tirer une information de nature globale (toute la courbe définie par la fonction f est dans un demi-plan délimité par l'hyperplan tangent (la tangente) en x_1 d'équation $y = f(x_1) + \nabla f(x_1)^T(x - x_1)$ dans le plan (x, y)).

Si f est différentiable deux fois alors la seconde condition est équivalente à

$$\forall x \in \mathcal{D}, \nabla^2 f(x) \geq 0$$

(semi définie positive) avec

$$\nabla^2 f(x) = \underbrace{\begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & \dots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}}_{\text{matrice hessienne}}$$

Remarque $\forall x \in \mathcal{D}, \nabla^2 f(x) > 0$ est une condition suffisante mais pas nécessaire de stricte convexité.

Exemple de fonction convexe $f(x) = x^T P x + 2q^T x + r$ avec $P = P^T \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, r \in \mathbb{R}$.
Quelle(s) condition(s) doit (vent) vérifier les paramètres de f (P, q et r) pour que f soit convexe ?

Propriété Si $f_1(x)$ et $f_2(x)$ sont deux fonctions convexes (même domaine de définition) alors $f(x) = \max(f_1(x), f_2(x))$ est convexe.

Propriété Si $f(x, y)$ est convexe en (x, y) et C un domaine convexe alors $g(x) = \inf_{y \in C} f(x, y)$ est une fonction convexe.

Definition 2.3.2 (fonction quasi-convexe) Soit une fonction f

$$\begin{array}{lcl} \mathcal{D} \subset \mathbb{R}^m & \rightarrow & \mathbb{R} \\ x & \mapsto & f(x) \end{array}$$

La fonction f est quasi convexe si :

¹Dans le cas d'une fonction de \mathbb{R} dans \mathbb{R} , le gradient coïncide avec la dérivée.

1. le support \mathcal{D} de f est convexe

2. $\forall x_1 \in \mathcal{D}, \forall x_2 \in \mathcal{D}, \forall \lambda \in [0, 1], f(\lambda x_1 + (1 - \lambda)x_2) \leq \max(f(x_1), f(x_2))$

La seconde condition peut se réécrire : les ensembles $C(\alpha) = \{x \in \mathcal{D} \mid f(x) \leq \alpha\}$ sont convexes.

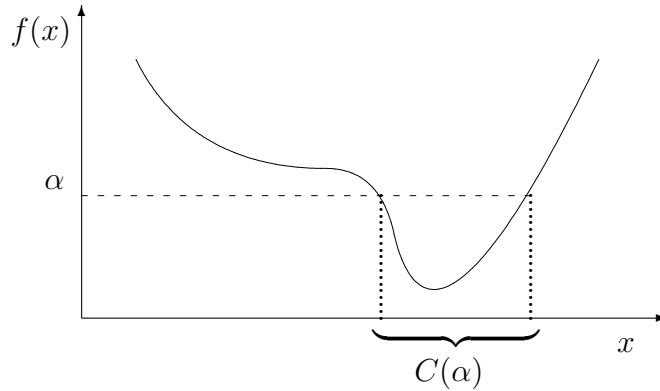


FIG. 2.6 – Fonction quasiconvexe

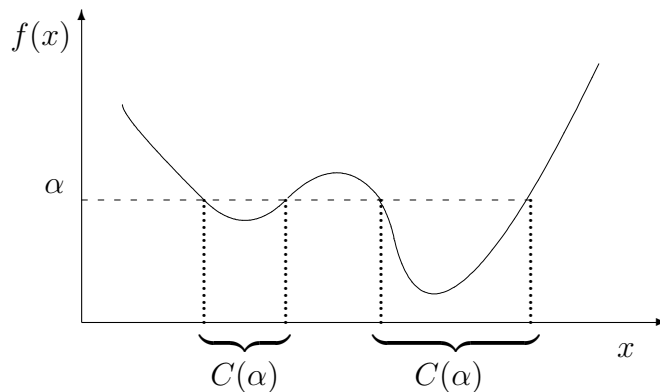


FIG. 2.7 – Fonction non quasiconvexe

Si f est différentiable alors la seconde condition est équivalente à

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}, f(y) \geq f(x) \Rightarrow \nabla f(x)^T (y - x) \leq 0$$

Si f est différentiable deux fois alors la seconde condition est équivalente à

$$\forall x \in \mathcal{D}, y^T \nabla f(x) = 0 \Rightarrow y^T \nabla^2 f(x) y \geq 0$$

Remarque Pour avoir un problème d'optimisation convexe, il est d'une part nécessaire que l'ensemble des contraintes \mathcal{C} soit convexe. D'autre part, la fonction de coût f doit être définie sur un domaine contenant l'ensemble des contraintes et doit être convexe au moins sur l'ensemble des contraintes \mathcal{C} .

2.4 Exercice : optimisation (quasi) convexe ou pas ?

Soit le problème d'optimisation suivant :

$$\min_{\xi \in C_1 \cup C_2} f(\xi)$$

où f est une fonction de \mathbb{R} dans \mathbb{R} et où C_1 et C_2 sont deux intervalles de \mathbb{R} . Pour les quatre cas représentés sur la figure 2.8 et correspondant à différentes fonctions f et différents ensembles C_1 et C_2 , indiquez si le problème d'optimisation correspondant est convexe (quasi convexe) ou pas. Justifier en quelques mots.

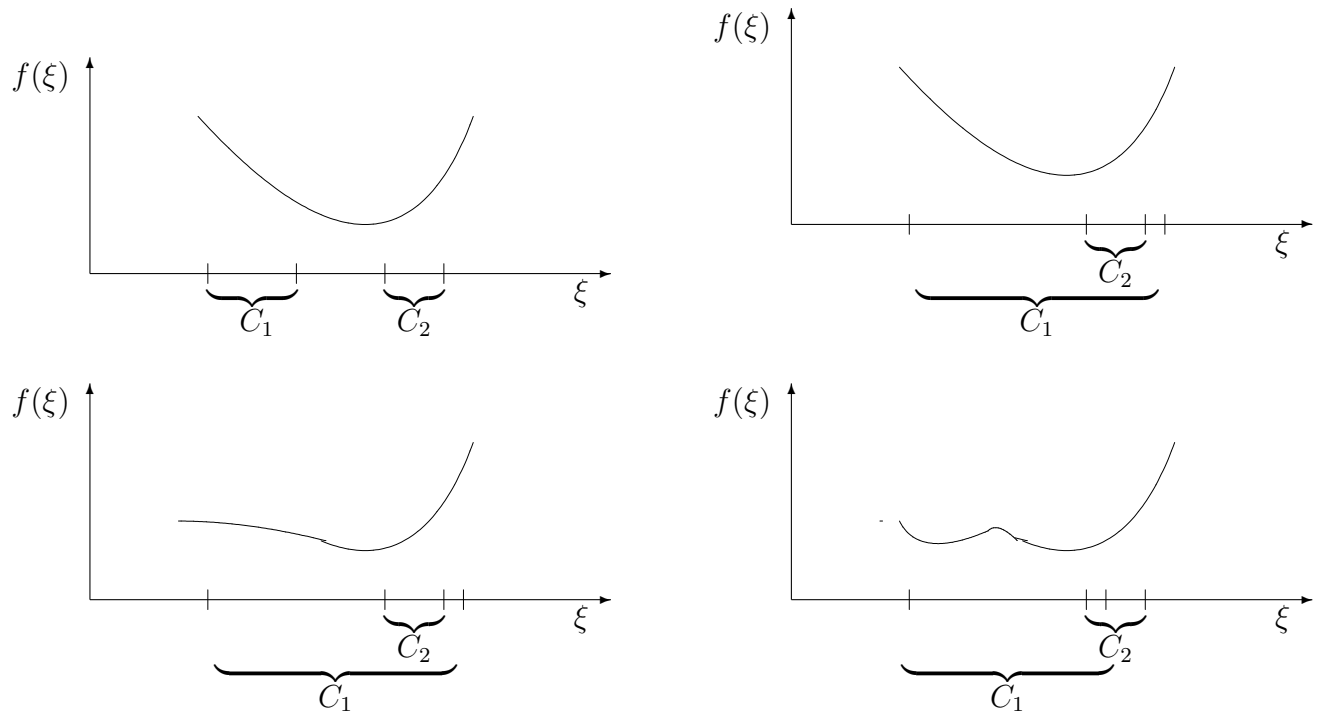


FIG. 2.8 – cas de figure considérés

2.5 Problème d'optimisation avec \mathcal{C} non convexe

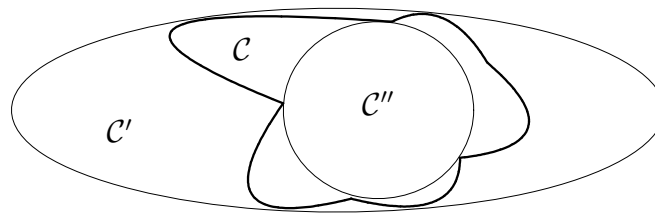
Soit

$$\min_{\xi \in \mathcal{C}} f(\xi)$$

avec f convexe mais \mathcal{C} non convexe. En général la résolution de ce problème est complexe. Peut-on utiliser l'optimisation convexe pour obtenir une borne inférieure et une borne supérieure sur la valeur du minimum $f(\xi^*)$?

Borne inférieure par relaxation de l'ensemble des contraintes \mathcal{C} Supposons que l'on dispose d'un ensemble convexe \mathcal{C}' tel que $\mathcal{C} \subset \mathcal{C}'$ (par exemple \mathcal{C}' coquille convexe de \mathcal{C}). Alors

$$\min_{\xi \in \mathcal{C}'} f(\xi) \leq f(\xi^*).$$

FIG. 2.9 – Ensembles \mathcal{C} , \mathcal{C}' et \mathcal{C}''

Borne supérieure par restriction de l'ensemble des contraintes \mathcal{C} Supposons que l'on dispose d'un ensemble convexe \mathcal{C}'' tel que $\mathcal{C}'' \subset \mathcal{C}$. Alors

$$f(\xi^*) \leq \min_{\xi \in \mathcal{C}''} f(\xi).$$

Chapitre 3

Optimisation LMI

3.1 Problèmes d'optimisation sous contraintes LMI

$$\min_{\xi \in \mathcal{C}} f(\xi)$$

où

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

avec

$$F(\xi) \triangleq F_0 + \sum_{i=1}^m \xi_i F_i,$$

avec ξ le vecteur des variables de décision, vecteur de \mathbb{R}^m et où les F_i sont m matrices symétriques données de $\mathbb{R}^{n \times n}$, $i = 0, \dots, m$. Cette contrainte est appelée *contrainte Inégalité Matricielle Affine* ou contrainte *LMI*. Notation $F(\xi) > 0 (\geq 0)$. Le symbole $> 0 (\geq 0)$ signifie définie positive (semi définie positive)¹.

Propriété 1 \mathcal{C} est un ensemble convexe.

Propriété 2 Soient :

$$\mathcal{C}_1 = \{\xi \in \mathbb{R}^m \mid F^1(\xi) > 0\} \quad \text{et} \quad \mathcal{C}_2 = \{\xi \in \mathbb{R}^m \mid F^2(\xi) > 0\}$$

alors l'intersection est définie par

$$\mathcal{C}_1 \cap \mathcal{C}_2 = \left\{ \xi \in \mathbb{R}^m \mid \begin{bmatrix} F^1(\xi) & 0 \\ 0 & F^2(\xi) \end{bmatrix} > 0 \right\}$$

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. Pour plus de détails, voir les rappels sur la matrices, section 6.1.

¹Rappelons que $\forall x \in \mathbb{R}^n, x^T F(\xi)x > 0 \Leftrightarrow$ les valeurs propres de $F(\xi)$ sont strictement positives. Pour plus de détails, voir les rappels sur les matrices, section 6.1, page 63.

Exemple I Soit $F(\xi)$ la matrice symétrique de fonctions affines en le scalaire réel ξ :

$$\begin{aligned} F(\xi) &= \begin{bmatrix} 2 - \xi & 3\xi - 2 \\ 3\xi - 2 & 2 - \xi \end{bmatrix} \\ &= \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} + \xi \begin{bmatrix} -1 & 3 \\ 3 & -1 \end{bmatrix} \end{aligned}$$

Le polynôme caractéristique de la matrice $F(\xi)$ est donné par $\lambda^2 - 2(2 - \xi)\lambda + 8\xi(1 - \xi)$. Le produit étant donné par $8\xi(1 - \xi)$ et la somme par $2(2 - \xi)$, les racines sont $\lambda_1(F(\xi)) = 4(1 - \xi)$ et $\lambda_2(F(\xi)) = 2\xi$. Par suite, la contrainte LMI $\{\xi \mid F(\xi) > 0\}$ correspond à définir le domaine : $\{\xi \mid \xi \in]0; 1[\}$.

3.1.1 Problème de Faisabilité

Tester si $\mathcal{C} \neq \emptyset$ et si oui déterminer un $\xi \in \mathcal{C}$:

$$\begin{aligned} &\text{trouver } \xi \in \mathbb{R}^m \\ &\text{tel que } F(\xi) > 0 \end{aligned}$$

Exemple I (suite) Le problème est faisable car l'intervalle $]0; 1[$ n'est pas vide.

Exemple II : Etude de la stabilité de systèmes linéaires stationnaires Soit le système :

$$\dot{x}(t) = Ax(t)$$

exemple du pendule simple : $\ddot{\theta}(t) = -\eta\dot{\theta}(t) - \frac{3}{4}\frac{g}{l}\sin(\theta(t))$. Pour θ faible, $\sin(\theta) \simeq \theta$, d'où : $\ddot{\theta}(t) = -\eta\dot{\theta}(t) - \frac{3}{4}\frac{g}{l}\theta(t)$, soit :

$$\underbrace{\begin{bmatrix} \dot{\theta}(t) \\ \dot{\theta}(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{3}{4}\frac{g}{l} & -\eta \end{bmatrix}}_A \underbrace{\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}}_{x(t)}$$

Le vecteur $x(t)$ est appelé vecteur d'état. Ici l'état est donné par la position angulaire $\theta(t)$ et par la vitesse $\dot{\theta}(t)$. Pour une valeur donnée x_0 de l'état $x(t)$ à l'instant $t = 0$, on dit que le système est stable² si pour toute condition initial x_0 , on a :

$$\lim_{t \rightarrow +\infty} x(t) = 0 \quad \text{soit encore} \quad \lim_{t \rightarrow +\infty} \|x(t)\| = 0.$$

Vérifier la stabilité du système pendule revient à assurer que lorsque le pendule est abandonné au temps $t = 0$ avec position angulaire θ_0 et une vitesse angulaire $\dot{\theta}_0$, il tend à s'immobiliser à la position angulaire $\theta = 0$ et ceci, pour toute valeur de θ_0 et pour toute valeur de $\dot{\theta}_0$.

²En toute rigueur, cette définition n'est pas correcte, surtout si le système est décrit par des équations non linéaires. Elle a été introduite pour simplifier l'exposé de l'exemple dont l'objectif n'est pas de présenter les différentes notions de stabilité des systèmes dynamiques.

D'après la théorie de Lyapunov, on a stabilité pour toute condition initiale s'il existe une fonction $V(x)$ telle que pour $x \neq 0$, $V(x) > 0$ et $\dot{V}(x) < 0$. On prend $V(x) = x^T P x$ où P est une matrice symétrique à déterminer.

$$\forall x \neq 0, \quad \begin{cases} V(x) > 0 \\ \dot{V}(x) < 0 \end{cases} \Leftrightarrow \begin{cases} P > 0 \\ A^T P + P A < 0 \end{cases} \Leftrightarrow \begin{bmatrix} P & 0 \\ 0 & -(A^T P + P A) \end{bmatrix} > 0 \quad (3.1)$$

La dernière équivalence découle du fait que les valeurs propres d'une matrice "diagonale par blocs" sont constituées par les valeurs propres des matrices sur la diagonale. Pour plus de détails, voir les rappels sur la matrices, section 6.1. Avec

$$P = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i P_i$$

avec $P_i, i \in \{1, \dots, \frac{n(n+1)}{2}\}$ une base de l'espace des matrices symétriques de dimension $n \times n$. Par exemple, pour $n = 2$, on peut prendre comme base :

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}.$$

En posant $F_0 = 0$,

$$F(\xi) = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i F_i \quad \text{avec} \quad F_i = \begin{bmatrix} P_i & 0 \\ 0 & -(A^T P_i + P_i A) \end{bmatrix}$$

On a

$$\text{trouver } V \text{ tel que } \forall x \neq 0, \quad \begin{cases} V(x) > 0 \\ \dot{V}(x) < 0 \end{cases} \Leftrightarrow \text{trouver } \xi \text{ tel que } F(\xi) > 0 \quad (3.2)$$

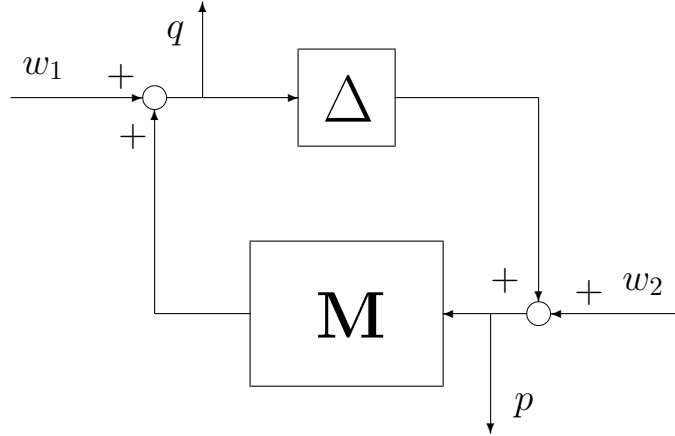
Remarque importante sur les notations Dans l'exemple précédent, les contraintes LMIs peuvent être écrites sous deux formats particuliers : d'une part, en utilisant les variables de décision sous la forme matricielle (par exemple P , dans l'inégalité (3.1)), d'autre part, en mettant les variables de décision sous forme vectorielle ξ , dans l'inégalité (3.2). Dans la suite, les variables de décision sous forme matricielle apparaissant dans les inégalités seront soulignées.

Exemple III : analyse de la stabilité d'un système incertain On désire vérifier qu'un système incertain (représenté figure 8.4 avec M stable) est stable pour toute matrice de fonctions de transfert $\Delta(p)$, de norme H_∞ inférieure à 1 et telle que $\forall \omega, \Delta(j\omega) \in \underline{\Delta}$ avec

$$\underline{\Delta} = \left\{ \Delta \in \mathbb{C}^{k \times k} / \Delta = \begin{bmatrix} \Delta_1 & 0 & \dots \\ 0 & \ddots & \ddots \\ \vdots & \ddots & \Delta_q \end{bmatrix} \text{ avec } \begin{cases} \Delta_i \in \mathbb{C}^{q_i \times q_i}, \quad i \in \{1, \dots, q\} \\ k = \sum_{i=1}^q q_i \end{cases} \right\}. \quad (3.3)$$

D'après le théorème du petit gain structuré, une condition nécessaire et suffisante est donnée par :

$$\forall \omega, \quad \mu_{\underline{\Delta}}(M(j\omega)) < 1.$$

FIG. 3.1 – Connexion de M avec Δ

(Dans ce qui suit, on considère une pulsation ω donnée. Pour alléger les notations la dépendance en ω de $M(j\omega)$ n'est plus indiquée.) Le problème est qu'en général le calcul de $\mu_{\Delta}(M)$ est NP difficile. Par contre, on dispose d'une borne supérieure $\mu_{\Delta}^{sup}(M)$ telle que

$$\mu_{\Delta}(M) \leq \mu_{\Delta}^{sup}(M)$$

où la borne supérieure de μ est obtenue par la résolution du problème d'optimisation suivant :

$$\begin{aligned} \mu_{\Delta}^{sup}(M) = \min_{\substack{D \in \mathcal{D} \\ \beta \in \mathbb{R}^+ \\ M^*DM - \beta^2 D < 0}} \beta \end{aligned}$$

avec

$$\mathcal{D} = \left\{ D \in \mathbb{C}^{k \times k} / D = \begin{bmatrix} d_1 I_{q_1} & 0 & \dots \\ 0 & \ddots & \ddots \\ \vdots & \ddots & d_q I_{q_q} \end{bmatrix} \text{ avec} \right.$$

$$d_i \in \mathbb{R}, \quad d_i > 0, \quad i \in \{1, \dots, q\} \left. \right\}.$$

Pour notre problème, il s'agit de démontrer que $\mu_{\Delta}(M) < 1$. Une condition suffisante est alors :

$$\mu_{\Delta}^{sup}(M)(M) < 1.$$

Pour vérifier cela, il s'agit de trouver $D \in \mathcal{D}$ telle que

$$M^*DM - D < 0$$

Mettons ce problème sous forme d'un problème de faisabilité LMI. La première difficulté est que la matrice M est une matrice complexe. Il est possible de transformer des inégalités matricielles complexes en inégalités matricielles réelles.

Propriété 1 Soit une matrice $A \in \mathbb{C}^{N \times N}$ donnée telle que³ $A = A^*$ et $A = A_R + jA_I$ avec $A_R \in \mathbb{R}^{N \times N}$ et $A_I \in \mathbb{R}^{N \times N}$. Alors on a

$$A > 0 \Leftrightarrow \begin{bmatrix} A_R & A_I \\ -A_I & A_R \end{bmatrix} > 0$$

³où A^* note la matrice A transposée et conjuguée.

