



HAL
open science

Optimisation et Sciences de l'ingénieur : une approche basée sur l'Automatique (Notes de cours)

Gérard Scorletti

► To cite this version:

Gérard Scorletti. Optimisation et Sciences de l'ingénieur : une approche basée sur l'Automatique (Notes de cours). DEA. Université de Caen/Basse-Normandie ENSICAEN, France. 2012, pp.81. cel-00673925

HAL Id: cel-00673925

<https://cel.hal.science/cel-00673925v1>

Submitted on 24 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ de CAEN/BASSE-NORMANDIE
U.F.R. : SCIENCES CAEN

**Optimisation et Sciences de l'ingénieur : une approche
basée sur l'Automatique (Notes de cours)**

Master Recherche ESCI 2A

Université de Caen Basse-Normandie

Gérard Scorletti

Ecole Centrale de Lyon
Département EEA - Laboratoire Ampère
bât. H9, 36 avenue Guy de Collongue
69134 Ecully Cedex
e-mail : gerard.scorletti@ec-lyon.fr
tel : 04 72 18 64 55

©Gérard Scorletti, France, 2008

[http ://www.greyc.ensicaen.fr/~scorlett/Gerard_S/DEA_opti.html](http://www.greyc.ensicaen.fr/~scorlett/Gerard_S/DEA_opti.html)

23 janvier 2012

“The control theoreticians role may be viewed as one of developing methods that allow the control engineer to make assumptions which seem relatively natural and physically motivated. The ultimate question of the applicability of any mathematical technique to a specific physical problem will always require a “leap of faith” on the part of the engineer and the theoritician can only hope to make this leap smaller”, John Doyle [?].

“Principle (Simple Case First) Consider first only the very simplest problem - but strive for a representation of the simplest problem which generalizes easily”, Safonov [?]

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	25
3.1	Problèmes d'optimisation sous contraintes LMI	25
3.1.1	Problème de Faisabilité	26
3.1.2	Problème de minimisation d'une fonction de coût linéaire	30
3.1.3	Minimisation de la valeur propre généralisée maximale	32
3.2	Problèmes d'optimisation LMI particuliers	34
3.3	Exercice : Optimisation sur des variables de décisions "discrètes"	35
3.4	Notions sur la résolution de problèmes sous contraintes LMI	36
3.4.1	Minimisation d'un coût linéaire sous contraintes LMI	36
3.4.2	Recherche d'un point faisable	38
3.5	Au delà des contraintes LMIs : les contraintes BMIs	38
3.6	A la recherche de la LMI cachée...	39
3.6.1	Règles de transformation si $F(\xi)$ n'est pas une fonction affine en ξ	40
3.6.2	Règle de transformation quand $x \in \mathcal{X}(\xi)$ avec $\mathcal{X}(\xi) \subset \mathbb{R}^n$ et $\mathcal{X}(\xi) \neq \mathbb{R}^n$	43
3.6.3	Exercices	46
3.6.3.1	Mise sous forme de problèmes d'optimisation LMI	46
3.6.3.2	Commande par placement de pôles	46
3.6.3.3	Commande par précompensateur	48
4	Formulation de problèmes sous forme de problèmes d'optimisation LMI	51
4.1	Des moindres carrés (exemple)	51
4.2	Réglage d'un correcteur Proportionnel Dérivé sur un moteur à courant continu (exemple)	57
4.3	Retour de sortie statique ou la difficulté de chercher des lois de commande simples (exercice)	59
4.4	Retour d'état avec coût quadratique garanti pour une condition initiale donnée (exercice)	59
4.5	Simulation temporelle des systèmes (exercice)	60

5	Annexe A : rappels sur les matrices	63
5.1	Rappels sur les matrices	63
5.1.1	Cas des matrices carrées et réelles	66
6	Annexe B : Ensembles convexes particuliers	69
6.1	Ellipsoïdes	69
6.2	Polyèdres	70
7	Annexe C : Programmes de la section 4.1	73
7.1	Programme principal	73
7.2	Moindres carrés par LMI	75
7.3	Moindres carrés avec positivité de la solution par LMI	75
7.4	Moindres carrés, approche polytopique	76
7.5	Moindres carrés, approche alternative	78
	Bibliographie	81

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'algorithmes sur des calculateurs. 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

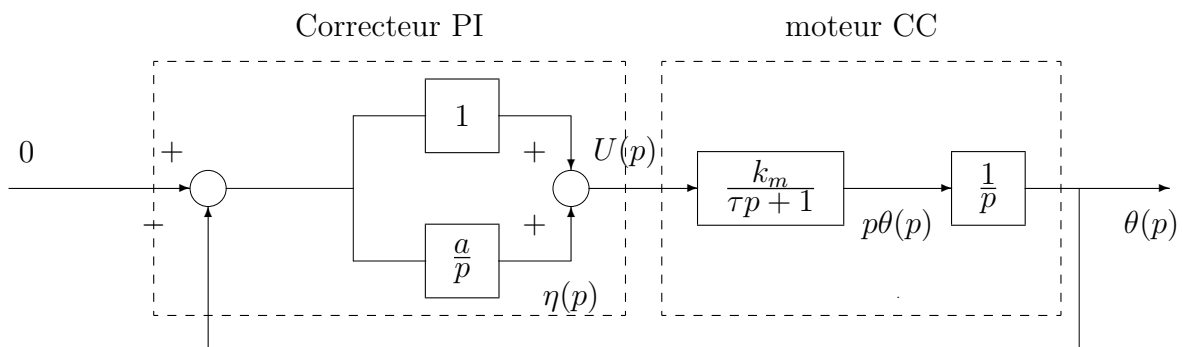


FIG. 1.1 – Commande d'un moteur à courant continu par un correcteur PI

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 $t = 0$.