Propriété 2 Soit une matrice $A \in \mathbb{C}^{N \times N}$ donnée telle que $A = A_R + jA_I$ avec $A_R \in \mathbb{R}^{N \times N}$ et $A_I \in \mathbb{R}^{N \times N}$ et soit une matrice B définie de la même façon. Alors

$$\begin{bmatrix} (AB)_R & (AB)_I \\ -(AB)_I & (AB)_R \end{bmatrix} = \begin{bmatrix} A_R & A_I \\ -A_I & A_R \end{bmatrix} \begin{bmatrix} B_R & B_I \\ -B_I & B_R \end{bmatrix}$$

De même

$$\begin{bmatrix} (A+B)_R & (A+B)_I \\ -(A+B)_I & (A+B)_R \end{bmatrix} = \begin{bmatrix} A_R & A_I \\ -A_I & A_R \end{bmatrix} + \begin{bmatrix} B_R & B_I \\ -B_I & B_R \end{bmatrix}$$

En appliquant ces deux propriétés, $M^*DM - D < 0$ est équivalent à

$$\begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}^T \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix} - \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} < 0$$

soit en remplaçant D_R et D_I par leurs valeurs

$$\begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}^T \begin{bmatrix} d_1 I_{q_1} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & d_q I_{q_q} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_1 I_{q_1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & d_q I_{q_q} \end{bmatrix} \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix} - \begin{bmatrix} d_1 I_{q_1} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & d_q I_{q_q} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_1 I_{q_1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & d_q I_{q_q} \end{bmatrix} < 0$$

Les variables d'optimisation sont d_1, \dots, d_q . Par suite, en posant $\xi = [d_1 \ \cdots \ d_q]^T$, on a $M^*DM - D < 0 \Leftrightarrow F(\xi) > 0$ avec

$$F_i = \begin{bmatrix} \Omega_i & & & & & 0 \\ & \Omega_i & 0 & & & \\ 0 & \begin{bmatrix} \Omega_i & 0 \\ 0 & \Omega_i \end{bmatrix} & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix} - \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}^T \begin{bmatrix} \Omega_i & 0 \\ 0 & \Omega_i \end{bmatrix} \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}$$

avec Ω_i une matrice dont tous les éléments sont nuls sauf le $i^{\text{ième}}$ de coordonnées (i, i) qui est mis à 1. On est donc ramener à un problème de faisabilité.

3.1.2 Problème de minimisation d'une fonction de coût linéaire

$$\begin{aligned} & \text{minimiser} && c^T \xi \\ & \text{pour } \xi \in \mathbb{R}^m \\ & \text{contraint par} && F(\xi) > 0 \end{aligned}$$

où c^T est un vecteur ligne donné.

Exemple I (suite) On considère maintenant le problème suivant :

$$\begin{aligned} & \min && \lambda \\ & \lambda \in \mathbb{R}, \xi \in \mathbb{R} && \lambda I - F(\xi) \geq 0 \end{aligned}$$

où $F(\xi)$ est défini dans l'exemple I. En utilisant le fait que pour une matrice A symétrique⁴ $\lambda_{\max}(A) = \min_{\lambda I - A \geq 0} \lambda$, ce problème est équivalent à :

$$\begin{aligned} & \min && \lambda_{\max}(F(\xi)). \\ & \xi \in \mathbb{R}^2 \end{aligned}$$

On peut noter que, dans ce cas, un problème d'optimisation sous contraintes peut être réécrit de façon équivalente en un problème d'optimisation sans contrainte. Ils sont équivalents au sens qu'ils admettent la même solution : même valeur de ξ^* et même valeur de la fonction de coût pour ξ^* .

Dans ce qui suit, on voit que si la fonction de ξ $\lambda_{\max}(F(\xi))$ est convexe, elle n'est cependant pas dérivable. Pour cela, avec $\lambda_{\max}(F(\xi)) = \max\{\lambda_1(F(\xi)), \lambda_2(F(\xi))\}$, on a la figure 3.2. Par suite, cette fonction de ξ n'est pas différentiable pour $\xi = \frac{2}{3}$.

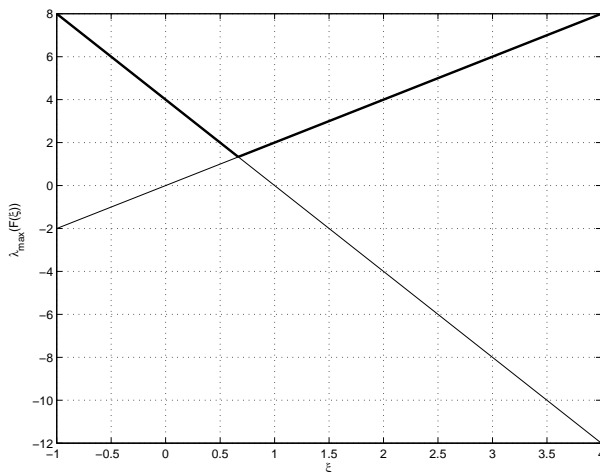


FIG. 3.2 – Tracé de $\lambda_{\max}(F(\xi))$ en fonction de x

⁴Les valeurs propres d'une matrice symétrique sont réelles.

Exemple II Etude du \mathcal{L}_2 gain de systèmes linéaires stationnaires (conditions initiales nulles) :

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

où $w(t)$ est l'entrée du système (influence de l'environnement sur le système) et $z(t)$ est la sortie (influence du système sur l'environnement). Question : quand on considère les signaux sur $[0, T]$, que dire de l'énergie du signal de sortie :

$$\int_0^T z(t)^T z(t) dt$$

par rapport à l'énergie du signal d'entrée $w(t)$:

$$\int_0^T w(t)^T w(t) dt ?$$

Existe-il un réel positif γ tel que pour tout signal d'entrée $w(t)$ d'énergie borné :

$$\int_0^T w(t)^T w(t) dt < +\infty$$

on ait

$$\forall T > 0, \quad \frac{\int_0^T z(t)^T z(t) dt}{\int_0^T w(t)^T w(t) dt} \leq \gamma^2 ? \quad (3.4)$$

(on parle d'atténuation si $\gamma \leq 1$). On démontre que si le système est stable alors un tel γ existe. On recherche alors le plus petit γ tel que l'inégalité (3.4) soit vérifiée. Cette valeur est appelée le \mathcal{L}_2 gain du système. Dans le cas d'un système linéaire stationnaire (cas présent), on démontre que le \mathcal{L}_2 gain du système est égale à la norme H_∞ de la fonction de transfert associée.

On démontre (lemme réel borné) que (3.4) est vérifiée pour toute entrée w bornée et non nulle si et seulement si il existe une matrice symétrique \underline{P} telle que :

$$\left\{ \begin{array}{l} \underline{P} > 0 \\ \left[\begin{array}{cc} A^T \underline{P} + \underline{P} A + C^T C & \underline{P} B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \gamma^2 I \end{array} \right] < 0 \end{array} \right. \quad (3.5)$$

Pour une valeur de γ donnée, l'inégalité matricielle ci-dessus est affine en la variable de décision \underline{P} : elle définit donc une contrainte LMI. Rechercher \underline{P} revient donc à résoudre un problème de faisabilité. Si on désire rechercher le plus petit γ tel qu'il existe \underline{P} vérifiant (3.5) alors γ devient aussi une variable de décision. Cependant, (3.5) dépend de façon affine de γ^2 et non de γ . D'où le changement de variable $\eta = \gamma^2$: minimiser η est équivalent à minimiser γ car γ est positif. Par suite, le problème se réécrit :

$$\begin{array}{ll} \min & \underline{\eta} \\ \underline{\eta} \in \mathbb{R}, \underline{P} = \underline{P}^T \in \mathbb{R}^{n \times n} & \end{array} \quad \left\{ \begin{array}{l} \underline{P} > 0 \\ \left[\begin{array}{cc} A^T \underline{P} + \underline{P} A + C^T C & \underline{P} B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \underline{\eta} I \end{array} \right] < 0 \end{array} \right.$$

Avec :

$$\underline{P} = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i P_i$$

où $P_i, i \in \{1, \dots, \frac{n(n+1)}{2}\}$ une base de l'espace des matrices symétriques de dimension $n \times n$ et $\xi_{\frac{n(n+1)}{2}+1} = \eta$. En posant $F(\xi) = \sum_{i=1}^{\frac{n(n+1)}{2}+1} \xi_i F_i$ avec :

$$\forall i \in \{1, \dots, \frac{n(n+1)}{2}\}, \quad F_i = \begin{bmatrix} P_i & 0 & 0 \\ 0 & -(A^T P_i + P_i A) & -P_i B \\ 0 & -B^T P_i & 0 \end{bmatrix}$$

$$F_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -C^T C & -C^T D \\ 0 & -D^T C & -D^T D \end{bmatrix} \quad \text{et} \quad F_{\frac{n(n+1)}{2}+1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}$$

$c = [0 \ \dots \ 0 \ 1]^T$ et $m = \frac{n(n+1)}{2} + 1$, la recherche du plus petit γ tel qu'il existe \underline{P} vérifiant (3.5) s'écrit comme la minimisation d'un coût linéaire sous contraintes LMI :

$$\begin{aligned} & \text{minimiser} && c^T \xi \\ & \text{pour } \xi \in \mathbb{R}^m \\ & \text{contraint par} && F(\xi) > 0 \end{aligned}$$

La plus petite valeur de γ est alors donnée par la racine carrée du minimum obtenu.

3.1.3 Minimisation de la valeur propre généralisée maximale

$$\begin{aligned} & \text{minimiser} && \lambda \\ & \text{pour } \lambda \in \mathbb{R}, \xi \in \mathbb{R}^m \\ & \text{contraint par} && \lambda F(\xi) - G(\xi) > 0 \\ & && F(\xi) > 0 \text{ et } H(\xi) > 0 \end{aligned}$$

Autre écriture En utilisant la valeur propre généralisée définie page 68,

$$\begin{aligned} & \text{minimiser} && \lambda_{\max}(G(\xi), F(\xi)) \\ & \text{pour } \xi \in \mathbb{R}^m \\ & \text{contraint par} && F(\xi) > 0 \text{ et } H(\xi) > 0 \end{aligned}$$

Propriété c'est un problème d'optimisation quasi convexe car on cherche le minimum d'une fonction de coût quasi convexe ($f(\xi) = \lambda_{\max}(G(\xi), F(\xi))$) pour la variable de décision ξ appartenant à un ensemble convexe défini par les contraintes $F(\xi) > 0$ et $H(\xi) > 0$.

Exemple II : Etude du taux de décroissance des systèmes linéaires stationnaires Soit le système :

$$\dot{x}(t) = Ax(t). \quad (3.6)$$

Pour un système stable, on peut calculer le taux de décroissance exponentiel défini comme le plus grand α positif tel que pour toute condition initiale x_0

$$\lim_{t \rightarrow \infty} e^{\alpha t} \|x(t)\| = 0.$$

On peut relier le taux de décroissance aux valeurs propres de la matrice A . Dans le cas où A est diagonalisable dans \mathbb{C} , en notant $\lambda_i = \sigma_i + j\omega_i$ les valeurs propres de la matrice A , on a :

$$x(t) = \sum_i e^{\sigma_i t} (\alpha_i \cos(\omega_i t) + \beta_i \sin(\omega_i t)).$$

La limite (3.6) sera satisfaite si et seulement si $\sigma_i \leq -\alpha$.

Il est possible de démontrer une condition nécessaire et suffisante : le taux de décroissance est (strictement) supérieur à α s'il existe une fonction $V(x)$ telle que pour $x \neq 0$, $V(x) > 0$ et $\dot{V}(x) < -2\alpha V(x)$. On prend $V(x) = x^T \underline{P} x$ où \underline{P} est une matrice symétrique à déterminer.

$$\begin{cases} V(x) > 0 \\ \dot{V}(x) < -2\alpha V(x) \end{cases} \Leftrightarrow \begin{cases} \underline{P} > 0 \\ A^T \underline{P} + \underline{P} A + 2\alpha \underline{P} < 0 \end{cases} \Leftrightarrow \begin{cases} (-\alpha)(2\underline{P}) - (A^T \underline{P} + \underline{P} A) > 0 \\ 2\underline{P} > 0 \end{cases} \quad (3.7)$$

Rechercher le plus grand α tel que (3.7) soit satisfait revient à minimiser $-\alpha$ tel que (3.7) soit satisfait. En posant

$$\underline{P} = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i P_i$$

avec P_i , précédemment introduit, on peut définir $F(\xi) = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i F_i$ avec $\forall i \in \{1, \dots, \frac{n(n+1)}{2}\}$, $F_i = 2P_i$; $G(\xi) = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i G_i$ avec $\forall i \in \{1, \dots, \frac{n(n+1)}{2}\}$, $G_i = A^T P_i + P_i A$. En posant $\lambda = -\alpha$, la recherche du plus grand taux de décroissance s'écrit alors :

$$\begin{aligned} & \text{minimiser} && \lambda \\ & \text{pour } \lambda \in \mathbb{R}, \xi \in \mathbb{R}^m \\ & \text{contraint par} && \lambda F(\xi) - G(\xi) > 0 \\ & && F(\xi) > 0 \end{aligned}$$

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

Exemple III : Calcul de la borne supérieure de μ dans le cas d'incertitudes complexes
On reprend l'exemple III présenté page 25. On cherche maintenant à calculer la borne supérieure de $\mu_{\underline{\Delta}}$ qui est obtenue par la résolution du problème d'optimisation suivant :

$$\begin{aligned} \mu_{\underline{\Delta}}^{sup}(M) &= \min_{\substack{D \in \mathcal{D} \\ \beta \in \mathbb{R}^+}} \beta \\ & M^* D M - \beta^2 D < 0 \end{aligned}$$

avec

$$\mathcal{D} = \left\{ D \in \mathbb{C}^{k \times k} / D = \begin{bmatrix} d_1 I_{q_1} & 0 & \cdots \\ 0 & \ddots & \ddots \\ \vdots & \ddots & d_q I_{q_q} \end{bmatrix} \text{ avec} \right.$$

$$\left. d_i \in \mathbb{R}, d_i > 0, i \in \{1, \dots, q\} \right\}.$$

On peut réécrire sous forme d'inégalité sur des matrices réelles :

$$\mu^{sup}(M) = \min_{\substack{D \in \mathcal{D} \\ \beta \in \mathbb{R}^+}} \beta$$

$$\begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}^T \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix} - \beta^2 \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} < 0$$

En posant $\eta = \beta^2$

$$\mu^{sup}(M)^2 = \min_{\eta \in \mathbb{R}^+} \eta$$

$$\eta \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} - \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix}^T \begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} \begin{bmatrix} M_R & M_I \\ -M_I & M_R \end{bmatrix} > 0$$

$$\begin{bmatrix} D_R & D_I \\ -D_I & D_R \end{bmatrix} > 0$$

ce qui est un problème de minimisation d'une valeur propre généralisée.

3.2 Problèmes d'optimisation LMI particuliers

Les problèmes d'optimisation LMI contiennent des problèmes d'optimisation convexe classiques : la programmation linéaire et la programmation quadratique (voir la figure 3.3).

Programmation linéaire Etant donné la matrice $A \in \mathbb{R}^{p \times m}$ et le vecteur $b \in \mathbb{R}^p$ et $c \in \mathbb{R}^m$:

$$\begin{aligned} & \text{minimiser} && c^T \xi \\ & \text{pour} && \xi \in \mathbb{R}^m \\ & \text{contraint par} && A\xi \geq b \end{aligned}$$

où \geq signifie, dans cette section, que la i^{ieme} composante du vecteur du vecteur $A\xi$ est supérieure ou égale à la i^{ieme} composante du vecteur du vecteur b .

Exemple d'un problème de programmation linéaire : Calcul d'un régime alimentaire équilibré et économique Un régime alimentaire équilibré doit contenir m nutriments avec des quantités au moins égales à b_1, \dots, b_m . Un aliment contient en général plusieurs nutriments. Une quantité "1" de l'aliment j contient une quantité a_{ij} du nutriment i , pour un coût c_j . On dispose de k aliments. Comment calculer la quantité de chaque aliment à consommer de façon à obtenir un régime alimentaire équilibré, pour le coût le plus faible possible ?

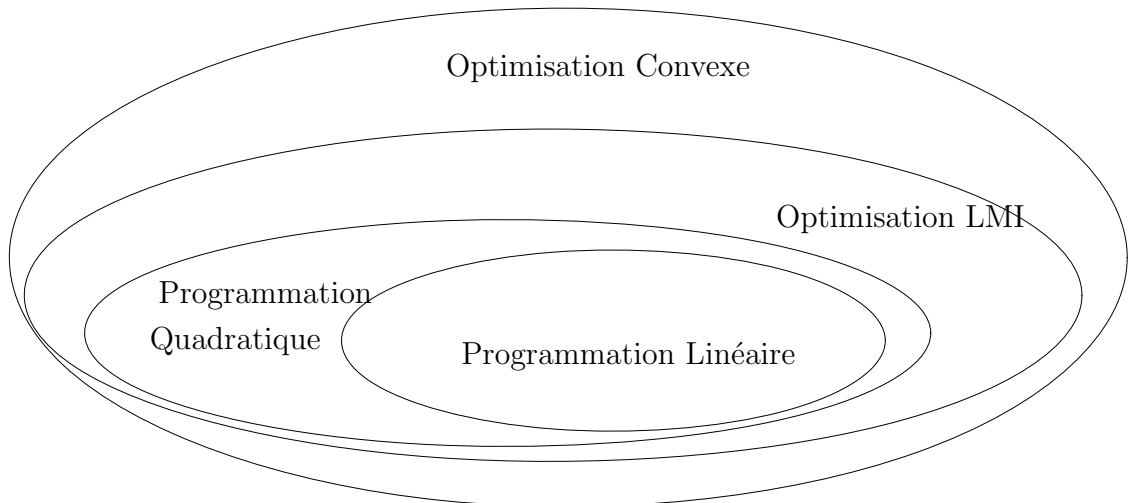


FIG. 3.3 – Différentes classes d'optimisation convexe

Programmation quadratique Etant donnés $A \in \mathbb{R}^{m \times m}$ et $b \in \mathbb{R}^m$, $c \in \mathbb{R}$, $F \in \mathbb{R}^{p \times m}$ et $g \in \mathbb{R}^p$, avec la matrice A semi définie positive :

$$\text{minimiser} \quad \xi^T A \xi + 2b^T \xi + c$$

$$\text{pour } \xi \in \mathbb{R}^m$$

$$\text{contraint par} \quad F\xi \geq g$$

Exercice Démontrer que tout problème de programmation linéaire et tout problème de programmation quadratique peut s'écrire sous la forme d'un problème d'optimisation LMI.

3.3 Exercice : Optimisation sur des variables de décisions “discrètes”

Un certain nombre de problèmes de conception de systèmes se ramène à des problèmes d'optimisation dans lesquels les variables de décision ne peuvent prendre qu'un nombre fini de valeurs, par exemple 0 ou 1 :

$$\min_{x \in \mathcal{C}} c^T x$$

avec $\mathcal{C} = \{x \in \mathbb{R}^m, \quad \forall i \in \{1, \dots, m\}, \quad x_i = 0 \text{ ou } x_i = 1 \text{ et } Ax \geq b\}$ avec $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}^m$. Par $Ax \geq b$, il faut comprendre que chaque composante du vecteur Ax est supérieure ou égale à la composante de b de même indice. De plus, c désigne un vecteur de \mathbb{R}^m . Soit $\mu_{opt} = \min_{x \in \mathcal{C}} c^T x$. Ce problème d'optimisation est dit discret car les variables de décision ne peuvent prendre qu'un nombre fini de valeurs.

En général, un tel problème est complexe à résoudre car même si x ne prend qu'un nombre fini de valeurs, il y en a quand même 2^m . Par suite, l'algorithme naïf qui consiste à tester toutes les valeurs de x présente une complexité exponentielle.

A ce problème d'optimisation est alors associé un autre problème d'optimisation dans lequel les contraintes $\forall i \in \{1, \dots, m\}, x_i = 0$ ou $x_i = 1$ sont remplacées par les contraintes $\forall i \in \{1, \dots, m\}, 0 \leq x_i \leq 1$. Le problème d'optimisation associé s'écrit alors

$$\min_{x \in \underline{\mathcal{C}}} c^T x$$

avec $\underline{\mathcal{C}} = \{x \in \mathbb{R}^m, \forall i \in \{1, \dots, m\}, 0 \leq x_i \leq 1 \text{ et } Ax \geq b\}$

A quelle classe de problèmes d'optimisation convexe appartient le problème associé? Soit $\underline{\mu}_{opt} = \min_{x \in \underline{\mathcal{C}}} c^T x$. Est ce que $\underline{\mu}_{opt} \leq \mu_{opt}$ ou est ce que $\underline{\mu}_{opt} \geq \mu_{opt}$? Justifier en quelques mots.

3.4 Notions sur la résolution de problèmes sous contraintes LMI

Plusieurs méthodes de résolution, avec différentes variantes sont possibles. On se concentre sur une seule : une forme particulière de la méthode des points intérieurs qui est présentée à travers un exemple. Comme il a été vu dans l'exemple présenté page 28, l'optimisation sous contraintes LMI recouvre des problèmes d'optimisation sans contrainte avec une fonction de coût non différentiable. Pour résoudre ce problème la méthode des points intérieurs consiste à se ramener à la résolution d'une série de problèmes d'optimisation différentiables sans contrainte.

3.4.1 Minimisation d'un coût linéaire sous contraintes LMI

On cherche la solution d'un problème de minimisation d'une fonction de coût linéaire, c'est-à-dire un problème de la forme :

$$\begin{aligned} & \text{minimiser} && c^T \xi \\ & \text{pour } \xi \in \mathbb{R}^m && \\ & \text{contraint par} && F(\xi) \geq 0 \end{aligned} \tag{3.8}$$

On suppose que l'on connaît un point strictement faisable ξ^0 , c'est-à-dire que $F(\xi^0) > 0$. En fait, l'algorithme ne fonctionne que s'il existe un point strictement faisable, c'est-à-dire que $\{\xi \mid F(\xi) > 0\} \neq \emptyset$.

Exemple de programmation linéaire

$$\begin{aligned} & \text{minimiser} && \xi_1 + \xi_2 \\ & \text{pour } \xi_1 \in \mathbb{R}, \xi_2 \in \mathbb{R} && \\ & \text{contraint par} && \xi_1 \geq 0, \xi_1 \leq 1, \xi_2 \geq 0, \xi_2 \leq 1 \end{aligned}$$

Ce problème est un cas particulier de (3.8) avec $\xi = [\xi_1 \ \xi_2]^T$, $c = [1 \ 1]^T$ et

$$F(\xi) = \begin{bmatrix} \xi_1 & 0 & 0 & 0 \\ 0 & 1 - \xi_1 & 0 & 0 \\ 0 & 0 & \xi_2 & 0 \\ 0 & 0 & 0 & 1 - \xi_2 \end{bmatrix} \geq 0.$$

Le point $\xi^0 = [0,5 \ 0,5]^T$ est manifestement un point strictement faisable ($F(\xi^0) > 0$). Par suite, la valeur du minimum est telle que $c^T \xi^* \leq c^T \xi^0$.

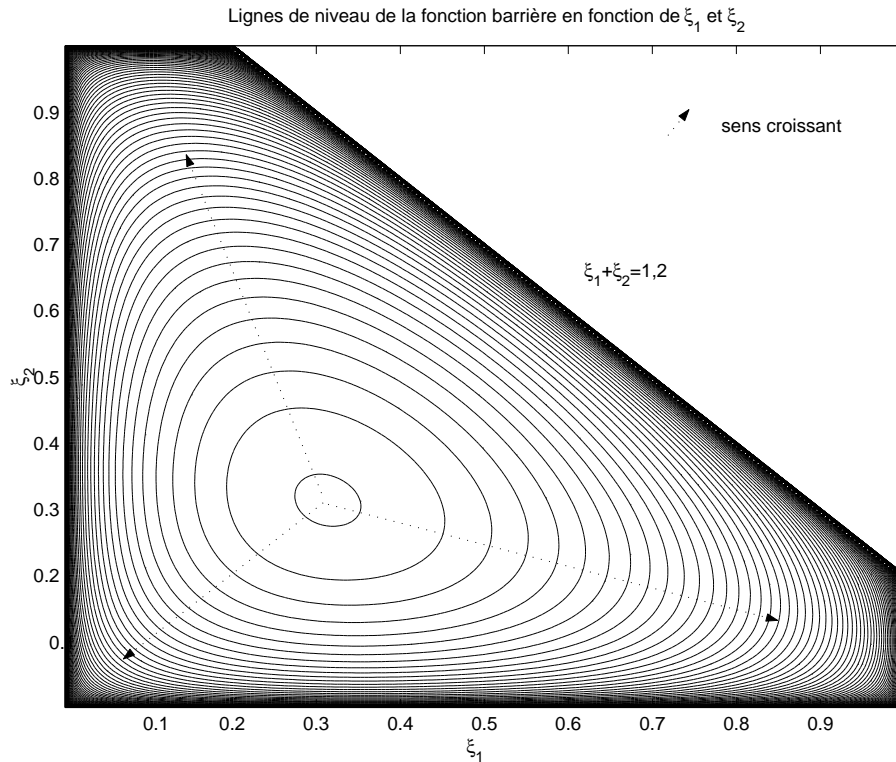


FIG. 3.4 – Courbes de niveau de la fonction barrière $\phi(\xi)$

1. Soit $\lambda^0 > c^T \xi^0$ on considère la région définie par $c^T \xi \leq \lambda^0$ et $F(\xi) \geq 0$. Forcément, le minimum est atteint sur cette région. Pour notre exemple, avec $\lambda^0 = 1,2$, la région est définie par $\xi_1 \geq 0$, $\xi_1 \leq 1$, $\xi_2 \geq 0$, $\xi_2 \leq 1$ et $\lambda^0 > \xi_1 + \xi_2$. A cette région, on peut associer une fonction “barrière” $\phi(\xi)$ définie sur cette région telle $\phi(\xi)$ tende vers $+\infty$ quand ξ tend vers la frontière de la région. Cette fonction est convexe avec un minimum qui correspond au “centre géométrique” de la région. Il est appelé *centre analytique* $\xi^c = \operatorname{argmin}(\phi(\xi))$. La fonction barrière est définie par :

$$\phi(\xi) = -\log(\det(F(\xi))) - \log(\lambda^0 - c^T \xi).$$

Sur notre exemple :

$$\phi(\xi) = -(\log(\xi_1) + \log(1 - \xi_1) + \log(\xi_2) + \log(1 - \xi_2) + \log(1,2 - \xi_1 - \xi_2))$$

(voir figure 3.4).

2. On recherche du minimum de $\phi(\xi)$ (problème d’optimisation convexe, différentiable, “sans contrainte”) par l’algorithme de Newton basé sur :

$$\xi(k+1) = \xi(k) - h(k)(\nabla^2 \phi(\xi(k)))^{-1} \nabla \phi(\xi(k))$$

où $h(k)$ est un coefficient permettant d’améliorer la convergence de l’algorithme de Newton.

3. Si un critère d’arrêt est vérifié ($\xi_c \sim \xi^*$) alors on s’arrête, sinon on effectue les initialisations $\lambda^0 \leftarrow (1 - \theta)c^T \xi^c + \theta c^T \xi^0$, $\xi^0 \leftarrow \xi^c$ avec $\theta \in]0, 1[$ et on retourne à l’étape 1.

La convergence de cet algorithme a été démontrée.

Condition d'application Le domaine défini par $F(\xi) \geq 0$ doit être borné.

Mise en œuvre Il est mis en œuvre dans des boîtes à outils de logiciels de calcul scientifique (numérique) général, commerciaux comme `Matlab` (LMI control toolbox) ou freeware comme `Scilab` (LMI tool). Ce n'est pas forcément le même algorithme qui est utilisé.

Complexité (pratique) Le temps de calcul croît comme un polynôme d'ordre 6 en fonction de la taille du problème.

3.4.2 Recherche d'un point faisable

Pour la résolution du problème précédent, il est nécessaire d'avoir un point strictement faisable (s'il existe) sinon de conclure à $\{\xi \mid F(\xi) > 0\} = \emptyset$.

$$\begin{aligned} & \text{minimiser} && \lambda \\ & \text{pour } \lambda \in \mathbb{R}, \xi \in \mathbb{R}^m && \\ & \text{contraint par} && \lambda I + F(\xi) \geq 0 \end{aligned} \tag{3.9}$$

Ce problème est un problème de minimisation d'un coût linéaire sous contraintes LMI (montrez-le) pour lequel il est simple de trouver un point strictement faisable. Par exemple, on peut choisir $\xi = 0$ et $\lambda > \lambda_{\max}(-F_0)$. Par suite, il peut être résolu par l'algorithme ébauché dans la section 3.4.1. Au minimum λ^* , deux cas :

1. $\lambda^* < 0$: $F(\xi^*) \geq -\lambda^* I > 0$. ξ^* est le point faisable recherché.
2. $\lambda^* \geq 0$: il n'existe pas de ξ^* tel que $F(\xi^*) > 0$ (faire un raisonnement par l'absurde). Par suite, il n'existe pas de point faisable.

3.5 Au delà des contraintes LMIs : les contraintes BMIs

Contrainte Bilinear Matrix inequalities :

$$\mathcal{C} = \{\xi \in \mathbb{R}^m \mid \forall x \in \mathbb{R}^n \setminus \{0\}, \quad x^T B(\xi) x > 0\}$$

avec

$$B(\xi) \triangleq B_0 + \sum_{i=1}^m \xi_i B_i + \sum_{k=1}^m \sum_{l=1}^m \xi_k \xi_l B_{kl}$$

Des problèmes NP difficiles peuvent se formuler comme des problèmes d'optimisation sous contraintes BMI. Mais parfois, pour un problème d'optimisation sous contraintes BMI, il est possible de construire un problème d'optimisation convexe sous contraintes LMIs qui est équivalent.

Exemple

$$\begin{aligned} & \text{minimiser} && \frac{\xi_1^2}{\xi_2} \\ & \text{pour } \xi_1 \in \mathbb{R}, \xi_2 \in \mathbb{R} && \\ & \text{contraint par} && \xi_1 > 0, \xi_2 > 0, 1 > \xi_1^2 + \xi_2^2 \end{aligned}$$

A priori, un problème qui n'est pas LMI. Cependant, en posant $\mu = \frac{\xi_1^2}{\xi_2}$, ce qui revient à faire le changement de variables (ξ_1, ξ_2) par (μ, ξ_2) :

$$\begin{aligned} & \text{minimiser} && \mu \\ & \text{pour } \mu \in \mathbb{R}, \xi_2 \in \mathbb{R} \\ & \text{contraint par} && \mu > 0, \xi_2 > 0, \frac{1}{\xi_2} > \mu + \xi_2 \end{aligned}$$

Ce qui est équivalent à

$$\begin{aligned} & \text{minimiser} && \mu \\ & \text{pour } \mu \in \mathbb{R}, \xi_2 \in \mathbb{R} \\ & \text{contraint par} && \mu > 0, \begin{bmatrix} \frac{1}{\xi_2} & 1 \\ 1 & \frac{1}{\xi_2} - \mu \end{bmatrix} > 0 \end{aligned}$$

En posant $\beta = \frac{1}{\xi_2}$, ce qui revient à faire le changement de variables (μ, ξ_2) par (μ, β) :

$$\begin{aligned} & \text{minimiser} && \mu \\ & \text{pour } \beta \in \mathbb{R}, \mu \in \mathbb{R} \\ & \text{contraint par} && \mu > 0, \begin{bmatrix} \beta & 1 \\ 1 & \beta - \mu \end{bmatrix} > 0 \end{aligned}$$

Ce qui est un problème LMI. A priori, pour transformer de façon équivalente le problème BMI en un problème LMI, on a utilisé des transformations tirées du chapeau. Il est cependant possible de proposer des règles de transformation systématiques. Ces règles sont présentées dans la section suivante.

3.6 A la recherche de la LMI cachée...

Nous allons voir des règles de transformation de problèmes d'optimisation en problèmes d'optimisation sous contraintes LMI. Une contrainte LMI est définie par

$$\mathcal{C} = \{\xi \in \mathbb{R}^m \mid \forall x \in \mathbb{R}^n \setminus \{0\}, \quad x^T F(\xi)x > 0\} \quad \text{avec} \quad F(\xi) \triangleq F_0 + \sum_{i=1}^m \xi_i F_i \quad (3.10)$$

où ξ le vecteur des variables de décision, vecteur de \mathbb{R}^m et où les F_i sont m matrices symétriques données de $\mathbb{R}^{n \times n}$, $i = 0, \dots, m$. Lors de l'établissement d'un problème d'optimisation, les contraintes peuvent être similaires à (3.10) mais différer sous deux points :

1. $F(\xi)$ n'est pas une fonction affine en ξ (par exemple, c'est une fonction biaffine, d'où contrainte BMI) ;
2. l'inégalité $x^T F(\xi)x \geq 0$ est définie pour x appartenant à un sous ensemble de \mathbb{R}^n qui n'est pas \mathbb{R}^n tout entier.

Néanmoins, dans certains cas, il est possible de présenter certaines règles de transformation qui permettent de réécrire le problème d'optimisation en un problème d'optimisation sous contraintes LMI équivalent.

3.6.1 Règles de transformation si $F(\xi)$ n'est pas une fonction affine en ξ

Règle 1 : changement de base dans \mathbb{R}^n Posons $x = T(\xi)\tilde{x}$, avec $\forall \xi \in \mathbb{R}^m$, $T(\xi) \in \mathbb{R}^{m \times m}$ inversible. Par suite :

$$\mathcal{C} = \{\xi \in \mathbb{R}^m \mid \forall x \in \mathbb{R}^n \setminus \{0\}, \quad x^T F(\xi)x > 0\} = \{\xi \in \mathbb{R}^m \mid \forall \tilde{x} \in \mathbb{R}^n \setminus \{0\}, \quad \tilde{x}^T H(\xi)\tilde{x} > 0\}$$

avec $H(\xi) = T(\xi)^T F(\xi) T(\xi)$. Si $H(\xi)$ est une fonction affine en ξ alors \mathcal{C} est bien une contrainte LMI.

Conséquence de la règle 1 : le lemme de Schur

Lemme 3.6.1 (lemme de Schur) Soit une matrice symétrique partitionnée :

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

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

La démonstration est basée sur la remarque que :

$$\begin{bmatrix} I & 0 \\ -B^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} I & -A^{-1} B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix}.$$

Les deux matrices étant congruentes, elles ont mêmes inerties. En fait, le lemme de Schur se ramène à un changement de base dans l'espace des signaux. En effet, au lieu de considérer l'ensemble des q et des p tels que :

$$\begin{bmatrix} q \\ p \end{bmatrix}^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} > 0,$$

nous considérons l'ensemble des \tilde{q} et des \tilde{p} tels que :

$$\begin{bmatrix} \tilde{q} \\ \tilde{p} \end{bmatrix}^T \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \tilde{p} \end{bmatrix} > 0,$$

avec

$$\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} I & -A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \tilde{p} \end{bmatrix}.$$

Intérêt du lemme de Schur Soient $A(\xi)$, $B(\xi)$, $C(\xi)$, $D(\xi)$ quatre matrices fonctions affines en ξ . Alors $C(\xi) - B(\xi)^T A(\xi)^{-1} B(\xi)$ n'est pas une fonction affine en ξ . Néanmoins, d'après le lemme de Schur :

$$\begin{cases} A(\xi) > 0 \\ C(\xi) - B(\xi)^T A(\xi)^{-1} B(\xi) > 0 \end{cases} \Leftrightarrow \begin{bmatrix} A(\xi) & B(\xi) \\ B(\xi)^T & C(\xi) \end{bmatrix} > 0 \Leftrightarrow \begin{bmatrix} C(\xi) & B(\xi)^T \\ B(\xi) & A(\xi) \end{bmatrix} > 0$$

Les deux dernières inégalités sont affines en ξ .

Exercice Soit $A(\xi)$ une matrice fonction affine de ξ . Mettre sous forme d'un problème d'optimisation LMI la recherche de ξ telle que la valeur singulière maximale de $A(\xi)$ soit minimisée.

Remarque En dépit de leur nom, les contraintes LMI peuvent être équivalentes à des contraintes non linéaires. Ce qui qualifie la difficulté d'un problème d'optimisation est plus le fait qu'il soit non convexe que le fait qu'il soit non linéaire.

Exemple (début) Soit le système linéaire stationnaire :

$$\dot{x}(t) = Ax(t) + Bu(t)$$

où $u(t)$ est l'entrée de commande et $x(t)$ est l'état du système, avec $A \in \mathbb{R}^{n \times n}$ et $B \in \mathbb{R}^{n \times p}$. Question : on recherche une loi de commande par retour d'état : $u(t) = \underline{K}x(t)$ (c'est-à-dire la matrice $\underline{K} \in \mathbb{R}^{p \times n}$) tel que le système en boucle fermée :

$$\dot{x}(t) = (A + B\underline{K})x(t) \quad (3.11)$$

soit stable. Il est stable s'il existe une matrice \underline{P} telle que

$$\begin{aligned} \forall x_1 \in \mathbb{R}^n \quad x_1^T \underline{P} x_1 &> 0 \\ \forall x_2 \in \mathbb{R}^n \quad x_2^T ((A + B\underline{K})^T \underline{P} + \underline{P}(A + B\underline{K})) x_2 &< 0 \end{aligned}$$

Problème : la seconde inégalité est bilinéaire en les inconnues \underline{P} et \underline{K} . Posons $x_1 = \underline{P}^{-1} \tilde{x}_1$ et $x_2 = \underline{P}^{-1} \tilde{x}_2$. Alors,

$$\begin{aligned} \forall \tilde{x}_1 \in \mathbb{R}^n \quad \tilde{x}_1^T \underline{P}^{-1} \tilde{x}_1 &> 0 \\ \forall \tilde{x}_2 \in \mathbb{R}^n \quad \tilde{x}_2^T (A \underline{P}^{-1} + \underline{P}^{-1} A^T + B \underline{K} \underline{P}^{-1} + \underline{P}^{-1} \underline{K}^T B^T) \tilde{x}_2 &< 0 \end{aligned} \quad (3.12)$$

Apparemment pas suffisant pour avoir une LMI : à suivre...

Règle 2 : changement de variable sur ξ

Exemple (suite) Les deux inégalités obtenues dans l'exemple (3.12) ne sont pas affines en les variables d'optimisation $\underline{P} \in \mathbb{R}^{n \times n}$ et $\underline{K} \in \mathbb{R}^{p \times n}$. Par contre, en posant $\underline{Q} = \underline{P}^{-1} \in \mathbb{R}^{n \times n}$ et $\underline{Y} = \underline{K} \underline{P}^{-1} \in \mathbb{R}^{p \times n}$, elles sont équivalentes à :

$$\begin{aligned} \forall \tilde{x}_1 \in \mathbb{R}^n \quad \tilde{x}_1^T \underline{Q} \tilde{x}_1 &> 0 \\ \forall \tilde{x}_2 \in \mathbb{R}^n \quad \tilde{x}_2^T (A \underline{Q} + \underline{Q} A^T + B \underline{Y} + \underline{Y}^T B^T) \tilde{x}_2 &< 0 \end{aligned} \quad (3.13)$$

Ce changement de variables est bien posé car la fonction qui relie $(\underline{P}, \underline{K})$ à $(\underline{Q}, \underline{Y})$ est une bijection. De plus, les inégalités (3.13) sont bien affines en les variables \underline{Q} et \underline{Y} . Par suite, la recherche d'une loi de commande $u(t) = \underline{K}x(t)$ stabilisant le système (3.11) peut s'effectuer de la façon suivante :

1. Trouver \underline{Q} et \underline{Y} tels que les inégalités (3.13) soient satisfaites (problème de faisabilité)
2. $\underline{P} = \underline{Q}^{-1}$ et $\underline{K} = \underline{Y} \underline{P}$

Exercice Soit le système linéaire stationnaire discret :

$$x(k+1) = Ax(k) + Bu(k)$$

où $u(k)$ est l'entrée de commande et $x(k)$ est l'état du système, avec $A \in \mathbb{R}^{n \times n}$ et $B \in \mathbb{R}^{n \times p}$.
Question : on recherche une loi de commande par retour d'état : $u(k) = \underline{K}x(k)$ (c'est-à-dire la matrice $\underline{K} \in \mathbb{R}^{p \times n}$) tel que le système en boucle fermée : $x(k+1) = (A + B\underline{K})x(k)$ soit stable. Il est stable s'il existe une matrice \underline{P} telle que

$$\forall x_1 \in \mathbb{R}^n \quad x_1^T \underline{P} x_1 > 0$$

$$\forall x_2 \in \mathbb{R}^n \quad x_2^T ((A + B\underline{K})^T \underline{P} (A + B\underline{K}) - \underline{P}) x_2 < 0$$

A partir de ces conditions, formuler la recherche d'une loi de commande stabilisante sous la forme d'un problème d'optimisation sous contraintes LMI.

Règle 3 : élimination de variables ξ

Lemme 3.6.2 Soit $G = G^T \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{r \times n}$ avec $\text{Rang}(U) = r < n$ et $V \in \mathbb{R}^{s \times n}$ avec $\text{Rang}(V) = s < n$. Alors, il existe une matrice $\underline{K} \in \mathbb{R}^{s \times r}$ telle que l'inégalité suivante est vérifiée :

$$G + U^T \underline{K}^T V + V^T \underline{K} U < 0$$

si et seulement si il existe $\sigma \in \mathbb{R}$:

$$G < \sigma U^T U \quad \text{et} \quad G < \sigma V^T V$$

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 .

Exemple : mise en œuvre sur le problème de synthèse de retour d'état voir chapitre 4.

Règle 4 : Complétion des carrés Elle est basée sur l'égalité :

$$X^T X + X^T Y + Y^T X = (X + Y)^T (X + Y) - Y^T Y.$$

Règle 5 : Introduction de variables supplémentaires (ou "variables bidons")

Exemple Soient $A(\xi)$ et $B(\xi)$ deux matrices fonctions affines en ξ . On veut mettre sous la forme d'un problème d'optimisation LMI la recherche d'une valeur de ξ telle que

$$\begin{cases} \text{Trace}(B(\xi)^T A(\xi)^{-1} B(\xi)) < 1 \\ A(\xi) > 0 \end{cases} \quad (3.14)$$

On introduit alors la variable supplémentaire \underline{X} telle que :

$$\text{Trace}(\underline{X}) < 1 \quad \text{et} \quad \begin{cases} \underline{X} > B(\xi)^T A(\xi)^{-1} B(\xi) \\ A(\xi) > 0 \end{cases} \quad (3.15)$$

Chercher ξ telle que (3.14) soit satisfaite est équivalent à chercher ξ et \underline{X} tels que (3.15) soient satisfaites. D'après le lemme de Schur, les conditions (3.15) sont équivalentes à :

$$\text{Trace}(\underline{X}) < 1 \quad \text{et} \quad \begin{bmatrix} \underline{X} & B(\xi)^T \\ B(\xi) & A(\xi) \end{bmatrix} > 0 \quad (3.16)$$

Exercice Soit un problème de programmation quadratique : étant donnés $A \in \mathbb{R}^{m \times m}$ et $b \in \mathbb{R}^m$, $c \in \mathbb{R}$, $F \in \mathbb{R}^{p \times m}$ et $g \in \mathbb{R}^p$, avec $A > 0$:

$$\begin{aligned} & \text{minimiser} && \xi^T A \xi + 2b^T \xi + c \\ & \text{pour } \xi \in \mathbb{R}^m \\ & \text{contraint par} && F \xi \geq g \end{aligned}$$

Le formuler comme un problème de minimisation d'un coût linéaire sous contraintes LMI.