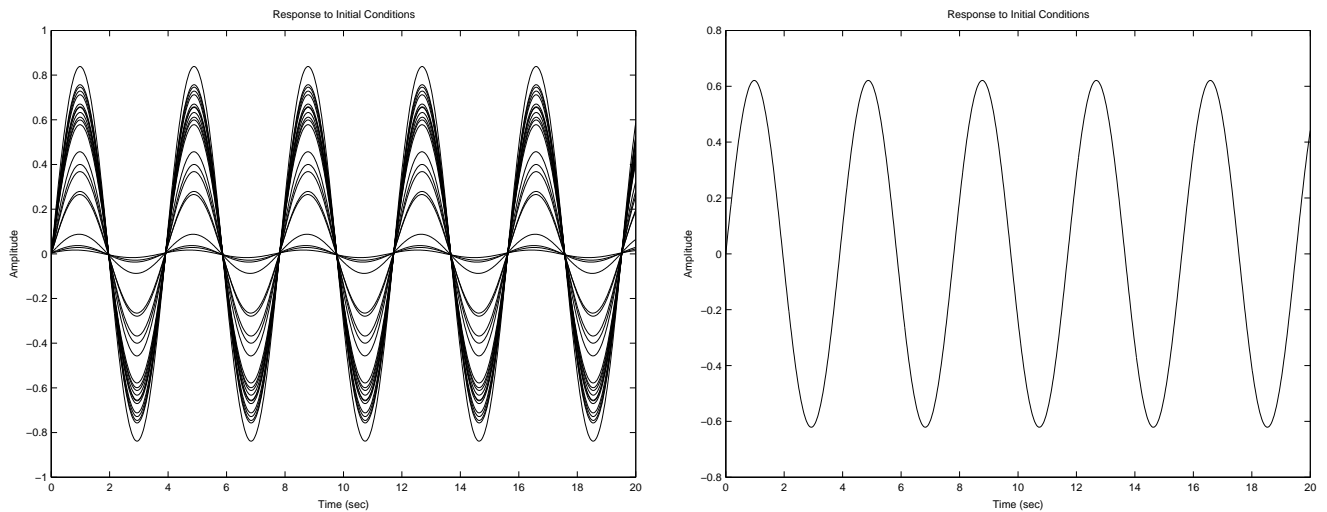


FIG. 1.2 – Simulation du moteur CC avec correcteur PI

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

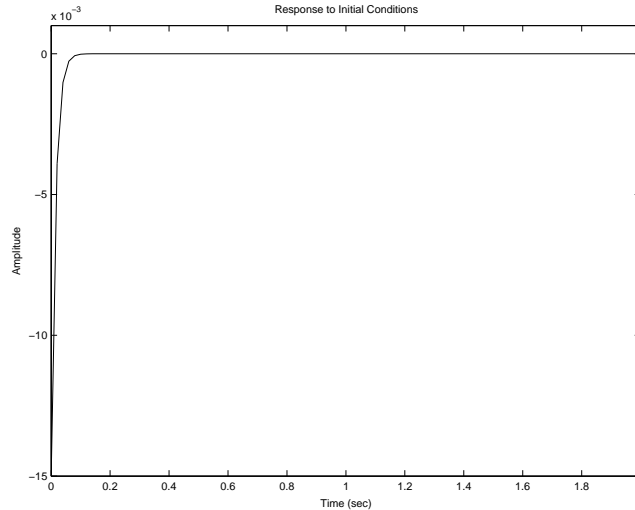


FIG. 1.3 – Simulation du moteur CC avec correcteur PI avec $\theta(0) = -0.0149$ radian, $\dot{\theta}(0) = 1$ radian/s et $\eta(0) = -2.458 \times 10^{-6}$

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 5 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 ?

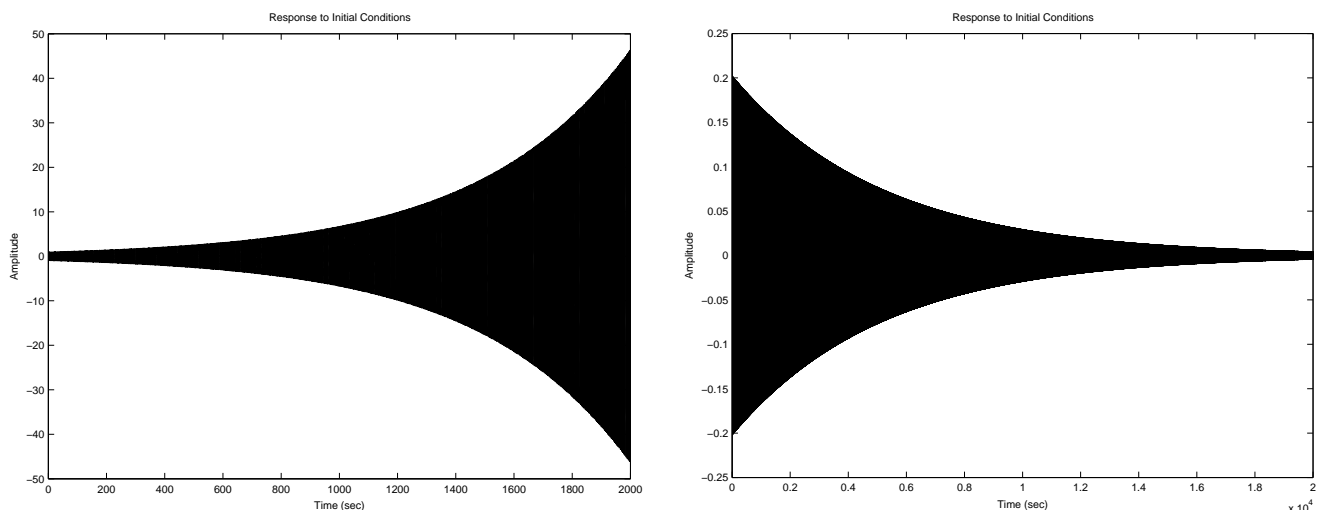


FIG. 1.4 – Simulation du moteur CC avec correcteur PI

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 5 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 :

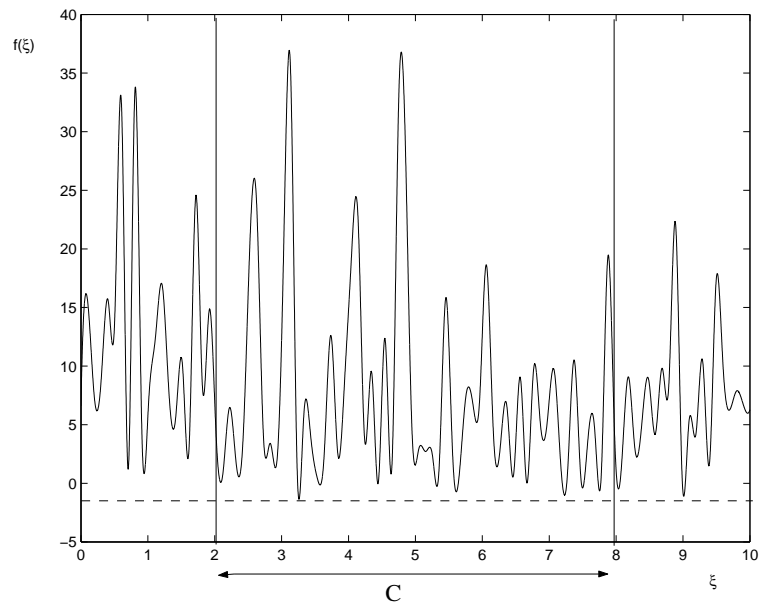
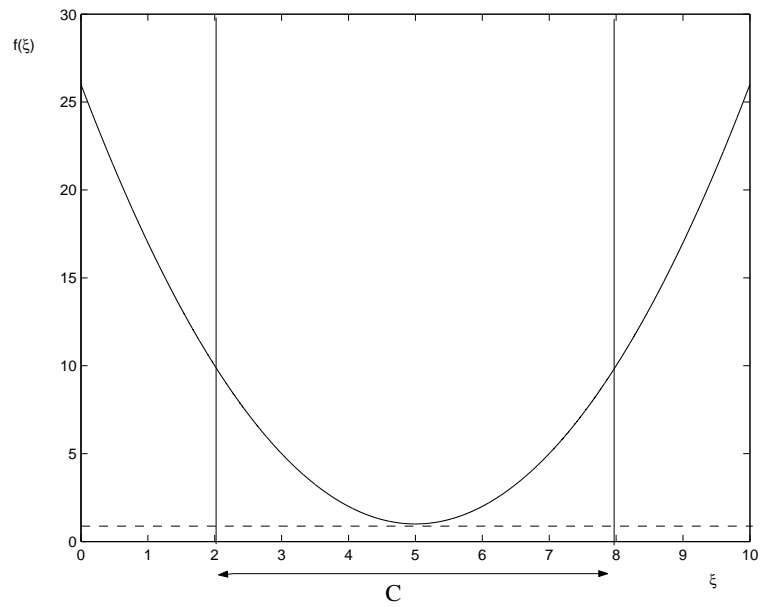
$$\xi^* = \operatorname{argmin}_{\xi \in \mathcal{C}} f(\xi)$$

- 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

On dit que :

- $\xi^* \in \mathcal{C}$ est un minimum local s'il existe un voisinage ouvert \mathbb{U} de ξ^* tel que, pour tout $\xi \in \mathbb{U} \cap \mathcal{C}$, on a $f(\xi^*) \leq f(\xi)$;
- $\xi^* \in \mathcal{C}$ est un minimum global si, pour tout $\xi \in \mathcal{C}$, on a $f(\xi^*) \leq f(\xi)$.

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 peut être 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

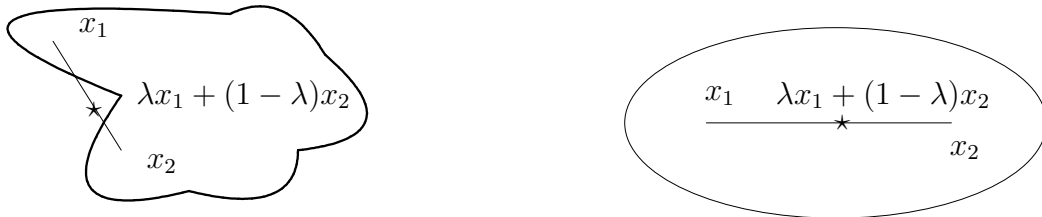


FIG. 2.3 – Ensemble non convexe (gauche), convexe (droite)

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

Propriété 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

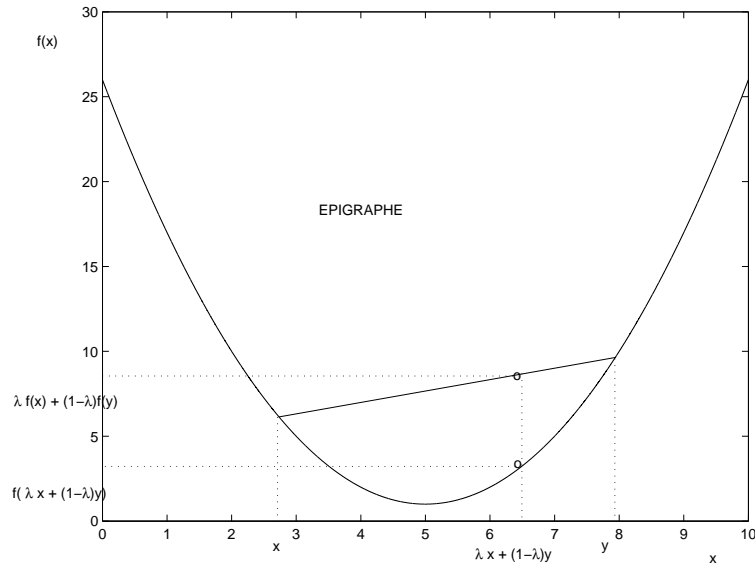
Definition 2.3.1 (Fonction convexe) Soit une fonction f

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

La fonction f est convexe si :

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

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

Remarque La condition 1. de la définition est implicitement contenue dans la condition 2.. La condition 2. en effet suppose que $\forall \lambda \in [0, 1]$, pour $x_1 \in \mathcal{D}$ et $x_2 \in \mathcal{D}$, $\lambda x_1 + (1 - \lambda)x_2 \in \mathcal{D}$, ce qui est la définition de \mathcal{D} ensemble convexe (Définition 2.2.1).

Propriété On appelle *épigraphe* d’une fonction f l’ensemble :

$$\varepsilon_f = \{(x, \tau) \in \mathcal{D} \times \mathbb{R} \mid f(x) \leq \tau\}.$$

Une fonction est convexe si et seulement si son épigraphe est convexe.

Si f est différentiable alors la condition 2. est équivalente à

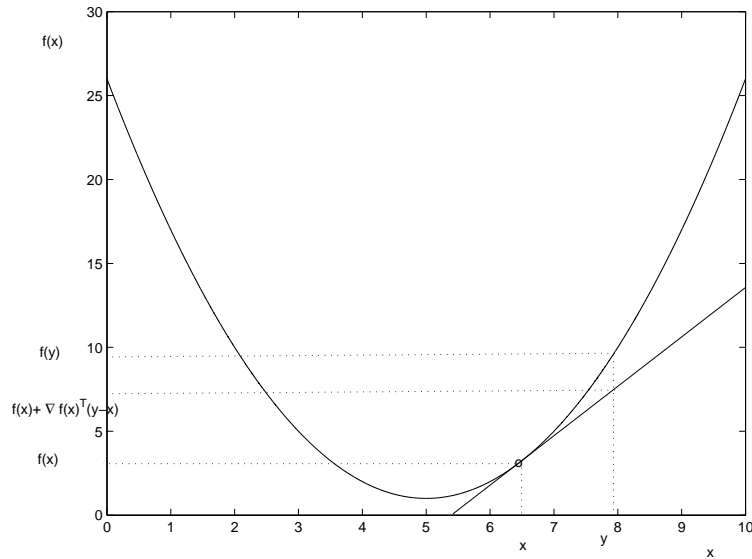
$$\forall x_1 \in \mathcal{D}, \quad \forall x_2 \in \mathcal{D}, \quad f(x_2) \geq f(x_1) + \nabla f(x_1)^T(x_2 - x_1) \quad (2.1)$$

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)).

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

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

Propriété Si f est une fonction convexe différentiable et que x^* est un minimum local de f alors il est global.

(En effet, si x^* est un minimum local de f alors $\nabla f(x^*) = 0$. D’après (2.1),

$$\forall x \in \mathcal{D}, \quad f(x) \geq f(x^*) + \nabla f(x^*)^T(x - x^*) = f(x^*). \quad (2.2)$$

ce qui correspond à la définition de x^* minimum global.)

Remarque Pour une fonction f différentiable, $\nabla f(x^*) = 0$ est une condition nécessaire pour que x^* soit un minimum local mais pas suffisante. Par exemple, la fonction $f(x) = x^3$ est telle que $\nabla f(0) = 0$ sans que $x = 0$ soit un minimum local. Par contre, pour une fonction f convexe, $\nabla f(x^*) = 0$ est une condition *suffisante* pour que x^* soit un minimum *global* du fait de l’inégalité (2.2). En réalité, on peut démontrer que la classe des fonctions f telles que $\nabla f(x^*) = 0$ soit une condition suffisante pour que x^* soit un minimum global dotée de quelques propriétés désirables correspond aux fonctions convexes différentiables².

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

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

(semi définie positive) avec

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

²Voir [?][Chapter 2] pour plus de détails.

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

Exemple de fonctions convexes $\mathcal{D} = \mathbb{R}, f(x) = e^x$; $\mathcal{D} =]-\pi, 0[$, $f(x) = \sin(x)$.

Exemple de fonction convexe $\mathcal{D} = \mathbb{R}^n, f(x) = x^T P x + 2q^T x + r$ avec $P = P^T \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n, r \in \mathbb{R}$. Quelle(s) condition(s) doi(ven)t 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

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

La fonction f est quasi convexe si :

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.