3.6.2 Règle de transformation quand $x \in \mathcal{X}(\xi)$ avec $\mathcal{X}(\xi) \subset \mathbb{R}^n$ et $\mathcal{X}(\xi) \neq \mathbb{R}^n$

Soit $\mathcal{X}(\xi) \subset \mathbb{R}^n$ et $\mathcal{X}(\xi) \neq \mathbb{R}^n$. Considérons la contrainte définie par :

$$\mathcal{C} = \{ \xi \in \mathbb{R}^m \mid \forall x \in \mathcal{X}(\xi) \setminus \{0\}, \quad x^T F(\xi) x > 0 \}$$

Ce n'est pas une contrainte LMI car $\mathcal{X}(\xi) \neq \mathbb{R}^n$. Dans certain cas, on peut essayer de se ramener à une contrainte LMI, par exemple, dans le cas où :

$$\mathcal{X}(\xi) = \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} x^T F_1(\xi) x > 0 \quad \dots \quad x^T F_p(\xi) x > 0 \\ x^T H_1(\xi) x = 0 \quad \dots \quad x^T H_q(\xi) x = 0 \end{array} \right\}.$$

Exemple : approximation de l'intersection de deux ellipsoïdes par un troisième ellipsoïde Soient deux ellipsoïdes ε_{P_1} et ε_{P_2} de centre 0 définis par les matrices définies positives P_1 et P_2 de dimension $n \times n$:

$$\varepsilon_{P_1} = \{ \tilde{x} \in \mathbb{R}^n \mid \tilde{x}^T P_1^{-1} \tilde{x} < 1 \} \quad \text{et} \quad \varepsilon_{P_2} = \{ \tilde{x} \in \mathbb{R}^n \mid \tilde{x}^T P_2^{-1} \tilde{x} < 1 \}$$

On cherche à approcher leur intersection par un troisième ellipsoïde de centre 0, c'est-à-dire, on recherche le plus petit ellipsoïde de centre 0, défini par la matrice définie positive \underline{P}_0 , contenant l'intersection de ces deux ellipsoïdes. Soit $\varepsilon_{\underline{P}_0}$ cet ellipsoïde. Alors, il contient l'intersection des ellipsoïdes ε_{P_1} et ε_{P_2} si on a

$$\tilde{x}^T \underline{P}_0^{-1} \tilde{x} < 1 \quad \text{pour tout } \tilde{x} \in \mathbb{R}^n \quad \text{tel que} \quad \tilde{x}^T P_1^{-1} \tilde{x} < 1 \quad \text{et} \quad \tilde{x}^T P_2^{-1} \tilde{x} < 1$$

Ce qui peut se réécrire :

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} -P_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} > 0$$

$$\text{pour tout } \tilde{x} \in \mathbb{R}^n \text{ tel que } \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} -P_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} > 0 \text{ et } \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} -P_2^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} > 0$$

ce qui est équivalent à :

$$x^T \begin{bmatrix} -P_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} x > 0 \tag{3.17}$$

$$\text{pour tout } x \in \mathbb{R}^{n+1} \text{ tel que } x^T \begin{bmatrix} -P_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} x > 0 \text{ et } x^T \begin{bmatrix} -P_2^{-1} & 0 \\ 0 & 1 \end{bmatrix} x > 0$$

Lemme 3.6.3 (S-procédure) *S'ils existent* $\tau_1 \in \mathbb{R}^+, \dots, \tau_p \in \mathbb{R}^+, \mu_1 \in \mathbb{R}, \dots, \mu_q \in \mathbb{R}$ *tels que :*

$$\forall x \in \mathbb{R}^n, \quad x^T \left(F(\xi) - \sum_{i=1}^p \tau_i F_i(\xi) - \sum_{j=1}^q \mu_j H_j(\xi) \right) x > 0 \tag{3.18}$$

alors

$$\forall x \in \mathcal{X}(\xi), \quad x^T F(\xi) x > 0 \tag{3.19}$$

Exercice Le démontrer.

Exemple : approximation de l'intersection de deux ellipsoïdes.. (suite) Par application de la \mathcal{S} procédure, une condition suffisante pour que la condition (3.17) soit vérifiée est que :

$$\exists \tau_1 \in \mathbb{R}^+, \exists \tau_2 \in \mathbb{R}^+, \quad \begin{bmatrix} -P_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} > \tau_1 \begin{bmatrix} -P_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} + \tau_2 \begin{bmatrix} -P_2^{-1} & 0 \\ 0 & 1 \end{bmatrix} \tag{3.20}$$

Ce qui est équivalent à :

$$\exists \tau_1 \in \mathbb{R}^+, \exists \tau_2 \in \mathbb{R}^+, \quad \begin{bmatrix} -P_0^{-1} + \tau_1 P_1^{-1} + \tau_2 P_2^{-1} & 0 \\ 0 & 1 - \tau_1 - \tau_2 \end{bmatrix} > 0 \tag{3.21}$$

ce qui est équivalent à $\tau_1 > 0, \tau_2 > 0, 1 > \tau_1 + \tau_2$ et

$$-P_0^{-1} + \tau_1 P_1^{-1} + \tau_2 P_2^{-1} > 0$$

soit par le lemme de Schur :

$$\tau_1 > 0, \tau_2 > 0, 1 > \tau_1 + \tau_2, \quad \begin{bmatrix} \tau_1 P_1^{-1} + \tau_2 P_2^{-1} & I \\ I & P_0 \end{bmatrix} > 0$$

Les variables de décision sont ici τ_1, τ_2 et P_0 : la contrainte ci-dessus est LMI puisqu'affine en les variables de décision. En mesurant la taille de l'ellipsoïde ε_{P_0} par la trace de la matrice P_0 , trouver le plus petit ellipsoïde revient à résoudre un problème de minimisation d'un coût linéaire sous contraintes LMI (écrivez le explicitement). L'intérêt de la \mathcal{S} procédure peut être discuté de façon plus générale.

Conséquence de la \mathcal{S} -procédure Posons $\mathbf{F}(\xi, \tau_i, \mu_j) = F(\xi) - \sum_{i=1}^p \tau_i F_i(\xi) - \sum_{j=1}^q \mu_j H_j(\xi)$. Le cas intéressant est celui où \mathbf{F} est fonction affine de ξ, τ_i et μ_j .

$$\mathcal{C}' = \{\xi \in \mathbb{R}^m, \tau_i \in \mathbb{R}, \mu_j \in \mathbb{R}^+ \mid \forall x \in \mathbb{R} \setminus \{0\}, x^T \mathbf{F}(\xi, \tau_i, \mu_j) x > 0\}$$

Notez que \mathcal{C}' est une contrainte LMI. Soit $\xi \in \mathbb{R}^m$ tel qu'il existe $\tau_i \in \mathbb{R}^+, \mu_j \in \mathbb{R}$ pour lesquels $(\xi, \tau_i, \mu_j) \in \mathcal{C}'$. Alors d'après la \mathcal{S} procédure, $\xi \in \mathcal{C}$. Par suite, dans le cas de

– **Problème de Faisabilité** remplacer le problème d'optimisation :

$$\text{trouver } \xi \in \mathbb{R}^m$$

$$\text{tel que } \xi \in \mathcal{C}$$

par

$$\text{trouver } \xi \in \mathbb{R}^m, \tau_i \in \mathbb{R}^+, \mu_j \in \mathbb{R}$$

$$\text{tel que } (\xi, \tau_i, \mu_j) \in \mathcal{C}'$$

revient à faire une **restriction**. Si le second problème permet de trouver un point faisable $(\xi^{faisa}, \tau_i^{faisa}, \mu_j^{faisa})$ alors ξ^{faisa} est solution du premier problème. Si le second problème n'admet pas de point faisable $(\xi^{faisa}, \tau_i^{faisa}, \mu_j^{faisa})$ alors on ne peut pas conclure sur l'existence d'un point faisable pour le premier problème.

– **Minimisation d'un coût linéaire** remplacer le problème d'optimisation :

$$\text{minimiser } c^T \xi$$

$$\text{pour } \xi \in \mathcal{C}$$

par

$$\text{minimiser } c^T \xi$$

$$\text{pour } (\xi, \tau_i, \mu_j) \in \mathcal{C}'$$

revient à faire une **restriction** : la résolution du second problème donnera une borne supérieure sur la valeur du minimum pour le premier problème.

3.6.3 Exercice : mise sous forme de problèmes d'optimisation LMI

Soit $F(\xi)$ une matrice de fonctions dépendant linéairement du vecteur réel :

$$\xi = \begin{bmatrix} \xi_1 \\ \cdots \\ \xi_n \end{bmatrix} \in \mathbb{R}^n.$$

Peut-on mettre les problèmes suivant sous forme de problèmes d'optimisation convexe sous contraintes LMI :

1. minimiser sur $\xi \in \mathbb{R}^n$ la valeur propre maximale de $F(\xi)$;
2. maximiser sur $\xi \in \mathbb{R}^n$ la valeur propre minimale de $F(\xi)$;
3. minimiser sur $\xi \in \mathbb{R}^n$ la valeur propre maximale de $F(\xi)^T X F(\xi)$ où X est une matrice donnée, symétrique, définie positive.

Si oui, donner le problème d'optimisation LMI correspondant.

Chapitre 4

Formulation de problèmes d'Automatiques sous forme de problèmes d'optimisation LMI : exemple de la synthèse H_∞

4.1 Synthèse de correcteur H_∞

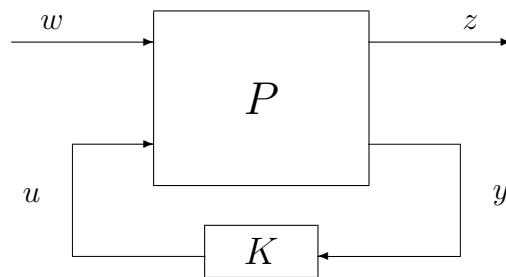


FIG. 4.1 – Problème sous forme standard

Soit le système P décrit sur le schéma bloc représenté figure 4.1 où

- le vecteur des sorties commandées $z(t)$ est de dimension p_z ;
- le vecteur des entrées de critère $w(t)$ est de dimension m_w ;
- le vecteur des sorties mesurées¹ $y(t)$ est de dimension p_y ;
- le vecteur des entrées de commande $u(t)$ est de dimension m_u .

Le système P admet pour équations d'état :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_w w(t) + B_u u(t) \\ z(t) &= C_z x(t) + D_{zw} w(t) + D_{zu} u(t) \\ y(t) &= C_y x(t) + D_{yw} w(t) + D_{yu} u(t) \end{aligned} \tag{4.1}$$

¹En fait, ce signal regroupe les signaux qui entrent dans la loi de commande : ce ne sont pas forcément les signaux mesurés !

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

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

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

soit

$$z(p) = \underbrace{(P_{zw}(p) + P_{zu}(p)K(p)(I - P_{yu}(p)K(p))^{-1}P_{yw}(p))}_{\text{noté } P \star K \text{ ou } \mathcal{F}_l(P,K)} w(p)$$

Problème H_∞ standard

1. Etant donné $\gamma > 0$, existe-il une loi de commande K telle que
 - le système bouclé $P \star K$ soit asymptotiquement stable (tous les pôles du système en boucle fermée sont à partie réelle strictement négative)
 - $\|P \star K\|_\infty \leq \gamma$
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 œuvre numériquement sur un ordinateur. Les deux méthodes de résolution ont été programmées sous `Matlab`. La première approche repose sur la résolution d'une série d'équations de Riccati. 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 hypothèses² sont :

Hypothèses solution par équations de Riccati

1. $\text{rang}(D_{zu}) = m_u$ et $\text{rang}(D_{yw}) = p_y$: 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 ($p_z \geq m_u$) et qu'il y a plus d'entrées de critère w que de mesures y ($m_w \geq p_y$).
2. $\text{rang} \begin{bmatrix} A - j\omega I_n & B_u \\ C_z & D_{zu} \end{bmatrix} = n + m_u$ garantit que le transfert P_{zu} n'a pas de zéro sur l'axe imaginaire.

²En plus de l'hypothèse évidente de stabilisabilité et détectabilité nécessaire pour un système en boucle fermée asymptotiquement stable.

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

Une deuxième solution au problème H_∞ standard permet de faire l'économie de ces hypothèses. Une telle approche 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. Par soucis de simplicité, seul le cas du retour d'état, ou "information complète", (c'est-à-dire, que le correcteur mesure l'état du système : $y(t) = x(t)$ soit $C_y = I$, $D_{yw} = 0$ et $D_{yu} = 0$) sera considéré. Le cas général nécessite des développements techniques assez lourds qui n'apportent rien à la compréhension des idées sous-jacentes. Sa présentation peut être trouvée dans la référence [23].

4.2 Synthèse d'un correcteur H_∞ par retour d'état

Dans le cas du retour d'état, le système P admet pour équations d'état :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_w w(t) + B_u u(t) \\ z(t) &= C_z x(t) + D_{zw} w(t) + D_{zu} u(t) \\ y(t) &= x(t) \end{aligned}$$

On recherche une loi de commande par retour d'état (c'est-à-dire une matrice $\underline{K} \in \mathbb{R}^{p \times n}$) telle que $u(t) = \underline{K}x(t)$ assure :

1. le système bouclé est stable ;
2. 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.

Lemma 4.2.1 (Réel borné) *Soit un système linéaire stationnaire multivariable stable H décrit par une représentation d'état minimale :*

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

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

$$\begin{cases} \underline{P} \geq 0 \\ \begin{bmatrix} A^T \underline{P} + \underline{P} A + C^T C & \underline{P} B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \gamma^2 I \end{bmatrix} \leq 0 \end{cases} \quad (4.2)$$

Synthèse directe

Le système en boucle fermée admet pour équations d'état :

$$\begin{aligned} \dot{x}(t) &= (A + B_u \underline{K})x(t) + B_w w(t) \\ z(t) &= (C_z + D_{zu} \underline{K})x(t) + D_{zw} w(t) \end{aligned}$$

En appliquant le lemme 4.2.1 au système en boucle fermée, il existe un correcteur par retour d'état s'il existe une matrice \underline{K} et une matrice symétrique \underline{P} telles que $\underline{P} > 0$ et :

$$\begin{bmatrix} (A + B_u \underline{K})^T \underline{P} + \underline{P}(A + B_u \underline{K}) + (C_z + D_{zu} \underline{K})^T (C_z + D_{zu} \underline{K}) & \underline{P} B_w + (C_z + D_{zu} \underline{K})^T D_{zw} \\ B_w^T \underline{P} + D_{zw}^T (C_z + D_{zu} \underline{K}) & D_{zw}^T D_{zw} - \gamma^2 I \end{bmatrix} \leq 0 \quad (4.3)$$

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 (4.3) peut se réécrire :

$$\begin{bmatrix} A^T \underline{P} + \underline{P} A + \underline{K}^T B_u^T \underline{P} + \underline{P} B_u \underline{K} & \underline{P} B_w \\ B_w^T \underline{P} & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} (C_z + D_{zu} \underline{K})^T \\ D_{zw}^T \end{bmatrix} \begin{bmatrix} C_z + D_{zu} \underline{K} & D_{zw} \end{bmatrix} \leq 0 \quad (4.4)$$

Lemme 4.2.1 (lemme de Schur) *Soit une matrice symétrique partitionnée :*

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}$$

où \mathbf{A} et \mathbf{C} sont des matrices carrées. Cette matrice est définie positive si et seulement si \mathbf{A} et $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ sont définies positives.

Lemme 4.2.2 (lemme de Schur modifié) *Soit une matrice symétrique partitionnée :*

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}$$

où \mathbf{A} et \mathbf{C} sont des matrices carrées, avec \mathbf{A} définie positive. Cette matrice est définie positive si et seulement si $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ est semi définie positive.

L'inégalité (4.4) peut alors se réécrire :

$$\underbrace{\begin{bmatrix} A^T \underline{P} + \underline{P} A + \underline{K}^T B_u^T \underline{P} + \underline{P} B_u \underline{K} & \underline{P} B_w \\ B_w^T \underline{P} & -\gamma^2 I \end{bmatrix}}_{\mathbf{C}} - \underbrace{\begin{bmatrix} (C_z + D_{zu} \underline{K})^T \\ D_{zw}^T \end{bmatrix}}_{\mathbf{B}^T} \underbrace{(-I)}_{\mathbf{A}^{-1}} \underbrace{\begin{bmatrix} C_z + D_{zu} \underline{K} & D_{zw} \end{bmatrix}}_{\mathbf{B}} \leq 0 \quad (4.5)$$

Par application du lemme de Schur modifié, on a alors l'inégalité (4.5) qui est équivalente à :

$$\begin{bmatrix} A^T \underline{P} + \underline{P} A + \underline{K}^T B_u^T \underline{P} + \underline{P} B_u \underline{K} & \underline{P} B_w & (C_z + D_{zu} \underline{K})^T \\ B_w^T \underline{P} & -\gamma^2 I & D_{zw}^T \\ C_z + D_{zu} \underline{K} & D_{zw} & -I \end{bmatrix} \leq 0 \quad (4.6)$$

L'intérêt de cette transformation est que, dans l'inégalité obtenue, il ne reste plus que $\underline{P} B_u \underline{K}$ et $\underline{K}^T B_u^T \underline{P}$ 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 :

Propriété $\forall \mathbf{T}$ inversible $\in \mathbb{R}^{n \times n}$, $\mathbf{A} \leq 0 \Leftrightarrow \mathbf{T}^T \mathbf{A} \mathbf{T} \leq 0$.

En appliquant cette propriété avec :

$$\mathbf{T} = \begin{bmatrix} \underline{P}^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \quad \text{et} \quad \mathbf{A} = \begin{bmatrix} A^T \underline{P} + \underline{P}A + \underline{K}^T B_u^T \underline{P} + \underline{P}B_u \underline{K} & \underline{P}B_w & (C_z + D_{zu} \underline{K})^T \\ B_w^T \underline{P} & -\gamma^2 I & D_{zw}^T \\ C_z + D_{zu} \underline{K} & D_{zw} & -I \end{bmatrix}$$

on obtient :

$$\begin{bmatrix} \underline{P}^{-1} A^T + A \underline{P}^{-1} + \underline{P}^{-1} \underline{K}^T B_u^T + B_u \underline{K} \underline{P}^{-1} & B_w & \underline{P}^{-1} (C_z^T + \underline{K}^T D_{zu}^T) \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{P}^{-1} + D_{zu} \underline{K} \underline{P}^{-1} & D_{zw} & -I \end{bmatrix} \leq 0 \quad (4.7)$$

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

$$\begin{bmatrix} \underline{Q} A^T + A \underline{Q} + \underline{Y}^T B_u^T + B_u \underline{Y} & B_w & \underline{Q} C_z^T + \underline{Y}^T D_{zu}^T \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{Q} + D_{zu} \underline{Y} & D_{zw} & -I \end{bmatrix} \leq 0 \quad (4.8)$$

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

Ce changement de variables est bien posé car la fonction qui relie $(\underline{P}, \underline{K})$ et $(\underline{Q}, \underline{Y})$ est une bijection. De plus, les inégalités (4.8) et $\underline{Q} > 0$ sont bien affines en les variables \underline{Q} et \underline{Y} . Rechercher \underline{Q} et \underline{Y} tels que $\underline{Q} > 0$ et (4.8) est donc un problème de faisabilité LMI.

En conclusion, la recherche d'une loi de commande $u(t) = \underline{K}x(t)$ assurant la stabilité de la boucle fermée et une norme H_∞ entre w et z inférieure à γ s'obtient de la façon suivante :

1. Trouver \underline{Q} et \underline{Y} tels que les inégalités $\underline{Q} > 0$ et (4.8) soient satisfaites (problème de faisabilité)
2. $\underline{P} = \underline{Q}^{-1}$ et $\underline{K} = \underline{Y} \underline{P}$

Existence d'un correcteur H_∞

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

Lemme 4.2.3 (Lemme d'élimination) Soit $\mathbf{G} = \mathbf{G}^T \in \mathbb{R}^{n \times n}$, $\mathbf{U} \in \mathbb{R}^{r \times n}$ avec $\mathbf{Rang}(\mathbf{U}) = r < n$ et $\mathbf{V} \in \mathbb{R}^{s \times n}$ avec $\mathbf{Rang}(\mathbf{V}) = s < n$. Alors, il existe une matrice $\mathbf{K} \in \mathbb{R}^{r \times s}$ telle que l'inégalité suivante est vérifiée :

$$\mathbf{G} + \mathbf{U}^T \mathbf{K} \mathbf{V} + \mathbf{V}^T \mathbf{K}^T \mathbf{U} < 0$$

si et seulement si :

$$\mathbf{U}_\perp^T \mathbf{G} \mathbf{U}_\perp < 0 \quad \text{et} \quad \mathbf{V}_\perp^T \mathbf{G} \mathbf{V}_\perp < 0,$$

avec

- \mathbf{U}_\perp engendre le noyau de l'application linéaire associée à la matrice³ \mathbf{U} ;
- \mathbf{V}_\perp engendre le noyau de l'application linéaire associée à la matrice \mathbf{V} .