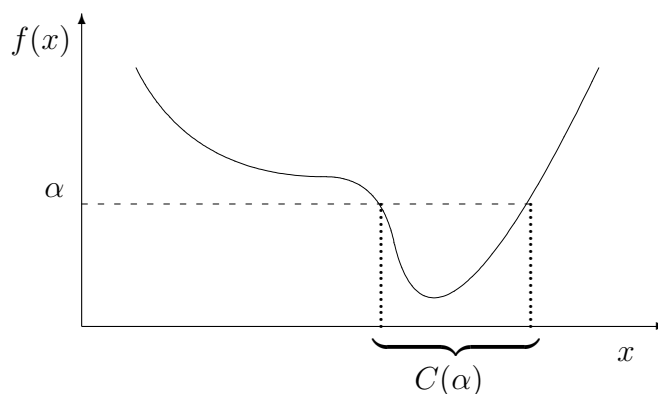


FIG. 2.6 – Fonction quasiconvexe

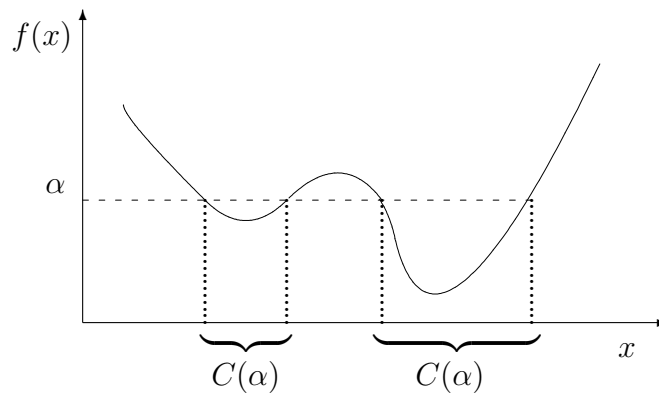


FIG. 2.7 – Fonction non quasiconvexe

Propriété Toute fonction convexe est quasi-convexe.

Si f est différentiable alors la seconde condition est équivalente à

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

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

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

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.

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

Soit

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

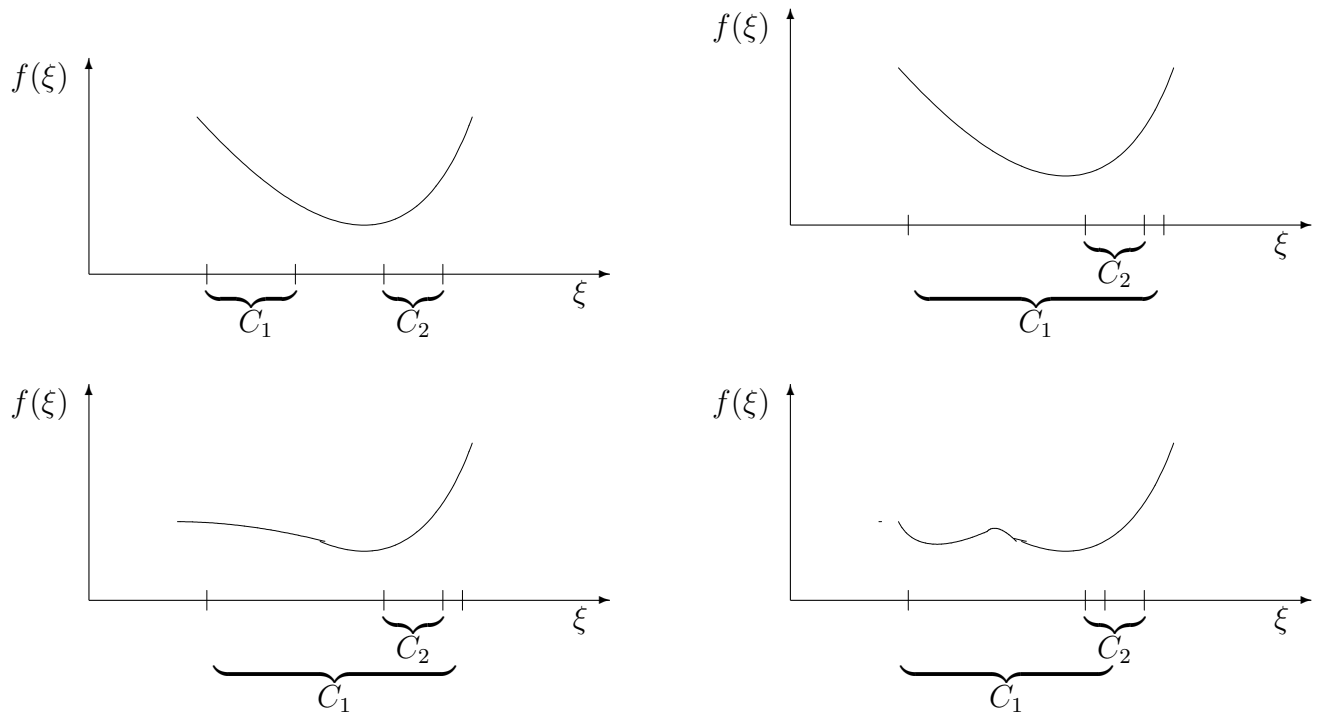
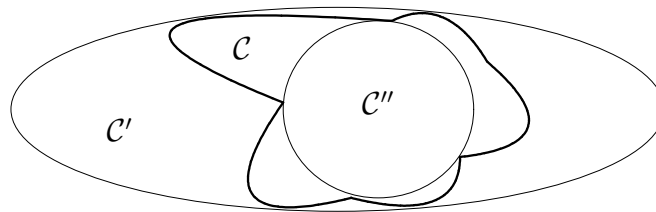


FIG. 2.8 – cas de figure considérés

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 $f(\xi^*)$?

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

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

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

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 5.1.

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 5.1, 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 5.1. 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

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

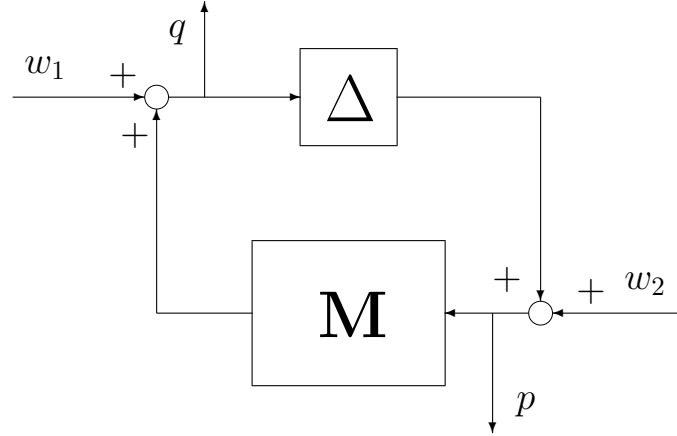
Remarque importante sur les notations Dans l'exemple précédent, les contraintes LMIs peuvent être écrites sous deux formats particuliers : d'une part, en utilisant les variables de décision sous la forme matricielle (par exemple P , dans l'inégalité (3.1)), d'autre part, en mettant les variables de décision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

$$\min_{\xi \in \mathbb{R}^2} \lambda_{\max}(F(\xi)).$$

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 $\xi = \frac{2}{3}$.

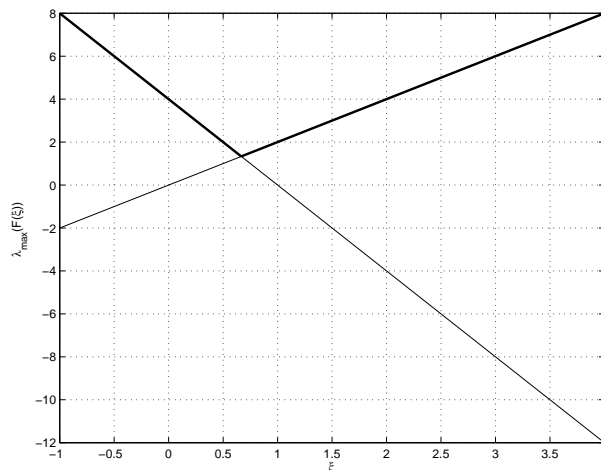


FIG. 3.2 – Tracé de $\lambda_{\max}(F(\xi))$ en fonction de x

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

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

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

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 :

$$\begin{cases} \underline{P} > 0 \\ \begin{bmatrix} A^T \underline{P} + \underline{P}A + C^T C & \underline{P}B + C^T D \\ B^T \underline{P} + D^T C & D^T D - \gamma^2 I \end{bmatrix} < 0 \end{cases} \quad (3.5)$$

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 (3.27) alors γ devient aussi une variable de décision. Cependant, (3.27) 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 :

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

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 (3.27) 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 27. 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 ?