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

L'inégalité (4.8) peut se réécrire :

$$\underbrace{\begin{bmatrix} \underline{Q}A^T + A\underline{Q} & B_w & \underline{Q}C_z^T \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -I \end{bmatrix}}_{\mathbf{G}} + \underbrace{\begin{bmatrix} B_u \\ 0 \\ D_{zu} \end{bmatrix}}_{\mathbf{U}^T} \underbrace{\begin{bmatrix} \underline{Y} \\ \mathbf{K} \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} I & 0 & 0 \end{bmatrix}}_{\mathbf{V}} + \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} \underline{Y}^T \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix} \leq 0 \quad (4.9)$$

En appliquant le lemme 4.2.3, l'inégalité 4.9 est équivalente à

$$\mathbf{U}_\perp^T \mathbf{G} \mathbf{U}_\perp \leq 0 \Leftrightarrow \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp^T \begin{bmatrix} \underline{Q}A^T + A\underline{Q} & B_w & \underline{Q}C_z^T \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -I \end{bmatrix} \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp \leq 0$$

et à

$$\mathbf{V}_\perp^T \mathbf{G} \mathbf{V}_\perp \leq 0 \Leftrightarrow \begin{bmatrix} I & 0 & 0 \end{bmatrix}_\perp^T \begin{bmatrix} \underline{Q}A^T + A\underline{Q} & B_w & \underline{Q}C_z^T \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \end{bmatrix}_\perp \leq 0.$$

Comme

$$\begin{bmatrix} I & 0 & 0 \end{bmatrix}_\perp = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix}$$

on a :

$$\mathbf{V}_\perp^T \mathbf{G} \mathbf{V}_\perp \leq 0 \Leftrightarrow \begin{bmatrix} -\gamma^2 I & D_{zw}^T \\ D_{zw} & -I \end{bmatrix} \leq 0 \Leftrightarrow D_{zw}^T D_{zw} \leq \gamma^2 I.$$

En résumé, une condition nécessaire et suffisante d'existence d'un correcteur par retour d'état assurant que

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

est donnée par

1. $D_{zw}^T D_{zw} \leq \gamma^2 I$ (valeur singulière maximale de D_{zw} inférieure ou égale à γ)

³Les vecteurs colonnes de \mathbf{U}_\perp forment une base du noyau.

2. l'existence d'une matrice \underline{Q} telle que $\underline{Q} > 0$ et

$$\begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp^T \begin{bmatrix} \underline{Q}A^T + A\underline{Q} & B_w & \underline{Q}C_z^T \\ B_w^T & -\gamma^2 I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -I \end{bmatrix} \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp \leq 0.$$

On a donc un problème de faisabilité LMI dans lequel le correcteur n'est pas recherché explicitement.

Cette solution a des liens très forts avec la solution par équations de Riccati par retour d'état (information complète). Rappelons que contrairement à la solution par équations de Riccati, il n'est pas nécessaire que les hypothèses présentées page 46 soient vérifiées.

4.3 Lien (ébauche) avec la solution par équation de Riccati

Dans la solution par équations de Riccati, pour obtenir des expressions plus simples, on introduit les conditions supplémentaires suivantes :

$$D_{zw} = 0 \quad D_{zu}^T [C_z \ D_{zu}] = [0 \ I_{m_u}] \quad (4.10)$$

La condition $D_{zu}^T D_{zw} \leq \gamma^2 I$ est automatiquement vérifiée.

Par suite, un choix possible⁴ pour $\begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp$ peut être :

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -D_{zu} B_u^T & 0 & C_z C_z^T \end{bmatrix}$$

Par suite, en remplaçant $\begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp$ par l'expression ci-dessus, on obtient :

$$\mathbf{U}_\perp^T \mathbf{G} \mathbf{U}_\perp \leq 0 \Leftrightarrow \begin{bmatrix} \underline{Q}A^T + A\underline{Q} - B_u B_u^T & B_w & \underline{Q}C_z^T C_z C_z^T \\ B_w^T & -\gamma^2 I & 0 \\ C_z C_z^T C_z \underline{Q} & 0 & -C_z C_z^T C_z C_z^T \end{bmatrix} \leq 0$$

En supposant que C_z est de rang plein et par application du Lemme de Schur modifié, ceci est équivalent à

$$\underline{Q}A^T + A\underline{Q} - B_u B_u^T + \gamma^{-2} B_w B_w^T + \underline{Q}C_z^T C_z \underline{Q} \leq 0$$

Soit, en post et pré multipliant par $\underline{P} = \underline{Q}^{-1}$:

$$\underline{P}A + A^T \underline{P} + \underline{P}(\gamma^{-2} B_w B_w^T - B_u B_u^T) \underline{P} + C_z^T C_z \leq 0.$$

Cette inégalité de Riccati correspond à l'égalité de Riccati de la solution par équation de Riccati :

$$X_\infty A + A^T X_\infty + X_\infty (\gamma^{-2} B_w B_w^T - B_u B_u^T) X_\infty + C_z^T C_z = 0.$$

⁴Il n'est pas unique.

4.4 Synthèse H_∞ par retour de sortie

Le théorème suivant donne la solution dans le cas où l'état n'est pas mesuré par formulation LMI.

Théorème 4.4.1 *Soit le système P d'ordre n défini par (4.1). Etant donné $\gamma > 0$, il existe un correcteur K d'ordre inférieur ou égal à n , défini par la représentation d'état :*

$$\begin{aligned}\dot{x}_K(t) &= A_K x_K(t) + B_K y(t) \\ u(t) &= C_K x(t) + D_K y(t)\end{aligned}$$

telle que

- le système bouclé $P \star K$ soit asymptotiquement stable
(tous les pôles du système en boucle fermée sont à partie réelle strictement négative)
- $\|P \star K\|_\infty \leq \gamma$

si et seulement si il existe deux matrices \underline{P} et \underline{Q} telles que :

$$\begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp^T \begin{bmatrix} \underline{Q}A^T + A\underline{Q} & B_w & \underline{Q}C_z^T \\ B_w^T & -\gamma I & D_{zw}^T \\ C_z \underline{Q} & D_{zw} & -\gamma I \end{bmatrix} \begin{bmatrix} B_u^T & 0 & D_{zu}^T \end{bmatrix}_\perp \leq 0 \quad (4.11)$$

$$\begin{bmatrix} C_y & D_{yw} & 0 \end{bmatrix}_\perp^T \begin{bmatrix} A^T \underline{P} + \underline{P}A & \underline{P}B_w & C_z^T \\ B_w^T \underline{P} & -\gamma I & D_{zw}^T \\ C_z & D_{zw} & -\gamma I \end{bmatrix} \begin{bmatrix} C_y & D_{yw} & 0 \end{bmatrix}_\perp \leq 0 \quad (4.12)$$

et

$$\begin{bmatrix} \underline{P} & I \\ I & \underline{Q} \end{bmatrix} \geq 0 \quad (4.13)$$

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.

Principe de la démonstration Comme dans le cas du retour d'état, le résultat s'obtient en appliquant le lemme réel borné au système en boucle fermée en ayant soin de faire apparaître dans

sa représentation d'état les matrices d'état du correcteur rassemblées dans une matrice par blocs :

$$\begin{aligned}
\begin{bmatrix} \dot{x}(t) \\ \dot{x}_K(t) \end{bmatrix} &= \left(\underbrace{\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}}_{\hat{A}} + \underbrace{\begin{bmatrix} 0 & B_u \\ I & 0 \end{bmatrix}}_{\hat{B}_u} \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_{\hat{K}} \underbrace{\begin{bmatrix} 0 & I \\ C_y & 0 \end{bmatrix}}_{\hat{C}_y} \right) \underbrace{\begin{bmatrix} x(t) \\ x_K(t) \end{bmatrix}}_{\hat{x}(t)} \cdots \\
&+ \left(\underbrace{\begin{bmatrix} B_w \\ 0 \end{bmatrix}}_{\hat{B}_w} + \underbrace{\begin{bmatrix} 0 & B_u \\ I & 0 \end{bmatrix}}_{\hat{B}_u} \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_{\hat{K}} \underbrace{\begin{bmatrix} 0 \\ D_{yw} \end{bmatrix}}_{\hat{D}_{yw}} \right) w(t) \\
z(t) &= \left(\underbrace{\begin{bmatrix} C_z & 0 \\ 0 & 0 \end{bmatrix}}_{\hat{C}_z} + \underbrace{\begin{bmatrix} 0 & D_{zu} \end{bmatrix}}_{\hat{D}_{zu}} \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_{\hat{K}} \underbrace{\begin{bmatrix} 0 & I \\ C_y & 0 \end{bmatrix}}_{\hat{C}_y} \right) \begin{bmatrix} x(t) \\ x_K(t) \end{bmatrix} \\
&+ \left(D_{zw} + \begin{bmatrix} 0 & D_{zu} \end{bmatrix} \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_{\hat{K}} \underbrace{\begin{bmatrix} 0 \\ D_{yw} \end{bmatrix}}_{\hat{D}_{yw}} \right) w(t)
\end{aligned}$$

Ceci peut s'interpréter comme commander le système P augmenté de n_K intégrateurs :

$$\begin{aligned}
\hat{\dot{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}_w w(t) + \hat{B}_u \hat{u}(t) \\
z(t) &= \hat{C}_z \hat{x}(t) + D_{zw} w(t) + \hat{D}_{zu} \hat{u}(t) \\
\hat{y}(t) &= \hat{C}_y \hat{x}(t) + \hat{D}_{yw} w(t)
\end{aligned}$$

par le correcteur par retour statique de sortie :

$$u(t) = \hat{K} \hat{y}(t).$$

L'étape suivante est d'appliquer le lemme réel borné (énoncé page 47). On obtient alors une formulation similaire à la formulation 4.5 : il existe une matrice \hat{P} définie positive telle que :

$$\begin{bmatrix} \hat{A}^T \hat{P} + \hat{P} \hat{A} + \hat{C}_y^T \hat{K}^T \hat{B}_u^T \hat{P} + \hat{P} \hat{B}_u \hat{K} \hat{C}_y & \hat{P} \hat{B}_w & (\hat{C}_z + \hat{D}_{zu} \hat{K} \hat{C}_y)^T \\ \hat{B}_w^T \hat{P} & -\gamma^2 I & (\hat{D}_{zw} + \hat{D}_{zu} \hat{K} \hat{D}_{yw})^T \\ \hat{C}_z + \hat{D}_{zu} \hat{K} \hat{C}_y & \hat{D}_{zw} + \hat{D}_{zu} \hat{K} \hat{D}_{yw} & -I \end{bmatrix} \leq 0 \quad (4.14)$$

Par application du lemme d'élimination, on obtient les conditions (4.11) et (4.12) dans lesquelles on a P et Q telles que :

$$\hat{P} = \begin{bmatrix} P & \star \\ \star & \star \end{bmatrix} \quad \text{et} \quad \hat{P}^{-1} = \begin{bmatrix} Q & \star \\ \star & \star \end{bmatrix}.$$

La condition (4.13) garantit que $\hat{P} > 0$.

4.5 Exemples

4.5.1 Correcteur avec effet intégral

On désire commander un système G de représentation d'état :

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

Seule la sortie à commander de G est mesurée. Le correcteur admet pour entrée l'erreur de suivi de référence $\epsilon(t) = r(t) - y(t)$. L'objectif est de faire du suivi d'échelon de référence avec une certaine rapidité et une erreur statique nulle. La commande devra être raisonnable.

Pour assurer un bon suivi de référence, il est nécessaire de contraindre la fonction de transfert entre r et ϵ ($T_{r \rightarrow \epsilon}$). Cela est fait par l'introduction d'une pondération $W_1(p)$. On recherche alors le correcteur satisfaisant

$$\forall \omega |T_{r \rightarrow \epsilon}(j\omega)| \leq \frac{1}{|W_1(j\omega)|}.$$

Pour avoir une erreur statique nulle, il est nécessaire d'avoir $T_{r \rightarrow \epsilon}(0) = 0$. Pour imposer cela, W_1 doit contenir un intégrateur, ce qui n'est pas possible car toute pondération doit avoir ses pôles à partie strictement négative.

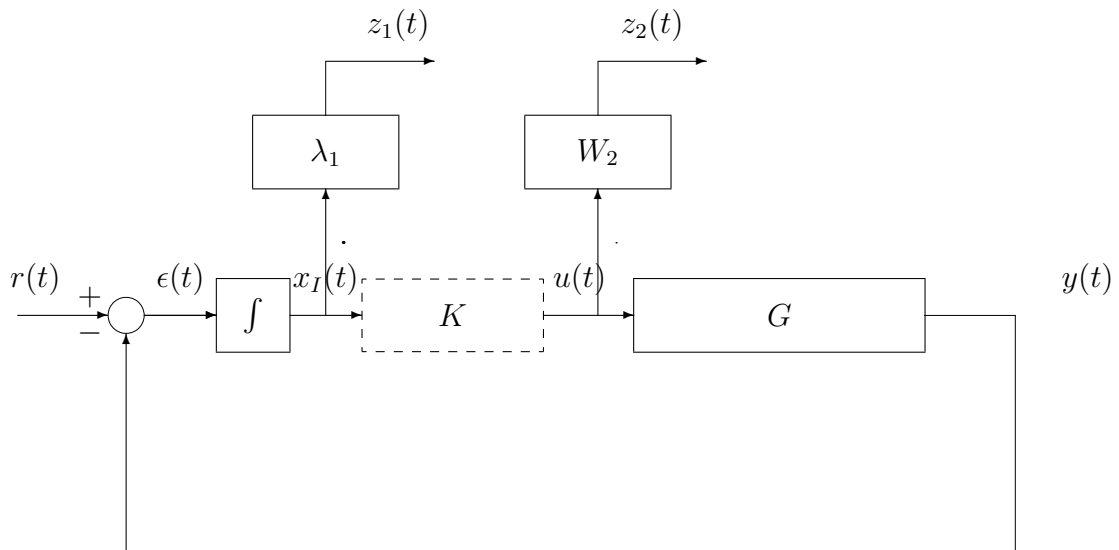


FIG. 4.2 – Commande intégrale

Afin d'avoir une erreur statique nulle, on décide d'introduire a priori et explicitement un intégrateur dans le correcteur. On obtient le schéma 4.2.

- Le gain λ_1 va permettre de contraindre $|T_{r \rightarrow \epsilon}(j\omega)|$ puisque si $\gamma \leq 1$ on a

$$\left| \frac{\lambda_1}{j\omega} T_{r \rightarrow \epsilon}(j\omega) \right| \leq 1 \Leftrightarrow |T_{r \rightarrow \epsilon}(j\omega)| \leq \frac{|\omega|}{\lambda_1}$$

– La pondération W_2 permet de limiter la bande passante du correcteur un imposant :

$$|T_{r \rightarrow u}(j\omega)| \leq \frac{1}{|W_2(j\omega)|}$$

avec $\frac{1}{W_2(p)}$ qui est un filtre passe-bas. W_2 admet pour représentation d'état :

$$\begin{aligned} \dot{x}_{W_2}(t) &= A_{W_2}x_{W_2}(t) + B_{W_2}u(t) \\ z_2(t) &= C_{W_2}x_{W_2}(t) \end{aligned}$$

Le système mis sous forme standard s'écrit alors

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{x}_I(t) \\ \dot{x}_{W_2}(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} A & 0 & 0 \\ -C & 0 & 0 \\ 0 & 0 & A_{W_2} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x(t) \\ x_I(t) \\ x_{W_2}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{\mathbf{B}_w} r(t) + \underbrace{\begin{bmatrix} B \\ 0 \\ B_{W_2} \end{bmatrix}}_{\mathbf{B}_u} u(t) \\ \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & \lambda_1 & 0 \\ 0 & 0 & C_{W_2} \end{bmatrix}}_{\mathbf{C}_z} \begin{bmatrix} x(t) \\ x_I(t) \\ x_{W_2}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\mathbf{D}_{zw}} r(t) + \underbrace{\begin{bmatrix} 0 \\ D_{W_2} \end{bmatrix}}_{\mathbf{D}_{zu}} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}_y} \begin{bmatrix} x(t) \\ x_I(t) \\ x_{W_2}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{\mathbf{D}_{yw}} r(t) \end{aligned}$$

Comme \mathbf{D}_{yw} est nul, il n'est pas de rang 1. On ne peut donc pas appliquer la solution par équation de Riccati.

4.5.2 Rejet de perturbation mesurée avec une certaine dynamique

On considère la mise au point d'un correcteur qui doit assurer le suivi de trajectoire de référence et assurer le rejet d'une perturbation b en entrée du système à commander. Une solution classique consiste à choisir un correcteur à un degré de liberté :

$$u(p) = K(p)(r(p) - y(p)).$$

On suppose que la perturbation est en fait mesurée avec une certaine dynamique, c'est-à-dire qu'il existe une fonction de transfert passe bas connue $F(p)$ (représentant le capteur) telle que l'on mesure le signal de transformée de Laplace $F(p)b(p)$. Par suite, on décide de choisir la structure de commande

$$u(p) = K(p) \begin{bmatrix} F(p)b(p) \\ r(p) - y(p) \end{bmatrix}.$$

$K(p)$ a donc deux entrées et une sortie. (voir le schéma 4.3).

Montrer que dans ce cas la matrice \mathbf{D}_{yw} n'est pas de rang égal à $m_u = 2$ et que la solution par équation de Riccati ne peut pas être utilisée.

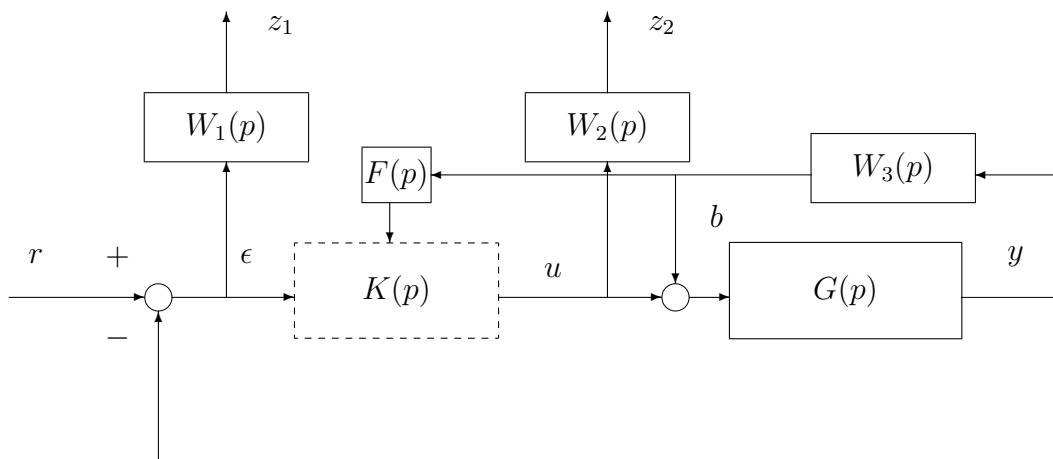


FIG. 4.3 – Rejet de perturbation mesurée avec une certaine dynamique

Chapitre 5

Formulation de problèmes d'Automatiques sous forme de problèmes d'optimisation LMI : au-delà des systèmes linéaires stationnaires

Un des grands apports de l'introduction de l'optimisation LMI, c'est de permettre de traiter de problèmes d'Automatique impliquant des systèmes non linéaires et/ou non stationnaires. Les aspects de manipulations et de mise en œuvre de l'optimisation LMI sont similaires à ce qui a été vu dans les chapitres précédents. Par contre, la présentation de ces résultats demandent l'introduction de concepts d'Automatique qui sortent largement du cadre de ce cours. Par suite, on se contentera de la solution à un problème particulier simple.

5.1 Systèmes à paramètres variant dans le temps

Soient A_i , i allant de 1 à L , L matrices carrées. On considère le système linéaire non stationnaire défini par les équations :

$$\dot{x}(t) = \sum_{i=1}^L \lambda_i(t) A_i x(t) \quad (5.1)$$

où pour i allant de 1 à L , les paramètres variant dans le temps $\lambda_i(t)$ peuvent prendre toutes les valeurs possibles telles que $\lambda_i(t) \geq 0$ et $\sum_{i=1}^L \lambda_i(t) = 1$.

5.2 Analyse de la stabilité

On cherche à démontrer la stabilité du système défini par 5.1. Une condition *suffisante* de stabilité est donnée par l'existence d'une matrice $\underline{P} > 0$ telle que :

$$\forall \lambda_i(t) \geq 0, \quad \sum_i \lambda_i(t) = 1, \quad \left(\sum_{i=1}^L \lambda_i(t) A_i \right)^T \underline{P} + \underline{P} \left(\sum_{i=1}^L \lambda_i(t) A_i \right) < 0 \quad (5.2)$$

(voir la similitude de cette condition avec la condition de stabilité des systèmes linéaires stationnaires introduite page 24).

La difficulté est que trouver $\underline{P} > 0$ telle que (5.2) soit satisfaite est un problème d'optimisation défini par un nombre infini de contraintes paramétrisées par λ_i . Néanmoins il est possible de le transformer en un problème d'optimisation de dimension finie par le lemme suivant.

Lemme 5.2.1

$$\forall \lambda_i \geq 0, \sum_i \lambda_i = 1, \Omega_0 + \sum_{i=1}^L \lambda_i \Omega_i > 0 \Leftrightarrow \forall i \in \{1, \dots, L\}, \Omega_0 + \Omega_i > 0$$

Par application de ce lemme, trouver $\underline{P} > 0$ telle que la condition (5.2) soit satisfaite est équivalent à trouver $\underline{P} > 0$ telle que

$$\forall i \in \{1, \dots, L\}, \quad A_i^T \underline{P} + \underline{P} A_i < 0 \quad (5.3)$$

ce qui est un problème d'optimisation de dimension finie.

5.3 Exemple d'application sur un missile

L'application considérée est la commande de la voie de tangage d'un missile, c'est-à-dire la commande d'un missile suivant le plan vertical. Celle-ci est constituée de deux boucles imbriquées :

1. la boucle (externe) de guidage qui en fonction de la position de l'engin et de la trajectoire désirée délivre des consignes en accélération à la boucle interne ;
2. la boucle (interne) de pilotage qui à partir de la mesure de l'accélération et de la vitesse de rotation du missile essaye de suivre les consignes en accélération délivrées par la voie de guidage.

C'est la seconde boucle qui nous intéresse ici, en ne considérant que les mouvements du missile dans le plan vertical.