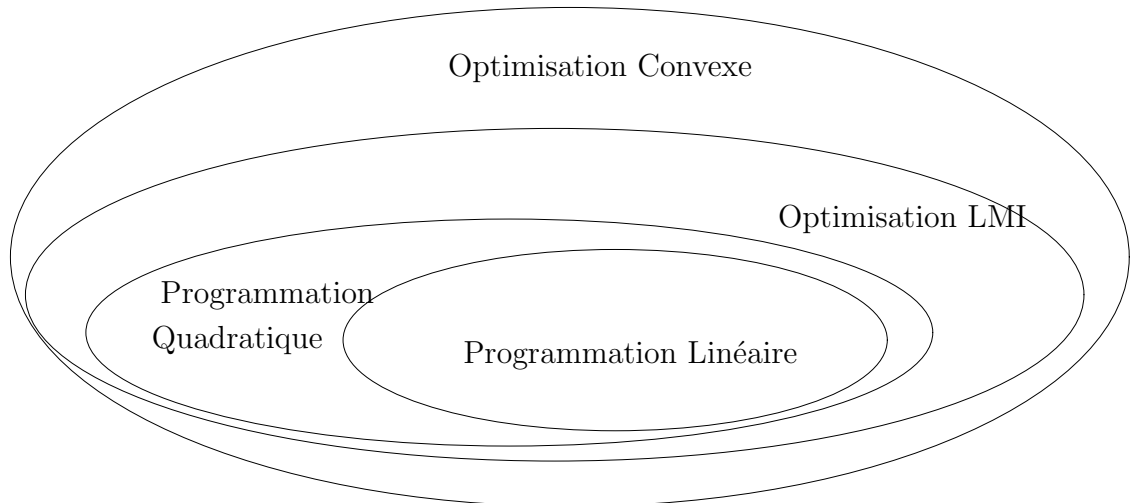


FIG. 3.3 – Différentes classes d'optimisation convexe

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 30, 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 $c^T \xi^* \leq c^T \xi^0$.

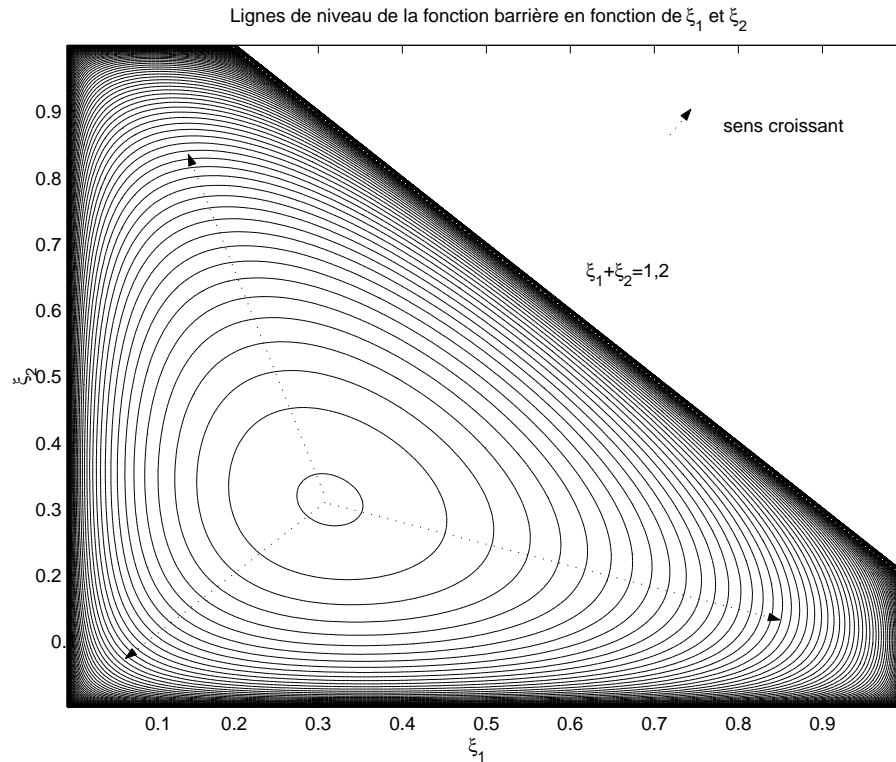


FIG. 3.4 – Courbes de niveau de la fonction barrière $\phi(\xi)$

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

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

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,

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

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 :

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

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 ??.

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 $A > 0$:

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

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

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

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

Correction de l'exercice On introduit une variable de décision "bidon" ("junk variable") $\lambda \in \mathbb{R}$ telle que :

$$\lambda > \xi^T A \xi + 2b^T \xi + c$$

soit :

$$\underbrace{\lambda - 2b^T \xi - c}_{C_s} - \underbrace{\xi}_{B_s}^T \underbrace{A}_{A_s^{-1}} \underbrace{\xi}_{B_s} > 0$$

Comme $A > 0$, le Lemme de Schur peut être appliqué :

$$\begin{bmatrix} \lambda - 2b^T \xi - c & \xi^T \\ \xi & A^{-1} \end{bmatrix} > 0$$

Par suite, le problème se récrit :

$$\text{minimiser} \quad \lambda$$

$$\lambda \in \mathbb{R}, \xi \in \mathbb{R}^m$$

$$\begin{bmatrix} \lambda - 2b^T \xi - c & \xi^T \\ \xi & A^{-1} \end{bmatrix} > 0$$

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 :

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} -\underline{P}_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} > 0$$

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

ce qui est équivalent à :

$$x^T \begin{bmatrix} -\underline{P}_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} x > 0 \tag{3.17}$$

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

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

$$\exists \tau_1 \in \mathbb{R}^+, \exists \tau_2 \in \mathbb{R}^+, \quad \begin{bmatrix} -\underline{P}_0^{-1} & 0 \\ 0 & 1 \end{bmatrix} > \tau_1 \begin{bmatrix} -P_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} + \tau_2 \begin{bmatrix} -P_2^{-1} & 0 \\ 0 & 1 \end{bmatrix} \tag{3.20}$$

Ce qui est équivalent à :

$$\exists \tau_1 \in \mathbb{R}^+, \exists \tau_2 \in \mathbb{R}^+, \quad \begin{bmatrix} -\underline{P}_0^{-1} + \tau_1 P_1^{-1} + \tau_2 P_2^{-1} & 0 \\ 0 & 1 - \tau_1 - \tau_2 \end{bmatrix} > 0 \tag{3.21}$$

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

$$-\underline{P}_0^{-1} + \tau_1 P_1^{-1} + \tau_2 P_2^{-1} > 0$$

soit par le lemme de Schur :

$$\tau_1 > 0, \tau_2 > 0, 1 > \tau_1 + \tau_2, \begin{bmatrix} \tau_1 P_1^{-1} + \tau_2 P_2^{-1} & I \\ I & \underline{P}_0 \end{bmatrix} > 0$$

Les variables de décision sont ici τ_1 , τ_2 et \underline{P}_0 : la contrainte ci-dessus est LMI puisqu'affine en les variables de décision. En mesurant la taille de l'ellipsoïde $\varepsilon_{\underline{P}_0}$ par la trace de la matrice \underline{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 Exercices

3.6.3.1 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.

3.6.3.2 Commande par placement de pôles

Soit le système continu défini par les équations d'état :

$$\dot{x}(t) = Ax(t) \tag{3.22}$$

où $x \in \mathbb{R}^n$ est le vecteur d'état et où $A \in \mathbb{R}^{n \times n}$ est la matrice d'évolution.

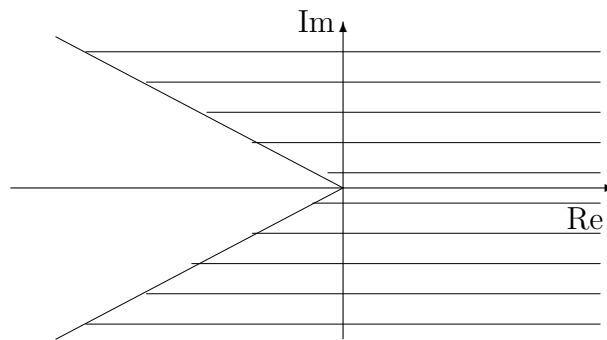


FIG. 3.5 – Secteur dans le plan complexe

Dans le plan complexe, on définit un secteur de pente $a > 0$ (a est ici donné) comme l'ensemble des points d'abscisse négative, en dessous de la droite de pente $-a$ et au dessus de la droite de pente $+a$ (voir Figure 3.5). L'ensemble des pôles du système (3.22) sont dans le secteur de pente a si et seulement s'il existe une matrice symétrique $P \in \mathbb{R}^{n \times n}$, définie positive telle que :

$$\begin{bmatrix} a(A^T P + PA) & PA - A^T P \\ A^T P - PA & a(A^T P + PA) \end{bmatrix} < 0 \tag{3.23}$$

Les trois questions suivantes peuvent être résolues indépendamment les unes des autres.

1. Ecrire la recherche de la matrice $P > 0$ telle que la contrainte (3.23) soit satisfaite sous la forme d'une contrainte LMI utilisant une variable de décision vectorielle ξ c'est-à-dire une contrainte de la forme :

$$\{\xi \mid F(\xi) > 0\}$$

où F est une matrice de fonctions affines en ξ . On précisera le choix du vecteur de variables de décision ξ ainsi que la fonction $F(\xi)$.

2. Montrer que la recherche de la matrice $P > 0$ telle que la contrainte (3.23) soit satisfaite est équivalente à la recherche d'une matrice $Q > 0$ telle que :

$$\begin{bmatrix} a(QA^T + AQ) & AQ - QA^T \\ QA^T - AQ & a(QA^T + AQ) \end{bmatrix} < 0 \quad (3.24)$$

3. On considère maintenant le système :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.25)$$

où $u(t)$ est l'entrée de commande. On désire mettre au point un correcteur par retour d'état défini par :

$$u(t) = Kx(t)$$

où la matrice K doit être déterminée telle que les pôles du système en boucle fermée sont tous localisés dans le secteur de pente a . A partir de l'inégalité (3.24), montrer que ce problème peut être formulé comme un problème de faisabilité sous contrainte LMI. On définira de façon précise les variables de décision qui seront conservées sous forme matricielle.

Correction

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 $n \times n$, on a

$$F(\xi) = \sum_{i=1}^{\frac{n(n+1)}{2}} \xi_i F_i$$

avec

$$F_0 = 0 \quad \text{et} \quad F_i = - \begin{bmatrix} a(A^T P_i + P_i A) & P_i A - A^T P_i \\ A^T P_i - P_i A & a(A^T P_i + P_i A) \end{bmatrix}.$$

2. L'inégalité (3.23) est équivalente à

$$\begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} a(A^T P + PA) & PA - A^T P \\ A^T P - PA & a(A^T P + PA) \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} < 0$$

avec $Q = P^{-1}$. En effectuant le produit des trois matrices et en simplifiant, on obtient (3.24).

3. On obtient :

$$\begin{bmatrix} a(Q(A+BK)^T + (A+BK)Q) & (A+BK)Q - Q(A+BK)^T \\ Q(A+BK)^T - (A+BK)Q & a(Q(A+BK)^T + (A+BK)Q) \end{bmatrix} < 0$$

soit en posant $Y = KQ$:

$$\begin{bmatrix} a(QA^T + BY^T + AQ + BY) & AQ + BY - A^TQ - Y^TB^T \\ A^TQ + Y^TB^T - AQ - BY & a(QA^T + BY^T + AQ + BY) \end{bmatrix} < 0$$

On a effectué un changement de variables de (Q, K) vers (Q, Y) qui est une bijection. Le problème obtenu est un problème de faisabilité LMI dont les variables de décision sont $Q = Q^T$ et Y .

3.6.3.3 Commande par précompensateur

On considère le système linéaire temps invariant suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ z(t) = Cx(t) + Du(t) \end{cases} \quad (3.26)$$

où $x(t) \in \mathbb{R}^n$ est le vecteur d'état du système, $u(t) \in \mathbb{R}^p$ est l'entrée du système et $z(t) \in \mathbb{R}^m$ est la sortie du système, avec $n > p$ et $n > m$.

Le système est stable et la norme H infini de la fonction de transfert de u vers z est inférieure à γ si et seulement si il existe une matrice symétrique \underline{P} telle que :

$$\begin{cases} \underline{P} > 0 \\ \begin{bmatrix} A^T\underline{P} + \underline{P}A + C^TC & \underline{P}B + C^TD \\ B^T\underline{P} + D^TC & D^TD - \gamma^2I \end{bmatrix} < 0 \end{cases} \quad (3.27)$$

Cette condition est équivalente à l'existence d'une matrice symétrique Q telle que :

$$\begin{cases} Q > 0 \\ \begin{bmatrix} QA^T + AQ & B & QC^T \\ B^T & -\gamma I & D^T \\ CQ & D & -\gamma I \end{bmatrix} < 0. \end{cases} \quad (3.28)$$

On s'intéresse à la détermination d'un précompensateur, c'est-à-dire d'une matrice telle que

$$u(t) = Kw(t)$$

avec $w(t)$ la nouvelle entrée. L'objectif est le système (3.26) avec ce précompensateur soit tel que la norme H infini de la fonction de transfert de w vers z soit inférieure à γ .

1. Montrer que ce problème peut être formulé comme un problème de faisabilité sous contrainte LMI. On définira de façon précise les variables de décision qui seront conservées sous forme matricielle.
2. Déterminer un problème de faisabilité sous contrainte LMI équivalent au problème précédent dans lequel la variable de décision liée à la matrice K a été éliminée.

Correction

1.

$$\left\{ \begin{array}{l} Q > 0 \\ \left[\begin{array}{ccc} QA^T + AQ & BK & QC^T \\ K^T B^T & -\gamma I & D^T \\ CQ & D & -\gamma I \end{array} \right] < 0. \end{array} \right.$$

La matrice totale est affine en Q et K : on a bien un problème de faisabilité LMI.