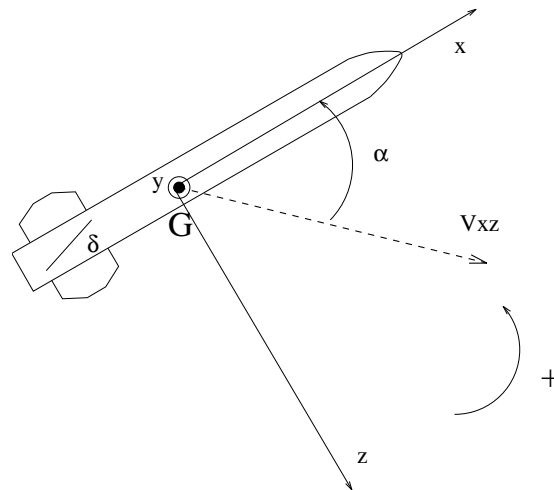


FIG. 5.1 – Missile dans le plan vertical

Le missile est représenté sur la figure 5.1. G désigne le centre de gravité. Le trièdre (Gx, Gy, Gz) est lié au missile. Le plan vertical est celui défini par (Gx, Gz) . V_{xz} est la projection de la vitesse du missile par rapport au vent sur le plan vertical. L'angle α entre V_{xz} et Gx est appelé l'angle d'incidence. On introduit la vitesse de rotation q du missile autour du point G avec les

conventions habituelles d'orientation. Le but du correcteur de pilotage est de suivre des demandes d'accélération η_c suivant Gz en échelon en agissant sur la déflexion δ de la gouverne correspondante. Il possède des caractéristiques intéressantes par rapport à ce qui a été vu précédemment : il est (fortement) non linéaire. Pour plus de détails sur la description d'un missile, se rapporter à la thèse [20].

Le système à analyser se compose donc du missile et du correcteur de la boucle interne. Dans ce qui suit, nous allons considérer le modèle proposé dans l'article [21].

Modèle du missile

En prenant comme variables d'état α et q pour le missile, la représentation d'état est :

$$\begin{aligned}\dot{\alpha} &= \cos(\alpha)K_\alpha MC_n(\alpha, \delta, M) + q \\ \dot{q} &= K_q M^2 C_m(\alpha, \delta, M)\end{aligned}$$

avec les coefficients de portance qui sont donnés par :

$$\begin{aligned}C_n(\alpha, \delta, M) &= a_n \alpha^3 + b_n |\alpha| \alpha + c_n (2 - M/3) \alpha + d_n \delta \\ C_m(\alpha, \delta, M) &= a_m \alpha^3 + b_m |\alpha| \alpha + c_m (-7 + 8M/3) \alpha + d_m \delta\end{aligned}$$

On peut estimer que leur domaine de validité est défini par des angles d'incidence appartenant à une plage de ± 20 degrés ($\pm 0,35$ radians). La sortie à commander est donnée par :

$$\eta = \frac{K_z}{g_{\text{grav}}} M^2 C_n(\alpha, \delta, M)$$

La dynamique de l'actionneur peut être modélisée par :

$$\ddot{\delta} = -\omega_a^2 \delta - 2\xi_a \omega_a \dot{\delta} + \omega_a^2 \delta_c$$

La signification et les valeurs numériques des différents grandeurs sont données dans les tableaux 5.1 et 5.2 à la fin de la section.

Correcteur PI linéaire

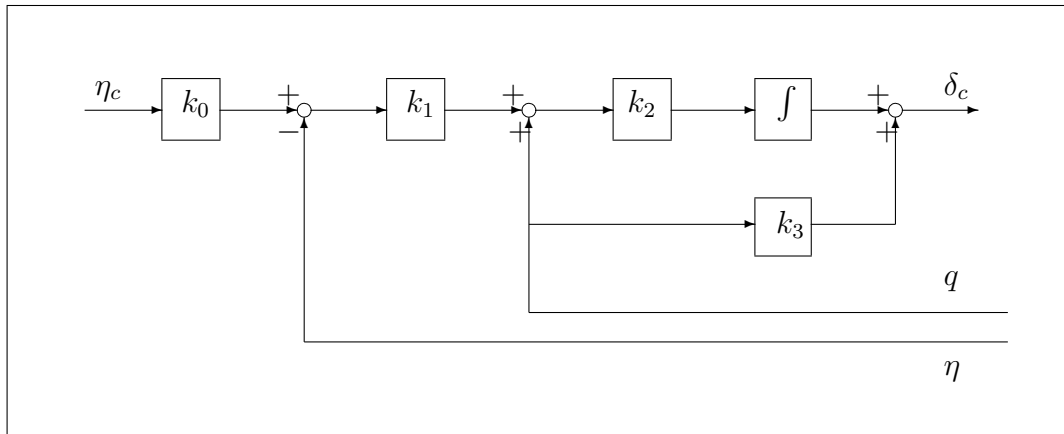


FIG. 5.2 – Structure du correcteur PI

Un correcteur proportionnel intégral de la forme de la figure 5.2 a été synthétisé. L'expression du PI est :

$$\dot{\delta}_c = k_3 \dot{q} + k_2 q + (k_1 k_2)(k_0 \eta_c - \eta)$$

Modélisation du système

Le modèle du système obtenu en supposant que $\cos(\alpha) = 1$, que $a_m \approx 2a_n$ et que $b_m \approx 2b_n$ est le suivant :

$$\dot{x} = (A + \theta(t)A_\theta)x$$

où x_{PI} est la variable d'état associée au PI et avec

$$A = \begin{bmatrix} K_\alpha M c_n (2 - \frac{M}{3}) & 1 & K_\alpha M d_n & 0 & 0 \\ K_q M^2 c_m (-7 + \frac{8M}{3}) & 0 & K_q M^2 d_m & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & k_3 w_a^2 & -w_a^2 & -2\xi_a w_a & w_a^2 \\ -k_1 k_2 \frac{K_z}{grav} M^2 c_n (2 - \frac{M}{3}) & k_2 & -k_1 k_2 \frac{K_z}{grav} M^2 d_n & 0 & 0 \end{bmatrix}$$

$$\theta(t) = a_n \alpha^2 + b_n |\alpha|$$

$$A_\theta = \begin{bmatrix} K_\alpha M & 0 & 0 & 0 & 0 \\ 2K_q M^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -k_1 k_2 \frac{K_z}{grav} M^2 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{et} \quad x = \begin{bmatrix} \alpha \\ q \\ \delta \\ \dot{\delta} \\ x_{PI} \end{bmatrix}$$

Vu la forme de la matrice A_θ , la non linéarité impliquée est $\theta(t)\alpha$. La non linéarité impliquée se trouve figure 5.3. On peut l'encadrer dans deux droites de pente -10 et 0 .

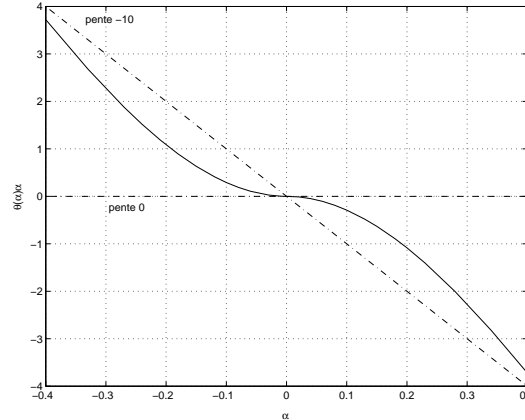


FIG. 5.3 – Secteur englobant la non linéarité du missile ($\theta(t)\alpha$)

Par suite, le missile peut être défini par un système LPV avec $L = 2$, $A_1 = A$ et $A_2 = A - 10A_\theta$. On trouve alors \underline{P} tel que :

$$A_1^T \underline{P} + \underline{P} A_1 < 0 \quad \text{et} \quad A_2^T \underline{P} + \underline{P} A_2 < 0$$

Valeurs des variables du modèle

Missile La signification des grandeurs est donnée dans le tableau 5.1. Les valeurs numériques sont données dans le tableau 5.2. Les unités ne sont pas celles du système international.

α	angle d'incidence en radians
q	vitesse de rotation dans le plan (Gx, Gz) (rad/s)
M	nombre de Mach
δ	angle de gouverne en radians
δ_c	commande d'angle de gouverne en radians
η_c	consigne d'accélération en g
η	accélération en g

TAB. 5.1 – Différentes grandeurs du modèle.

a_n	$1,0286 \times 10^{-4}$	deg^{-3}	P_0	$973,3$	lb/ft^2
b_n	$-0,94457 \times 10^{-2}$	deg^{-2}	S	$0,44$	ft^2
c_n	$-0,1696$	deg^{-1}	m	$13,98$	$slugs$
d_n	$-0,034$	deg^{-1}	V	$1036,4$	ft/s
a_m	$2,1524 \times 10^{-4}$	deg^{-3}	d	$0,75$	ft
b_m	$-1,9546 \times 10^{-2}$	deg^{-2}	I_y	$182,5$	$slug.ft^2$
c_m	$0,051$	deg^{-1}	K_α	$0,7PoS/m/V$	
d_m	$-0,206$	deg^{-1}	K_q	$0,7PoSd/I_y$	
ω_a	150		K_z	$0,7PoS/m$	
ξ_a	$0,7$		$grav$	$32,2$	

TAB. 5.2 – Valeurs numériques des paramètres.

PI Un réglage correct du PI est obtenu pour les valeurs :

$$k_0 = 1,12, \quad k_1 = 0,0867, \quad k_2 = 2,5, \quad \text{et} \quad k_3 = 0,5$$

Chapitre 6

Annexe A : rappels sur les matrices

6.1 Rappels sur les matrices

Propriété Sauf cas particuliers, étant données deux matrices A et B appartenant à $\mathbb{C}^{n \times n}$:

$$AB \neq BA$$

Definition 6.1.1 Soit $A \in \mathbb{C}^{n \times n}$. Le nombre complexe λ est valeur propre de A si :

$$\exists v \in \mathbb{C}^n, \quad v \neq 0, \quad Av = \lambda v.$$

Le vecteur v est un vecteur propre associé à la valeur propre λ . A possède alors n valeurs propres notées $\lambda_i(A)$, $i \in \{1, \dots, n\}$.

Interprétation On peut associer à la matrice A une application linéaire :

$$\begin{aligned} \mathbb{C}^n &\rightarrow \mathbb{C}^n \\ x &\mapsto y = Ax \end{aligned}$$

Pour x vecteur propre associé à une valeur propre λ de A , le vecteur image y est alors obtenu par la multiplication de x par le scalaire λ . Suivant la direction définie par x , A “se comporte” alors comme un scalaire.

Caractérisation λ est valeur propre de A si et seulement si $\det(\lambda I_n - A) = 0$ où I_n représente la matrice identité. De plus, $\det(\lambda I_n - A)$ est un polynôme en λ appelé polynôme caractéristique et noté $P_c(\lambda)$. Les valeurs propres de A sont les racines du polynôme $P_c(\lambda)$: les valeurs propres de A peuvent être ainsi calculées en calculant les racines de $P_c(\lambda)$. Une fois que les valeurs propres sont déterminées, les vecteurs propres peuvent être obtenus en résolvant l'équation en v :

$$(A - \lambda I_n)v = 0.$$

Definition 6.1.2 Une matrice $A \in \mathbb{C}^{n \times n}$ est inversible (ou non singulière) s'il existe une matrice, notée $A^{-1} \in \mathbb{C}^{n \times n}$, telle que :

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

Propriété 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 $\det(A) = \prod_{i=1}^n \lambda_i(A)$).

Propriété $(AB)^{-1} = B^{-1}A^{-1}$

Definition 6.1.3 La trace d'une matrice A (pas forcément symétrique) est la somme des éléments de sa diagonale. Notation : $\text{Trace}(A)$.

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

Propriété $\text{Trace}(AB) = \text{Trace}(BA)$

Definition 6.1.4 L'inertie d'une matrice A est un triplet de 3 entiers indiquant le nombre de valeurs propres positives, le nombre de valeurs propres négatives et le nombre de valeurs propres nulles.

Definition 6.1.5 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 $\{\lambda_i(A)\} = \bigcup_{j=1}^r \{\lambda_i(A_j)\}$

Décomposition en valeurs singulières d'une matrice complexe Pour la matrice $A \in \mathbb{C}^{m \times n}$, avec¹ $m \geq n$: elle est définie par 3 matrices dont le produit est :

$$A = U\Sigma V^* \tag{6.1}$$

où $U \in \mathbb{C}^{m \times m}$, $UU^* = I_m$, $V \in \mathbb{C}^{n \times n}$, $VV^* = I_n$ et

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \\ \hline 0 & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \quad \text{avec } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0.$$

Les scalaires réels σ_i sont appelés *valeurs singulières* de la matrice complexe A . Ce ne sont rien d'autre que les racines carrées des valeurs propres de la matrice² A^*A , c'est-à-dire $\sqrt{\lambda(A^*A)}$. La plus grande valeur singulière σ_1 définit une norme sur la matrice A . Le nombre de valeurs singulières de A non nulles indique le rang de la matrice A .

La commande `Matlab svd` permet d'obtenir tous les éléments de la décomposition en valeurs singulières d'une matrice (complexe). Son calcul est en général numériquement plus fiable que celui de la décomposition en valeurs propres.

¹Dans le cas où $n > m$, la décomposition en valeurs singulières se définit de façon similaire.

²Cette matrice étant symétrique, ces valeurs propres sont réelles. De plus, comme elle est (semi) définie positive, ses valeurs propres sont positives ou nulles.

Exemples Avec

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

on obtient :

$$U = \begin{bmatrix} 0.2673 & -0.5345 & -0.8018 \\ 0.5345 & 0.7745 & -0.3382 \\ 0.8018 & -0.3382 & 0.4927 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 3.7417 \\ 0 \\ 0 \end{bmatrix} \quad V = [1]$$

Avec

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 4 & 5 & 6 + j \end{bmatrix}$$

on obtient :

$$U = \begin{bmatrix} -0.3588 + 0.0278j & -0.5926 + 0.1391j & 0.6867 + 0.1685j \\ -0.3588 + 0.0278j & -0.5926 + 0.1391j & -0.6867 - 0.1685j \\ -0.8607 - 0.0139j & 0.5042 - 0.0695j & 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 10.2408 & 0 & 0 \\ 0 & 1.0607 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4063 & 0.7840 & 0.4694 \\ -0.5604 + 0.0041j & 0.1420 + 0.1967j & -0.7222 - 0.3250j \\ -0.7158 + 0.0922j & -0.5655 - 0.0819j & 0.3250 + 0.2166j \end{bmatrix}$$

La matrice A étant de rang 2, on observe que seules deux valeurs singulières sont non nulles.

Avant d'énoncer un certain nombre de propriétés, il est intéressant de regarder le cas des matrices réelles de dimension 2×2 . Dans ce cas-là, il existe deux scalaires θ_1 et θ_2 tels que :

$$A = \underbrace{\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix}}_U \underbrace{\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}}_\Sigma \underbrace{\begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) \end{bmatrix}^T}_{V^T}$$

Par suite, la matrice U (resp. V) a ses colonnes orthonormales : elles correspondent à des matrices associées à des rotations : la matrice U (resp. V) à une rotation d'angle θ_1 (resp. θ_2). Si on applique la matrice A à un vecteur x , celui-ci subit d'abord une rotation définie par la matrice V^T puis chacune de ces composantes est dilatée par σ_i avant de subir une nouvelle rotation définie par U .

Propriétés On a les propriétés suivantes :

Prop. 1 ³

$$\sigma_n(A) = \min_{\|v\| \neq 0} \frac{\|Av\|}{\|v\|}$$

Soient v_1, v_2, \dots, v_n les n vecteurs colonnes de la matrice V . Alors $\|Av_n\| = \sigma_n(A)$. De plus, $Av_n = \sigma_n u_n$, où u_n est la $n^{\text{ième}}$ colonne de la matrice U .

³Cette propriété n'est pas vérifiée pour $n > m$.

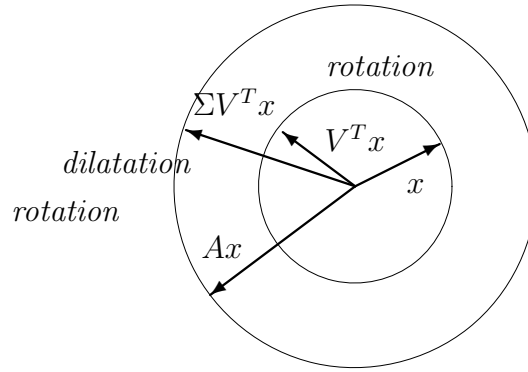


FIG. 6.1 – Transformation de v par A quand $\sigma_1 = 2$ et $\sigma_2 = 1$

Prop. 2

$$\sigma_1(A) = \max_{\|v\| \neq 0} \frac{\|Av\|}{\|v\|}$$

Comme précédemment, $Av_1 = \sigma_1 u_1$, où u_1 est la première colonne de la matrice U . Par suite, $\|Av_1\| = \sigma_1(A)$.

Prop. 3 $\forall i \in \{1, \dots, n\}$, $Av_i = \sigma_i u_i$.

Prop. 4 Pour une matrice complexe A inversible,

$$\sigma_1(A) = \frac{1}{\sigma_n(A^{-1})} \quad \text{et} \quad \sigma_n(A) = \frac{1}{\sigma_1(A^{-1})}.$$

Prop. 5 Pour deux matrices complexes A et B , un scalaire complexe α ,

1. $\sigma_1(\alpha A) = |\alpha| \sigma_1(A)$;
2. $\sigma_1(A + B) \leq \sigma_1(A) + \sigma_1(B)$ (inégalité triangulaire);
3. $\sigma_1(AB) \leq \sigma_1(A) \sigma_1(B)$ (propriété de gain).

6.1.1 Cas des matrices carrées et réelles

Si $A \in \mathbb{R}^{n \times n}$ alors ses valeurs propres sont soit réelles soit complexes conjuguées. Pour une valeur propre complexe λ de A , sa conjuguée $\bar{\lambda}$ est aussi valeur propre de A .

Propriété Toute matrice s'écrit comme la somme d'une matrice symétrique et d'une matrice anti symétrique : $A = S + G$ avec $S = S^T$ et $G = -G^T$. L'ensemble des matrices de dimension $n \times n$ forme un espace vectoriel de dimension n^2 , celui des matrices symétriques de dimension $n \times n$ forme un espace vectoriel de dimension $\frac{n(n+1)}{2}$ et celui des matrices anti symétriques de dimension $n \times n$ forme un espace vectoriel de dimension $\frac{n(n-1)}{2}$.

Propriété Si A est symétrique ($A = A^T$) alors ses valeurs propres sont réelles. De plus, ses valeurs singulières sont le module de ses valeurs propres.

Notations

$$\begin{aligned}\lambda_{max}(A) &= \max_{i \in \{1, \dots, n\}} \lambda_i(A) \\ \lambda_{min}(A) &= \min_{i \in \{1, \dots, n\}} \lambda_i(A)\end{aligned}$$

Definition 6.1.6 Si A est une matrice symétrique ($A = A^T$) alors on peut lui associer une forme quadratique, c'est-à-dire une fonction de \mathbb{R}^n dans \mathbb{R} qui à x associe $q_A(x) = x^T Ax$.

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).

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

Caractérisation des matrices définies positives 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 :

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

Propriété L'ensemble des matrices définies positives est un cône convexe.

Propriété Si $A > 0$ alors $\exists T$ inversible $\in \mathbb{R}^{n \times n}$, $A = T^T T$. Notation : $A^{1/2}$

Propriété $\forall T$ inversible $\in \mathbb{R}^{n \times n}$, $A > 0 \Leftrightarrow T^T A T > 0$.

Definition 6.1.7 (Relation d'ordre (partiel))

$$A \geq B \Leftrightarrow A - B \text{ est une matrice semi définie positive}$$

Caractérisations variationnelles Si A est une matrice symétrique alors

$$\lambda_{max}(A) = \max_{x \neq 0} \frac{x^T Ax}{x^T x} = \max_{x^T x = 1} x^T Ax = \min_{\lambda \in \mathbb{R}} \lambda \quad \lambda I - A \geq 0$$

et

$$\lambda_{min}(A) = \min_{x \neq 0} \frac{x^T Ax}{x^T x} = \min_{x^T x = 1} x^T Ax = \max_{\lambda \in \mathbb{R}} \lambda \quad \lambda I - A \leq 0$$

Par suite :

$$\begin{aligned}\lambda > \lambda_{max}(A) &\Rightarrow \lambda I > A \\ \lambda < \lambda_{min}(A) &\Rightarrow \lambda I < A\end{aligned}$$

Definition 6.1.8 (Produit scalaire de deux matrices) Le produit scalaire de deux matrices A et B noté par $\langle A, B \rangle$ est défini par :

$$\langle A, B \rangle = \text{Trace}(AB)$$

Propriété Si $A \geq 0$ et $B \geq 0$ alors $\langle A, B \rangle \geq 0$.

Definition 6.1.9 (Valeurs propres généralisées) Soit $A \in \mathbb{R}^{n \times n}$ et soit $B \in \mathbb{R}^{n \times n}$, $B = B^T$ et $B > 0$. Les valeurs propres généralisées de (A, B) sont les valeurs propres de la matrice⁴ $M = B^{-1/2}AB^{-1/2}$. De façon équivalente, le nombre complexe λ est valeur propre généralisée de (A, B) si :

$$\exists v \in \mathbb{C}^n, \quad v \neq 0, \quad Av = \lambda Bv.$$

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

$$\lambda_{max}(A, B) = \min_{\lambda \in \mathbb{R}} \lambda \quad \lambda B - A \geq 0$$

⁴Etant donnée une matrice $B = B^T$, $B > 0$, il est toujours possible de construire une matrice notée $B^{-1/2}$ telle que $B^{1/2} * B^{1/2} = B$, Propriété énoncée précédemment.