2. La seconde inégalité s'écrit :

$$\underbrace{\left[\begin{array}{ccc} QA^T + AQ & BK & QC^T \\ K^T B^T & -\gamma I & D^T \\ CQ & D & -\gamma I \end{array} \right]}_G + \left[\begin{array}{c} B \\ 0 \\ 0 \end{array} \right] K \underbrace{\left[\begin{array}{ccc} 0 & I & 0 \end{array} \right]}_U + \left[\begin{array}{c} 0 \\ I \\ 0 \end{array} \right] K^T \underbrace{\left[\begin{array}{ccc} B^T & 0 & 0 \end{array} \right]}_V < 0.$$

De plus, un choix possible est :

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

et

$$V_{\perp} = \begin{bmatrix} B^T_{\perp} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

En appliquant le lemme d'élimination, l'inégalité est équivalente à

$$\left[\begin{array}{cc} QA^T + AQ & QC^T \\ CQ & -\gamma I \end{array} \right] < 0$$

et

$$V_{\perp}^T \left[\begin{array}{ccc} QA^T + AQ & BK & QC^T \\ K^T B^T & -\gamma I & D^T \\ CQ & D & -\gamma I \end{array} \right] V_{\perp} < 0$$

Chapitre 4

Formulation de problèmes sous forme de problèmes d'optimisation LMI

4.1 Des moindres carrés (exemple)

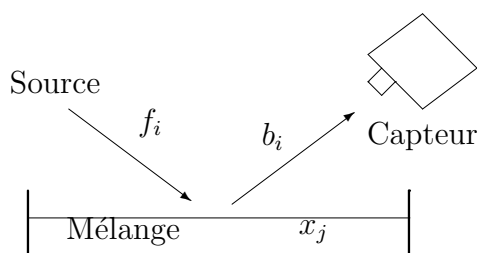


FIG. 4.1 – Exemple introductif

Exemple introductif Soient :

- un mélange de m produits : le produit j est de concentration x_j ;
- une source qui envoie de la lumière aux fréquences $f_i, i = 1, \dots, N$, avec $N > m$;
- des capteurs qui mesurent l'intensité b_i de ce qui est réfracté à chaque fréquence f_i .

On connaît a_{ij} les coefficients d'absorption du produit j à la fréquence f_i . L'objectif est de déterminer les concentrations x_j des m produits. Par suite, cela revient à déterminer les concentrations x_j telles que :

$$\forall i \in \{1, \dots, N\}, \quad b_i = \sum_{j=1}^m a_{ij} x_j$$

Puisque b_i est mesuré, il existe une erreur de mesure notée Δb_i , supposée “petit”. Par suite, il est plus réaliste d'écrire que :

$$\forall i \in \{1, \dots, N\}, \quad b_i = \sum_{j=1}^m a_{ij} x_j + \Delta b_i$$

Soit en introduisant la matrice $A = [a_{ij}]$, les vecteurs $x = [x_j]$, $b = [b_i]$, $\Delta b = [\Delta b_i]$:

$$Ax = b + \Delta b$$

On peut calculer une estimation \hat{x} du vecteur x en cherchant celui qui correspond à l'erreur la plus faible :

$$\min_{\hat{x} \in \mathbb{R}^m} \|\Delta b\| = \min_{\hat{x} \in \mathbb{R}^m} \|A\hat{x} - b\|. \quad (4.1)$$

Ici $\|A\hat{x} - b\|$ représente la mesure de l'écart entre :

- ce qui est effectivement mesuré b ;
- ce qui serait mesuré si $x = \hat{x}$ et s'il n'y avait pas de bruit (erreur) de mesure $\Delta b = 0$.

Ce problème est un problème de moindres carrés¹.

Quelle est la solution "standard" de ce problème ? Ce problème peut-il s'écrire comme un problème LMI ? Ce problème peut-il se mettre sous la forme d'un problème d'optimisation quadratique ?

D'autre part, notons que \hat{x} est un vecteur de concentrations :

$$\min_{\hat{x} \in \mathbb{R}^m, \hat{x} \geq 0} \|A\hat{x} - b\|. \quad (4.2)$$

La solution "standard" d'un problème de moindres carrés est-elle toujours valable quand on introduit la contrainte $\hat{x} \geq 0$? Peut-on adapter à ce problème les deux formulations par optimisation ?

On fait maintenant l'hypothèse que les coefficients d'absorption a_{ij} ne sont pas exactement connus : chaque coefficient a_{ij} appartient à un intervalle $[\underline{a}_{ij}, \bar{a}_{ij}]$ où les bornes \underline{a}_{ij} et \bar{a}_{ij} sont

¹**Digression sur l'intérêt du problème de moindres carrés.**

Prenons l'exemple défini par $N = 4$, $m = 2$ et

$$A = \begin{bmatrix} 1 & 100 \\ 0 & 1 \\ 0.5 & 0.25 \\ 100 & 1 \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} 201.8763 \\ 2.0164 \\ 1.0066 \\ 102.6585 \end{bmatrix}$$

Le vecteur de mesures est entâché d'une erreur inférieure à 1% : s'il n'y avait pas d'erreur de mesure, le vecteur b obtenu serait donné par :

$$\begin{bmatrix} 201 \\ 2 \\ 1 \\ 102 \end{bmatrix}.$$

Par suite, la valeur du vecteur x est

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Une première idée serait d'utiliser simplement les deux premières mesures pour estimer x puisque le vecteur x est de dimension deux. Une estimation \bar{x} serait alors :

$$\bar{x} = \begin{bmatrix} 1 & 100 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 201.8763 \\ 2.0164 \end{bmatrix} \quad \text{soit} \quad \bar{x} = \begin{bmatrix} 0.2338 \\ 2.0164 \end{bmatrix}.$$

La première composante du vecteur x est alors estimée avec une erreur de 76.6% pour une erreur de mesure sur b de 1% !

La résolution du problème de moindres carrés (4.1) donne :

$$\hat{x} = \begin{bmatrix} 1.0065 \\ 2.0087 \end{bmatrix}$$

L'erreur entre les composantes de \hat{x} et x est alors inférieure à 1% ! La solution obtenue est donc bien moins sensible à des erreurs de mesure. Cela illustre le fort intérêt des moindres carrés.

connues. Les coefficients de la matrice A appartiennent alors à un hypercube. Par exemple, pour $N = 3$ et $m = 1$, on a

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

avec $a_1 \in [\underline{a}_1, \bar{a}_1]$, $a_2 \in [\underline{a}_2, \bar{a}_2]$ et $a_3 \in [\underline{a}_3, \bar{a}_3]$. Le centre de l'hypercube est :

$$\begin{bmatrix} \frac{\underline{a}_1 + \bar{a}_1}{2} \\ \frac{\underline{a}_2 + \bar{a}_2}{2} \\ \frac{\underline{a}_3 + \bar{a}_3}{2} \end{bmatrix}$$

Les sommets sont

$$\begin{aligned} A_1 &= \begin{bmatrix} \underline{a}_1 \\ \underline{a}_2 \\ \underline{a}_3 \end{bmatrix} & A_2 &= \begin{bmatrix} \underline{a}_1 \\ \underline{a}_2 \\ \bar{a}_3 \end{bmatrix} & A_3 &= \begin{bmatrix} \underline{a}_1 \\ \bar{a}_2 \\ \underline{a}_3 \end{bmatrix} & A_4 &= \begin{bmatrix} \underline{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{bmatrix} \\ A_5 &= \begin{bmatrix} \bar{a}_1 \\ \underline{a}_2 \\ \underline{a}_3 \end{bmatrix} & A_6 &= \begin{bmatrix} \bar{a}_1 \\ \underline{a}_2 \\ \bar{a}_3 \end{bmatrix} & A_7 &= \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \underline{a}_3 \end{bmatrix} & A_8 &= \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{bmatrix} \end{aligned}$$

Soit 2^3 sommets. Si $A \in \mathbb{R}^{N \times m}$ alors on aura $L = 2^{N \times m}$ sommets A_i . A l'aide des coordonnées barycentriques, on peut définir l'ensemble \mathcal{A} des matrices A obtenues en faisant varier chaque coefficient a_{ij} dans son intervalle $[\underline{a}_{ij}, \bar{a}_{ij}]$:

$$\mathcal{A} = \left\{ A \mid \exists \lambda_i \geq 0, \sum_{i=1}^L \lambda_i = 1, A = \sum_{i=1}^L \lambda_i A_i \right\}$$

On suppose qu'il n'y a pas d'erreur sur le vecteur b .