Chapitre 7

Annexe B : Ensembles convexes particuliers

7.1 Ellipsoïdes

C'est un domaine de \mathbb{R}^n défini par :

$$\varepsilon = \{x \in \mathbb{R}^n \mid (x - x_c)^T P^{-1} (x - x_c) < 1\}$$

avec $P = P^T \in \mathbb{R}^{n \times n}$ définie positive et $x_c \in \mathbb{R}^n$: *centre de l'ellipsoïde*. La matrice P décrit l'extension de l'ellipsoïde dans chaque direction de \mathbb{R}^n . Par exemple pour $n = 1$, l'ellipsoïde est un intervalle de centre x_c et de largeur $2\sqrt{P}$.

Propriété Cette représentation d'un ellipsoïde est unique.

Demi-axes de l'ellipsoïde Soit la décomposition de P :

$$P = \sum_{i=1}^n \lambda_i v_i v_i^T$$

où $\lambda_i, i \in \{1, \dots, n\}$, sont les valeurs propres de P , classées par ordre décroissant et où les vecteurs propres associés v_i sont orthonormaux. Chaque *demi-axe* de l'ellipsoïde est défini par le segment $[x_c, x_c + \sqrt{\lambda_i} v_i]$. L'excentricité est mesurée par $\frac{\lambda_1}{\lambda_n}$. La figure 7.1 correspond à un ellipsoïde défini par la matrice :

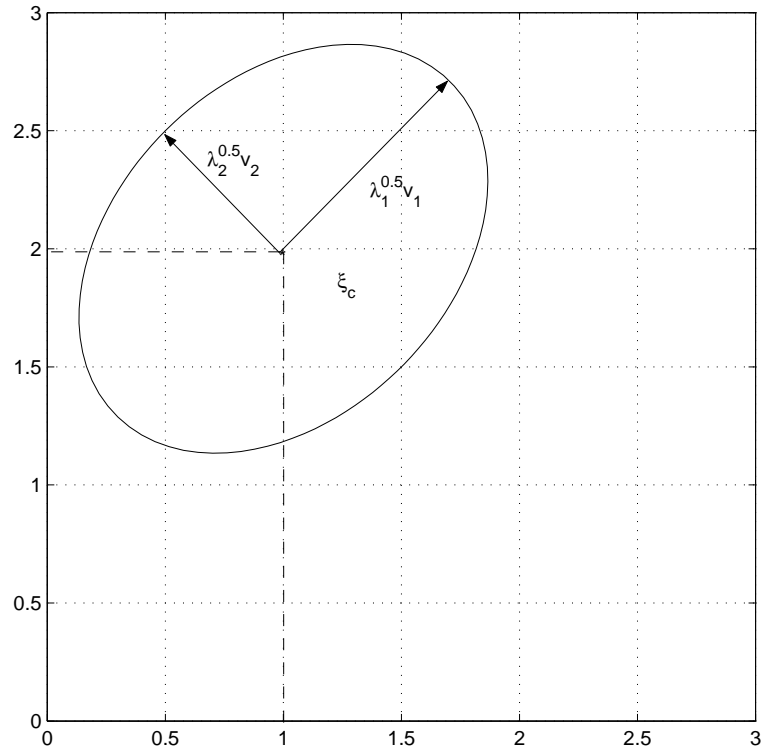
$$P = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

et $x_c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. on a alors $\lambda_1 = 1, v_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}^T, \lambda_2 = \frac{1}{2}, v_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}^T$.

La taille d'un ellipsoïde peut être mesurée par la somme des carrés des longueurs des demi axes, donnée par $\text{trace}(P)$ ou par le volume de l'ellipsoïde :

$$\text{vol}(\varepsilon) = \alpha_n (\det(P))^{\frac{1}{2}}$$

avec α_n le volume de la boule unité de dimension n .

FIG. 7.1 – Exemple d'un ellipsoïde dans \mathbb{R}^2

Représentation alternative Soit C une matrice¹ de rang plein telle que $P = CC^T$. Alors

$$(x - x_c)^T P^{-1} (x - x_c) = (C^{-1}(x - x_c))^T \underbrace{(C^{-1}(x - x_c))}_u \leq 1$$

Par suite,

$$\varepsilon = \{x \in \mathbb{R}^n \mid \exists u \in \mathbb{R}^n, \|u\| \leq 1, \text{ tel que } x = Cu + x_c\}.$$

7.2 Polyèdres

Un hyperplan est défini par : $\{x \mid a^T x = b\}$ où $a \neq 0$ est un vecteur de \mathbb{R}^n . Il définit deux demi plans : $\{x \mid a^T x \geq b\}$ et $\{x \mid a^T x \leq b\}$. Une intersection finie de demi-plans définit un polyèdre :

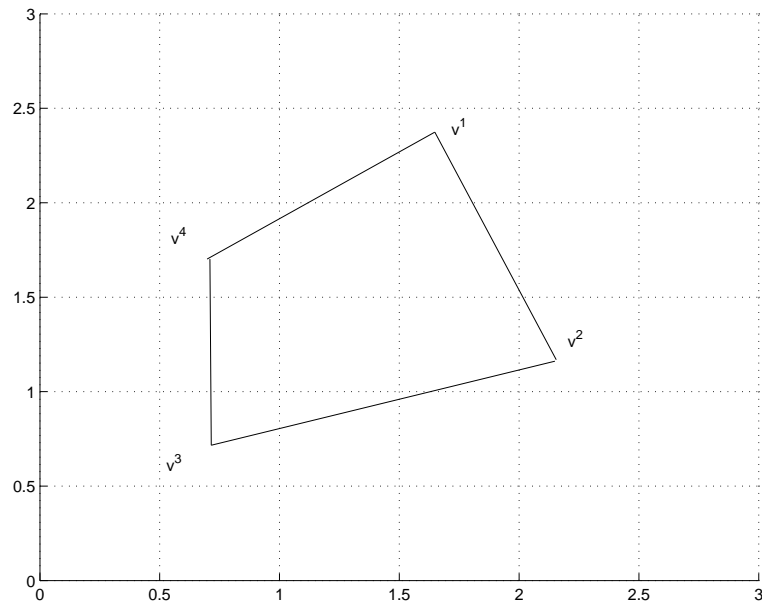
$$\mathcal{P} = \{x \mid a_j^T x \leq b_j, \quad j = 1, \dots, m\}$$

Notation compacte : $\mathcal{P} = \{x \mid Ax \leq b\}$ ATTENTION ICI Ax et b sont des vecteurs. Notation $x > y \Leftrightarrow \forall i, x_i > y_i$.

Un *polytope* est un polyèdre borné. Il peut se réécrire : il existe v^i vecteurs tels que :

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n \mid \exists \lambda_i \geq 0, \quad i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1, \quad x = \sum_{i=1}^k \lambda_i v^i \right\}$$

¹Une telle matrice existe d'après la propriété présentée page 67.

FIG. 7.2 – Exemple d'un polytope dans \mathbb{R}^2

v^i sont appelés sommets du polytope.

Le choix de la représentation est important. Par exemple on prend le cube unité de \mathbb{R}^n :

$$\text{Cube} = \{x \mid |x_i| \leq 1, i = 1, \dots, n\}$$

1. description 1 : $2n$ inégalités linéaires : $e_i^T x \leq 1$ avec $e_i = [0 \ \dots \ 0 \ \pm 1 \ 0 \ \dots \ 0]^T$
2. description 2 : 2^n sommets v^i (nombre fonction exponentielle de n).

Chapitre 8

Annexe C : TD sur l'analyse et la commande par LMI, Matlab et la LMI control toolbox

Les fonctions de Matlab sont organisées par thème. Le noyau Matlab contient les fonctions de base ; les fonctions plus spécialisées sont regroupées dans des boîtes à outils (pour en avoir la liste, taper `help` sous Matlab).

Remarque préliminaire La fonction Matlab : `help nomdefonction` permet d'avoir une aide en ligne sur cette fonction.

La boîte à outils "LMI control toolbox" regroupe deux groupes de fonctions :

1. Le premier groupe rassemble les fonctions permettant la résolution de problèmes d'optimisations sous contraintes LMIs (faisabilité, minimisation d'un coût linéaire sous contraintes LMIs, minimisation de valeur propre généralisée sous contraintes LMIs) :
2. le second groupe rassemble les fonctions résolvant des problèmes d'Automatique par l'utilisation de l'optimisation sous contraintes LMI (par exemple, calcul d'un correcteur H_∞).

Nous allons découvrir le premier groupe de fonctions à travers la programmation de la norme H_∞ d'une fonction de transfert. L'utilisation de l'optimisation LMI pour la μ analyse sera ensuite évoquée.

8.1 Calcul de la norme H_∞ d'une fonction de transfert

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

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bw(t) \\ z(t) &= Cx(t) + Dw(t)\end{aligned}\tag{8.1}$$

où $x(t)$, de dimension n , est l'état du système, $w(t)$ l'entrée et $z(t)$ est la sortie. La résolution du problème d'optimisation suivant permet d'obtenir le carré de la norme H_∞ de H :

$$\min_{\eta \in \mathbb{R}, \underline{P} = \underline{P}^T \in \mathbb{R}^{n \times n}} \underline{\eta} \quad \left\{ \begin{array}{l} \underline{P} \geq 0 \\ \left[\begin{array}{cc} A^T \underline{P} + \underline{P} A + C^T C & \underline{P} B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \underline{\eta} I \end{array} \right] \leq 0 \end{array} \right. \quad (8.2)$$

Il s'agit d'un problème de minimisation d'un coût linéaire sous contraintes LMIs. Les variables de décision sont $\underline{\eta}$ et \underline{P} . La fonction `mincx` permet de résoudre un tel problème. Exemple :

```
[copt,xopt] = mincx(lmisys, c)
```

Entrées la première entrée `lmisys` est une variable qui décrit l'ensemble des variables de décision et l'ensemble des contraintes LMIs, la seconde `c` est une variable qui décrit la fonction de coût.

Sorties La première sortie `copt` donne la valeur optimale du coût, la seconde `xopt` les valeurs des variables de décision pour lequel ce coût est atteint.

Comment définir la variable décrivant les contraintes LMIs? La commande Matlab `lmiedit` ouvre une interface utilisateur qui va permettre de la définir simplement.

```
>> lmiedit
```

La fenêtre représentée figure 8.1 s'ouvre alors. La figure se décompose en trois zones. Dans la première ("name the LMI") est entré le nom de la variable qui va contenir la description des variables et des contraintes LMI. Les variables sont introduites ligne par ligne. La première colonne contient le nom de la variable, la seconde sa structure (**S** pour matrice symétrique, **R** pour matrice rectangulaire et **G** pour autre) et la dernière colonne permet de définir ses dimensions (par exemple dans le cas d'une matrice symétrique `[n,1]` définit une matrice symétrique de dimension n par n).

La seconde zone ("describe the matrix variables") va contenir la description des variables d'optimisation. La troisième zone contient alors la description des contraintes LMIs. Par exemple, une condition nécessaire et suffisante de stabilité pour le système (8.1) est donnée par l'existence d'une matrice symétrique \underline{P} telle que

$$\left\{ \begin{array}{l} \underline{P} > 0 \\ A^T \underline{P} + \underline{P} A < 0 \end{array} \right.$$

La figure 8.2 illustre comment cet ensemble de variables et de contraintes est défini sous `lmiedit`. Cette définition peut être alors sauvegardée sous forme d'un fichier `.m` à l'aide du bouton `write` de `lmiedit`. On obtient alors le fichier suivant :

```
setlmis([]);
P=lmivar(1,[n,1]);
lmiterm([-1 1 1 P],1,1); % LMI #1: P
lmiterm([2 1 1 P],A',1,'s'); % LMI #2: A'*P+P*A
stab=getlmis;
```

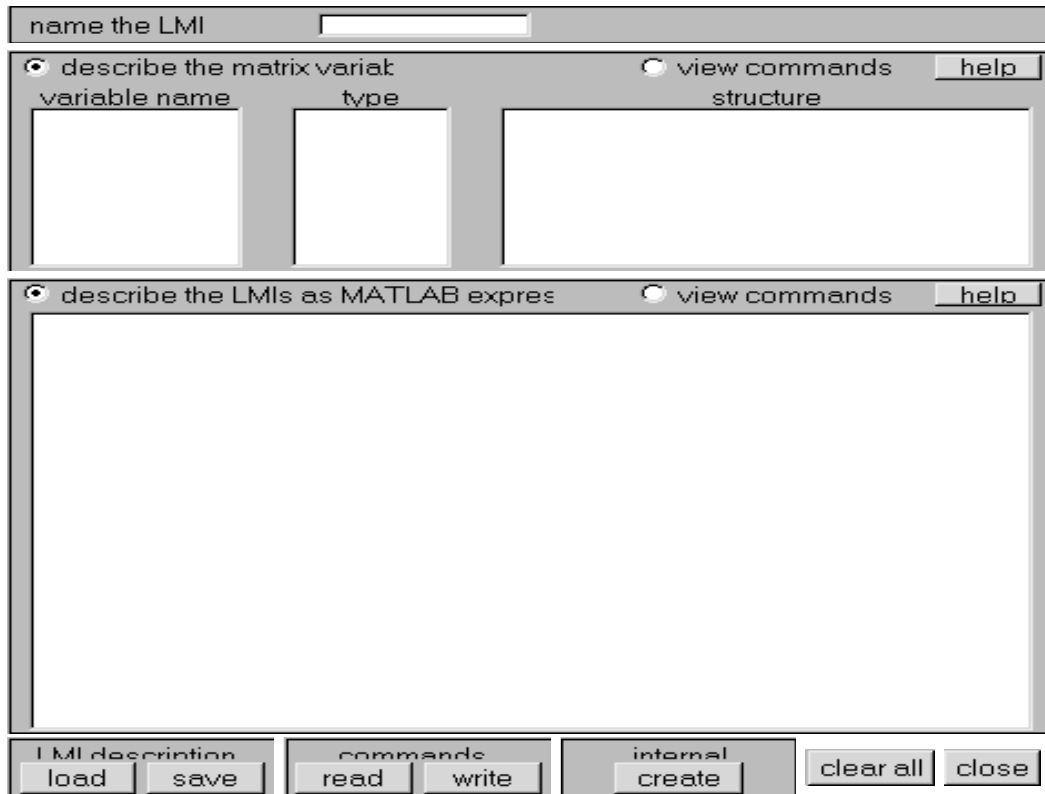



FIG. 8.1 – Fenêtre LMEdit

Dans la LMI control toolbox, chaque LMI est de la forme :

$$\underbrace{L(\xi)}_{\text{termedegauche}} < \underbrace{R(\xi)}_{\text{termededroite}} .$$

Par défaut, un terme vaut 0. Chaque terme peut être découpé en sous blocs en fonction du besoin considéré. Par exemple,

$$L(\xi) = \begin{bmatrix} L(\xi)_{11} & L(\xi)_{12} \\ L(\xi)_{21} & L(\xi)_{22} \end{bmatrix}$$

`setlmis([])` indique que l'on va créer un nouvelle ensemble de contraintes et de variables ;

`P=lmivar(1,[n,1])` définit la variable de décision P qui est symétrique (premier argument de `lmivar` à 1) de dimension n par n ; la fonction `lmivar` peut admettre trois arguments en sortie : `[P,ndec,Pstruct]=lmivar(1,[n,1])` où `ndec` indique le nombre de variables de décision définies jusqu'à maintenant (taille du vecteur ξ) et où `Pstruct` indique le lien entre la variable de décision matricielle \underline{P} et les différentes composantes du vecteur de variable de décision ξ . Par exemple, dans le cas où $n=2$, on a :

```
>> Pstruct
Pstruct =
     1     2
     2     3
```

l'élément (1,1) de \underline{P} correspond au premier élément du vecteur de variables de décision, etc..

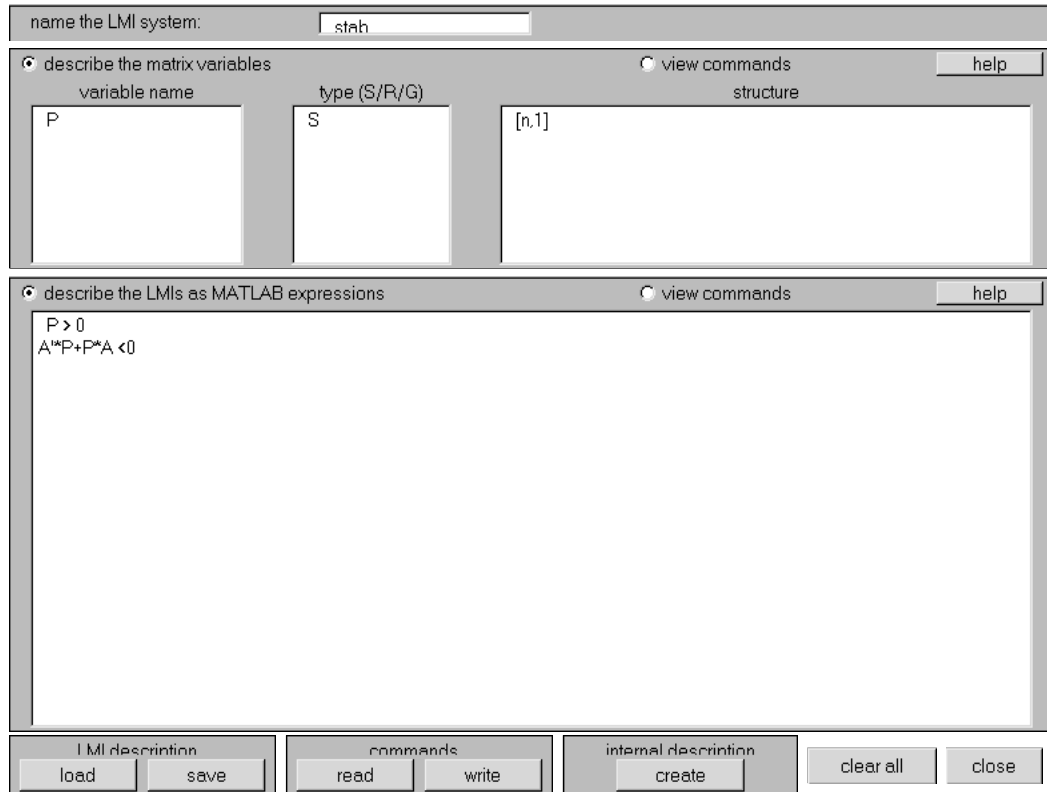


FIG. 8.2 – Fenêtre LMEdit

`lmiterm([-1 1 1 P],1,1)` définit (une partie du) sous bloc de coordonnées (1,1) (deuxième et troisième éléments de $[-1 \ 1 \ 1 \ P]$) du terme de droite (premier élément de $[-1 \ 1 \ 1 \ P]$ négatif) de la première contrainte LMI (premier élément de $[-1 \ 1 \ 1 \ P]$ dont la valeur absolue vaut 1); ce terme s'écrit $I \times P \times I$ (le premier I correspond au second argument de la commande `lmiterm([-1 1 1 P],1,1)`, la variable P est définie par le quatrième élément du vecteur $[-1 \ 1 \ 1 \ P]$, le second au troisième argument);

`lmiterm([2 1 1 P],A',1,'s')` définit (une partie du) sous bloc de coordonnées (1,1) (deuxième et troisième éléments de $[2 \ 1 \ 1 \ P]$) du terme de gauche (premier élément de $[2 \ 1 \ 1 \ P]$ positif) de la seconde contrainte LMI (premier élément de $[2 \ 1 \ 1 \ P]$ dont la valeur absolue vaut 2); ce terme s'écrit $A' \times P \times I$ (le premier A' correspond au second argument de la commande `lmiterm([2 1 1 P],A',1,'s')`, la variable P est définie par le quatrième élément du vecteur $[2 \ 1 \ 1 \ P]$, la matrice I au troisième argument de la commande). Le quatrième argument 's' indique que le sous bloc contient aussi le symétrique de $A' \times P \times I$ soit $I \times P \times A$.

Exercice En utilisant `lmiedit`, générer un fichier `.m` permettant de définir l'ensemble des variables et des contraintes associées au problème d'optimisation LMI (8.2) sous le nom de `lmisys`. On prendra soin de définir en premier `eta` afin qu'il apparaisse en premier dans le vecteur de variables de décision.

Par suite, le vecteur c définissant la fonction de coût est défini par :

```
c = zeros((n*(n+1)/2+1),1); c(1) = 1;
```

Compléter le fichier précédemment obtenu de cet ligne d'instruction ainsi que de la ligne :

```
[copt,xopt] = mincx(lmisys, c)
```

En prenant la racine carrée de `copt` (fonction `sqrt`) la norme H_∞ est alors obtenue. Le programme va être testé sur le calcul de la norme H_∞ d'un système du second ordre :

$$H(p) = \frac{\omega_0^2}{p^2 + 2\xi_0\omega_0p + \omega_0^2}$$

défini dans le fichier `sys_hinf.m` :

```
w0 = 10;  xsi0 = 0.01;
sys      = ltisys('tf', w0^2, [1, 2*xsi0*w0, w0^2]);
[A,B,C,D] = ltiss(sys);
n = 2;
```

Tester le programme ainsi obtenu.

Le vecteur `xopt` contient la valeur du vecteur de variables de décision ξ pour lequel le minimum est atteint. Il est plus important de connaître la valeur des variables de décision sous forme matricielle. Pour cela, on dispose de la fonction `dec2mat` : la ligne de commande

```
Popt = dec2mat(lmisys,xopt,P);
```

permet d'obtenir la valeur de la variable \underline{P} pour laquelle le minimum est atteint. On peut vérifier que la solution optimale est correcte c'est-à-dire que `Popt` a bien ses valeurs propres positives

```
>> eig(Popt)
```

et qu'avec $\underline{P} = \text{Popt}$ on a :

$$\begin{bmatrix} A^T \underline{P} + \underline{P} A + C^T C & \underline{P} B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \underline{\eta} I \end{bmatrix} \leq 0$$

Pour cela, la ligne de commande

```
>> evalsys = evallmi(lmisys, xopt);
```

permet d'évaluer les matrices définissant les contraintes pour $\xi = \text{xopt}$. La ligne de commande

```
>> [lhs,rhs] = showlmi(evalsys, 2);
```

permet alors d'évaluer les deux termes de la seconde LMI. On vérifie ensuite que

```
>> eig(lhs)
```

Paramètres d'optimisation La fonction `mincx` peut admettre un troisième argument d'entrée `options` qui est un vecteur de 5 éléments :

`options(1)` fixe la précision relative sur le coût optimal (par défaut, c'est 10^{-3});

`options(2)` fixe le nombre maximum d'itérations (par défaut, c'est 100);

`options(3)` fixe le rayon R de faisabilité (par défaut, c'est 10^9) : dans l'algorithme, la norme du vecteur de variables de décision est bornée par R : $\xi^T \xi \leq R^2$. Une valeur de R trop faible peut sur contraindre le problème et donc modifier la solution, une valeur de R trop élevée peut mener à des problèmes numériques.

`options(4)` fixe le nombre maximum d'itérations au bout duquel si la fonction de coût n'a pas décréu d'une valeur supérieure à la précision relative considérée le programme est arrêté (par défaut, c'est 10);

`options(5) = 1` supprime les affichages dans la fenêtre `Matlab` lors de l'exécution de `mincx`, `=0` les rétablit (par défaut, c'est 0).

Si un élément de `options` est à 0 alors la valeur par défaut est sélectionnée.

Jouer sur la première option pour obtenir une meilleure précision.

Efficacité par rapport à une approche directe Comparer le résultat obtenu avec un examen direct des valeurs singulières de $H(j\omega)$ en fonction de la pulsation ω (voir figure 8.3).

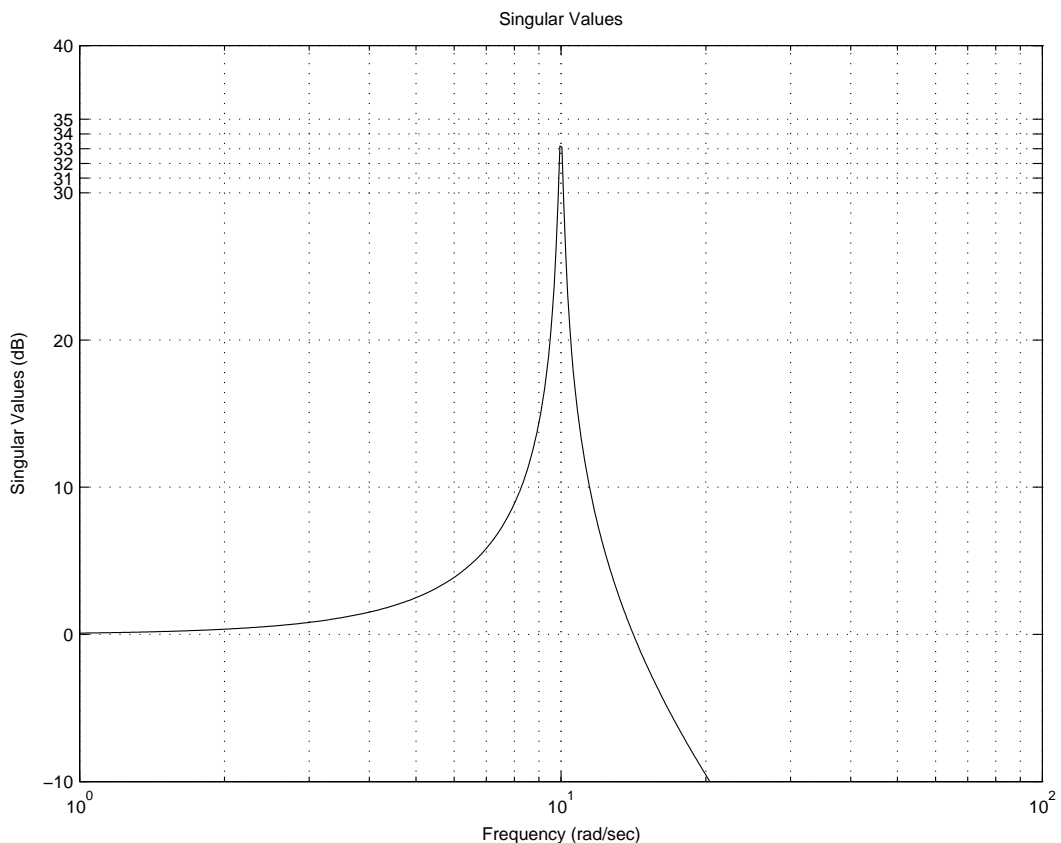
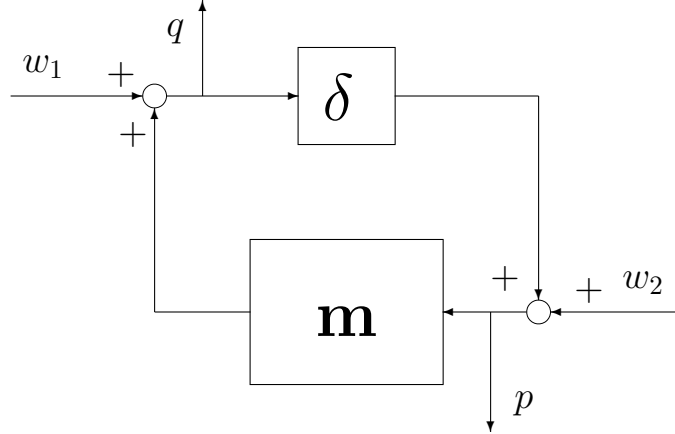


FIG. 8.3 – Valeurs singulières de $H(p)$ `splot(sys, 'sv', logspace(0, 2, 500))`

8.2 LMI et μ -analyse : des critères graphiques à l'optimisation

FIG. 8.4 – Connexion de M avec Δ

Soit m une fonction de transfert stable à une entrée et à une sortie. On désire vérifier que le système bouclé représenté figure 8.4 est stable pour tout scalaire $\delta \in [-1, 1]^1$. D'après le théorème du petit gain structuré, une condition nécessaire et suffisante est donnée par :

$$\forall \omega, \quad \mu_{\Delta}(m(j\omega)) < 1.$$

(Dans ce qui suit, on considère une pulsation ω donnée.) On dispose d'une borne supérieure $\mu_{sup}(m(j\omega))$ telle que

$$\mu_{\Delta}(m(j\omega)) \leq \mu^{sup}(m(j\omega))$$

où, dans le cas particulier considéré, la borne supérieure de μ est obtenue par la résolution du problème d'optimisation suivant :

$$\begin{aligned} \mu^{sup}(M(j\omega)) = \min \quad & \underline{d} \in \mathbb{R}^+ \quad \beta \\ & \underline{g} \in \mathbb{R} \\ & \beta \in \mathbb{R}^+ \\ & \underline{d}m(j\omega)^*m(j\omega) + j(\underline{g}m(j\omega) - \underline{g}m(j\omega)^*) - \beta^2\underline{d} < 0 \end{aligned}$$

Dans ce cas particulier, la borne supérieure de μ donne la valeur de μ . Ce problème est un problème de minimisation de valeur propre généralisée sous contraintes LMIs.

On garantira la stabilité de la boucle fermée pour tout scalaire $\delta \in [-1, 1]$ si et seulement s'il existe $\underline{d} \in \mathbb{R}^+$ et $\underline{g} \in \mathbb{R}$ tels que :

$$\underline{d}m(j\omega)^*m(j\omega) + j\underline{g}(m(j\omega) - m(j\omega)^*) - \underline{d} < 0 \quad (8.3)$$

ce qui est un problème de faisabilité LMI.

¹Il s'agit en fait d'un problème proche de celui de la marge de gain.

Essayons de donner une interprétation graphique à la condition (8.3). Comme $\underline{d} > 0$, elle peut se réécrire en posant $\underline{g} = \frac{g}{\underline{d}}$: trouver $\underline{g} \in \mathbb{R}$ tel que :

$$m(j\omega)^* m(j\omega) + j\underline{g}(m(j\omega) - m(j\omega)^*) - 1 < 0. \quad (8.4)$$

En complétant les carrés, on obtient

$$(m(j\omega) - j\underline{g})^* (m(j\omega) - j\underline{g}) < 1 + \underline{g}^2.$$

Dans le plan complexe, cette condition revient à vérifier que le point associé à $m(j\omega)$ est dans le disque ouvert de centre $j\underline{g}$ et de rayon $\sqrt{1 + \underline{g}^2}$. Ce disque est délimité par le cercle de centre $j\underline{g}$ et passant par les points -1 et 1 (voir la figure 8.5).

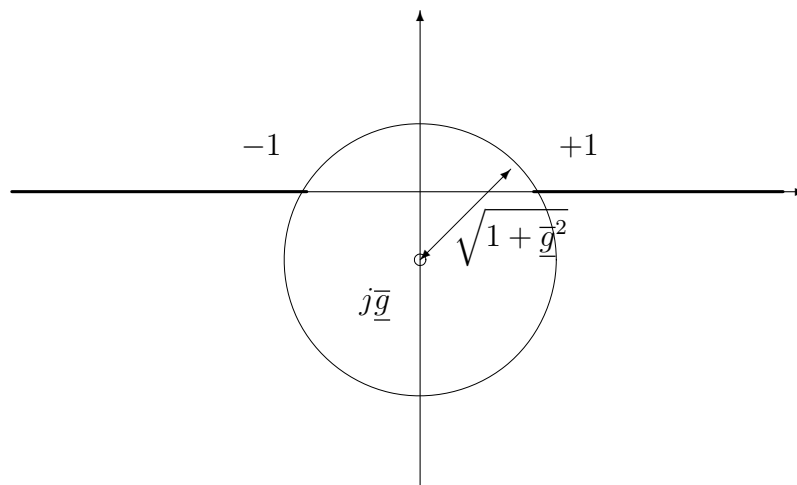


FIG. 8.5 – Disque de centre $j\underline{g}$ et de rayon $\sqrt{1 + \underline{g}^2}$ avec \underline{g} négatif

Interprétons cette condition par rapport au critère de Nyquist. La fonction de transfert $m(j\omega)$ étant stable, il suffit de vérifier que pour toute valeur de $\delta \in [-1, 1]$, le tracé de $m(j\omega)$ dans le plan complexe pour ω allant de $-\infty$ à ∞ n'entoure pas le point $\frac{1}{\delta}$. Pour cela, en utilisant un argument de continuité, une condition nécessaire et suffisante est d'assurer que pour toute pulsation ω et pour tout $\delta \in [-1, 1]$, le point $m(j\omega)$ ne recouvre pas le point $\frac{1}{\delta}$.

Quand δ décrit l'intervalle $[-1, 1]$, le point $\frac{1}{\delta}$ décrit le demi axe $[-\infty, -1]$ puis le demi axe $[1, +\infty]$, demi axes qui sont extérieurs au disque ouvert de centre $j\underline{g}$ et de rayon $\sqrt{1 + \underline{g}^2}$. La condition (8.3) assurant que le point $m(j\omega)$ est dans ce disque, on en déduit que pour la pulsation ω , le point $m(j\omega) \neq \frac{1}{\delta}$ pour tout $\delta \in [-1, 1]$.

Pour une pulsation ω donnée, le programme **Matlab** `mu_int.m` permet de construire un tel disque. La résolution du problème (8.4) correspond à un problème de faisabilité qui peut être codé sous la LMI control toolbox de la façon suivante (extrait du programme `mu_inter.m`) :

```
% Calcul de la reponse frequentielle a la pulsation w
Mw = sresp(sysm, w);

% Construction des matrices augmentees equivalentes aux matrices complexes
Mw_aug = [ real(Mw), imag(Mw);
           -imag(Mw), real(Mw)];

jMw_aug = [-imag(Mw), real(Mw);
           -real(Mw), -imag(Mw)];

% définition du problème LMI
setlmis([]);
g = lmivar(1, [1,1]);

% contrainte  $Mw' * Mw + jg(Mw - Mw^*) - 1 < 0$ 
lmiterm([1 1 1 0], Mw_aug', Mw_aug - 1);
lmiterm([1 1 1 g], 1, jMw_aug, 's');

Stab = getlmis;

options = zeros(1,5);
options(3) = 4;      % limitation de la norme de la variable d'optimisation

% calcul de g
[tmin, xfeas] = feasp(Stab, options);
gfeas = xfeas;
```

En guise de conclusion

Le critère de Nyquist tient une place centrale en Automatique fréquentielle classique dans l'étude de la robustesse des systèmes en boucle fermée mono variables. Son grand intérêt est lié à son utilisation sous forme de critères graphiques. Néanmoins, dans le cas des systèmes multivariables ou même pour l'étude de marges d'incertitudes plus complexes que la marge de gain ou la marge de phase, même si c'est le critère de Nyquist qui est utilisé, il ne peut plus, en général, être mis en œuvre sous forme de critères graphiques. Les critères graphiques sont alors remplacés par la résolution de problèmes d'optimisation sous contraintes LMIs. De ce fait, le rôle de l'optimisation LMI dans l'automatique fréquentielle avancée est similaire à celui (fondamental) des critères graphiques dans l'automatique fréquentielle classique.

Bibliographie

- [1] V. Blondel and J. N. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM J. of Control and Opt.*, 35(6), 1997.
- [2] S. Boyd and L. El Ghaoui. Method of centers for minimizing generalized eigenvalues. *Linear Algebra and Applications, special issue on Numerical Linear Algebra Methods in Control, Signals and Systems*, 188 :63–111, July 1993.
- [3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in Systems and Control Theory*, volume 15 of *Studies in Appl. Math.* SIAM, Philadelphia, June 1994.
- [4] S. Boyd and L. Vandenberghe. Introduction to convex optimization with engineering applications. Lecture Notes, Information Systems Laboratory. Stanford University, 1995.
- [5] R. D. Braatz, P. M. Young, J. C. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Trans. Aut. Control*, AC-39(5) :1000–1002, May 1994.
- [6] M. Chilali. *Méthodes LMI pour l'analyse et la synthèse multi-critère*. PhD thesis, Université Paris Dauphine, Paris, France, 1996.
- [7] Y-S. Chou. *Stability Robustness of Systems under Structured Time-Varying Mixed Uncertainty*. PhD thesis, University of Maryland, Colledge Park, US, 1996.
- [8] C. A. Desoer and M. Vidyasagar. *Feedback Systems : Input-Output Properties*. Academic Press, New York, 1975.
- [9] J. Doyle, A. Packard, and K. Zhou. Review of LFT's, LMI's and μ . In IEEE, editor, *Proc. IEEE Conf. on Decision and Control*, volume 2, pages 1227–1232, Brighton, England, December 1991.
- [10] J. C. Doyle. A review of μ for case studies in robust control. In *IFAC Munich*, pages 395–402, 1987.
- [11] J.C. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proc.*, 129-D(6) :242–250, November 1982.
- [12] J.C. Doyle and G. Stein. Multivariable feedback design : Concepts for a classical/modern synthesis. *IEEE Trans. Aut. Control*, AC-26(1) :4–16, February 1981.
- [13] L. El Ghaoui and J.P. Folcher. Multiobjective robust control of LTI systems subject to unstructured perturbations. *Syst. Control Letters*, 28 :23–30, 1996.
- [14] L. El Ghaoui, R. Nikoukhah, and F. Delebecque. *LMITool : A front-end for LMI optimization, users's guide*, February 1995. Available via anonymous ftp to <ftp.ensta.fr>, under </pub/elghaoui/lmitool>.
- [15] L. El Ghaoui and G. Scorletti. Control of rational systems using Linear-Fractional Representations and Linear Matrix Inequalities. *Automatica*, 32(9) :1273–1284, September 1996.

- [16] M. K. H. Fan, A. L. Tits, and J. C. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Aut. Control*, 36(1) :25–38, January 1991.
- [17] G. Ferreres, V. Fromion, G. Duc, and M. M'Saad. Non-conservative robustness evaluation of a multivariable H_∞ missile autopilot. *Int. J. Robust and Nonlinear Control*, 6(8) :743–769, June 1996.
- [18] G. Ferreres, G. Scorletti, and V. Fromion. Advanced computation of the robustness margin. In *Proc. IEEE Conf. on Decision and Control*, pages 4580–4584, December 1996.
- [19] S. Font. *Méthodologie pour prendre en compte la robustesse des systèmes asservis : optimisation H_∞ et approche symbolique de la forme standard*. PhD thesis, Université Paris XI Orsay, France, 1995.
- [20] V. Fromion. *Une approche incrémentale de la robustesse non linéaire ; application au domaine de l'aéronautique*. PhD thesis, Université de Paris Sud Orsay, January 1995.
- [21] V. Fromion, G. Scorletti, and G. Ferreres. Nonlinear performance of a PID controlled missile : a simple explanation. *Int. J. Robust and Nonlinear Control*, (9) :485–518, 1999.
- [22] V. Fromion, G. Scorletti, and G. Ferreres. Nonlinear performance of a PID controlled missile : a simple explanation. *Int. J. Robust and Nonlinear Control*, (9) :485–518, 1999.
- [23] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4 :421–448, 1994.
- [24] P. Gahinet, A. Nemirovsky, A. L. Laub, and M. Chilali. *LMI Control Toolbox*. The Mathworks Inc., 1995.
- [25] K.C. Goh and M.G. Safonov. Robust analysis, sectors and quadratic functionals. In IEEE, editor, *Proc. IEEE Conf. on Decision and Control*, New Orleans, Louisiana, 1995.
- [26] R. A. Horn and C. A. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [27] J.H. Ly, R.Y. Chiang, K.C. Goh, and M.G. Safonov. Multiplier K_m/μ -analysis-LMI approach. In *Proc. American Control Conf.*, pages 431–436, June 1995.
- [28] J.H. Ly, M.G. Safonov, and R.Y. Chiang. Real / complex multivariable stability margin computation via generalized popov multiplier - LMI approach. In *Proc. American Control Conf.*, 1994.
- [29] A. Nemirovskii. Several NP-hard problems arising in robust stability analysis. *MCSS*, 6 :99–105, 1993.
- [30] Yu. Nesterov and A. Nemirovsky. *Interior point polynomial methods in convex programming : Theory and applications*. SIAM, Philadelphia, PA, 1994.
- [31] A. Packard. Gain scheduling via linear fractional transformations. *Syst. Control Letters*, 22(2) :79–92, February 1994.
- [32] A. Packard and J.C. Doyle. The complex structured singular value. *Automatica*, 29(1) :71–109, 1993.
- [33] A. Packard and P. Pandey. Continuity properties of the real/complex structured singular value. *IEEE Trans. Aut. Control*, 38(3) :415–428, 1993.
- [34] V. M. Popov. *Hyperstability of Control Systems*. Springer-Verlag, New York, 1973.
- [35] A. Rantzer and A. Megretski. System analysis via integral quadratic constraints. In *Proc. IEEE Conf. on Decision and Control*, pages 3062–3067, December 1994.
- [36] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, Princeton, second edition, 1970.

- [37] M. Safonov and T.-C. Tsao. *Feedback Control, Nonlinear Systems and Complexity*, chapter The Unfalsified Control Concept : A Direct Path from Experiment to Controller, pages 196–214. B. A. Francis and A. R. Tannenbaum (editors), Springer-Verlag : Berlin, 1995.
- [38] M. G. Safonov. *Stability and Robustness of Multivariable Feedback Systems*. MIT Press, Cambridge, 1980.
- [39] M. G. Safonov. Stability margin of diagonally perturbed multivariable feedback systems. *IEE Proc., Part D*, 129(6) :251–256, 1982.
- [40] M. G. Safonov and M. Athans. A multiloop generalization of the circle criterion for stability margin analysis. *IEEE Trans. Aut. Control*, AC-26(2) :415–422, April 1981.
- [41] G. Scorletti. *Approche Unifiée de l'analyse et la commande des systèmes par formulation LMI*. PhD thesis, Université d'Orsay, Paris, France, 1997. In French.
- [42] G. Scorletti. Robustness analysis with time delays. In IEEE, editor, *Proc. IEEE Conf. on Decision and Control*, pages 3824–3829, San Diego, California, December 1997.
- [43] G. Scorletti and L. El Ghaoui. Improved LMI conditions for gain scheduling and related problems. *Int. J. Robust and Nonlinear Control*, 8(10) :845–877, August 1998.
- [44] A.L. Tits, V. Balakrishnan, and L. Lee. Robustness under bounded uncertainty with phase information. *IEEE Trans. Aut. Control*, 44(1), 1999.
- [45] O. Toker and H. Ozbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *American Control Conference*, pages 2525–2526, June 1995.
- [46] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1) :49–95, March 1996.
- [47] J. C. Willems. *The Analysis of Feedback Systems*, volume 62 of *Research Monographs*. MIT Press, Cambridge, Massachusetts, 1969.
- [48] J. C. Willems. Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Trans. Aut. Control*, AC-16(6) :621–634, december 1971.
- [49] V. A. Yakubovich. A frequency theorem in control theory. *Siberian Mathematical Journal*, 14(2) :265–289, March–April 1973.
- [50] G. Zames. On the input-output stability of time-varying nonlinear feedback systems—Part I : Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Trans. Aut. Control*, AC-11 :228–238, April 1966.
- [51] G. Zames. On the input-output stability of time-varying nonlinear feedback systems—Part II : Conditions involving circles in the frequency plane and sector nonlinearities. *IEEE Trans. Aut. Control*, AC-11 :465–476, July 1966.
- [52] G. Zames. Feedback and optimal sensitivity : Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans. Aut. Control*, AC-26(2) :301–320, April 1981.