On cherche maintenant à résoudre le problème :

$$\begin{aligned} \min_{\hat{x} \in \mathbb{R}^m} \quad & \max_{A \in \mathcal{A}} \|A\hat{x} - b\|. \\ & \hat{x} \geq 0 \end{aligned} \quad (4.3)$$

On considère un problème *pire cas*. Pour \hat{x} donné, la quantité $\max_{A \in \mathcal{A}} \|A\hat{x} - b\|$ correspond à l'erreur maximale (la "pire") qui va être faite sur l'ensemble de tous les éléments de \mathcal{A} . On va donc choisir \hat{x} pour que l'erreur dans le pire cas soit la plus faible possible.

Montrer que cela peut se réécrire comme la résolution du problème d'optimisation suivant :

$$\begin{aligned} \min_{\mu \in \mathbb{R}, \hat{x} \in \mathbb{R}^m} \quad & \mu \\ & \hat{x} \geq 0 \\ & \forall \lambda_i \geq 0, \sum_i \lambda_i = 1, \begin{bmatrix} \mu & (\sum_i \lambda_i A_i \hat{x} - b)^T \\ \sum_i \lambda_i A_i \hat{x} - b & I \end{bmatrix} \geq 0 \end{aligned} \quad (4.4)$$

Ce problème d'optimisation est convexe mais avec un nombre *infini* de contraintes.

Lemme 4.1.1

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

Le démontrer. Le résultat est valable avec $>$ remplacé par \geq . L'intérêt est que la vérification d'un nombre infini d'inégalités (gauche) peut être équivalent à la vérification d'un nombre fini d'égalité (droite).

Par suite, le problème d'optimisation (4.4) est *équivalent* au problème d'optimisation suivant :

$$\begin{aligned} \min_{\mu \in \mathbb{R}, \hat{x} \in \mathbb{R}^m} \quad & \mu \\ \forall i \in \{1, \dots, L\}, \quad & \begin{bmatrix} \hat{x} \geq 0 \\ \mu & (A_i \hat{x} - b)^T \\ A_i \hat{x} - b & I \end{bmatrix} \geq 0 \end{aligned} \quad (4.5)$$

qui est un problème d'optimisation avec un nombre fini de contraintes. Cependant, le nombre de contraintes est une fonction exponentielle de la taille du problème puisque $L = 2^{N \times m} \dots$

Approche alternative Le fait que chaque coefficient a_{ij} varie dans l'intervalle $[\underline{a}_{ij}, \bar{a}_{ij}]$ peut se réécrire de la façon suivante :

$$[\underline{a}_{ij}, \bar{a}_{ij}] = \{a_{ij} \mid \exists \delta_{ij} \in [-1, 1], a_{ij} = a_{ij}^{nom} + a_{ij}^{gab} \delta_{ij}\}$$

avec $a_{ij}^{nom} = \frac{\bar{a}_{ij} + \underline{a}_{ij}}{2}$ et $a_{ij}^{gab} = \frac{\bar{a}_{ij} - \underline{a}_{ij}}{2}$.

Dans le cas où $N = 3$ et $m = 1$, on aura :

$$\begin{aligned} A &= \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \\ &= \begin{bmatrix} a_1^{nom} \\ a_2^{nom} \\ a_3^{nom} \end{bmatrix} + \begin{bmatrix} a_1^{gab} & 0 & 0 \\ 0 & a_2^{gab} & 0 \\ 0 & 0 & a_3^{gab} \end{bmatrix} \underbrace{\begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 \end{bmatrix}}_{\Delta} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

Dans le cas général :

$$\begin{aligned}
A &= \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{N1} & \dots & a_{Nm} \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} a_{11}^{nom} & \dots & a_{1m}^{nom} \\ \vdots & \ddots & \vdots \\ a_{N1}^{nom} & \dots & a_{Nm}^{nom} \end{bmatrix}}_{A^{nom}} + \dots \\
&\quad \underbrace{\begin{bmatrix} a_{11}^{gab} & 0 & 0 & \dots & a_{1m}^{gab} & 0 & 0 \\ 0 & \ddots & 0 & \dots & 0 & \ddots & 0 \\ 0 & 0 & a_{N1}^{gab} & \dots & 0 & 0 & a_{Nm}^{gab} \end{bmatrix}}_B \underbrace{\begin{bmatrix} \delta_{11} & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & \ddots & \dots & \dots & \dots & \vdots \\ \vdots & \ddots & \delta_{N1} & \ddots & \dots & \dots & \vdots \\ \vdots & \dots & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & \ddots & \delta_{1m} & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & \delta_{Nm} \end{bmatrix}}_{\Delta} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_C
\end{aligned}$$

Le problème d'optimisation (4.3) se réécrit alors :

$$\begin{aligned}
&\min_{\mu \in \mathbb{R}, \hat{x} \in \mathbb{R}^m} \mu \\
&\quad \hat{x} \geq 0 \\
&\quad \forall \delta_{ij} \in [-1, 1], \left[\begin{array}{cc} \mu & ((A^{nom} + B\Delta C)\hat{x} - b)^T \\ (A^{nom} + B\Delta C)\hat{x} - b & I \end{array} \right] \geq 0 \tag{4.6}
\end{aligned}$$

$$\forall \delta_{ij} \in [-1, 1], \left[\begin{array}{cc} \mu & ((A^{nom} + B\Delta C)\hat{x} - b)^T \\ (A^{nom} + B\Delta C)\hat{x} - b & I \end{array} \right] \geq 0$$

\Leftrightarrow

$$\forall \delta_{ij} \in [-1, 1], \forall v_1 \in \mathbb{R}, \forall v_2 \in \mathbb{R}^N,$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T \\ A^{nom}\hat{x} - b & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + v_2^T B\Delta C\hat{x}v_1 + v_1^T \hat{x}^T C^T \Delta B^T v_2 \geq 0$$

Posons $p = \Delta B^T v_2$. Alors

$$\begin{aligned} & \forall \delta_{ij} \in [-1, 1], \quad \forall v_1 \in \mathbb{R}, \quad \forall v_2 \in \mathbb{R}^N, \\ & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T \\ A^{nom}\hat{x} - b & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + v_2^T B \Delta C \hat{x} v_1 + v_1^T \hat{x}^T C^T \Delta^T B^T v_2 \geq 0 \\ & \quad \Updownarrow \\ & \forall v_1 \in \mathbb{R}, \quad \forall v_2 \in \mathbb{R}^N, \\ & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T \\ A^{nom}\hat{x} - b & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + p^T C \hat{x} v_1 + v_1^T \hat{x}^T C^T p \geq 0 \\ & \text{avec } p \in \{p \mid \exists \delta_{ij} \in [-1, 1], p = \Delta B^T v_2\} \end{aligned}$$

Or

$$p^T p = v_2^T B \Delta \Delta B^T v_2 = v_2^T B \begin{bmatrix} \delta_{11}^2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \delta_{Nm}^2 \end{bmatrix} B^T v_2 \leq v_2^T B B^T v_2$$

En introduisant

$$S = \begin{bmatrix} s_1 & 0 & \dots & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & s_k \end{bmatrix} \text{ avec } s_i > 0, \quad i = 1, \dots, k.$$

avec $k = N \times m$ on a :

$$p^T S p \leq v_2^T B S B^T v_2$$

(prise en compte de la structure particulière de la matrice Δ .) Vrai pour toute valeur de s_i strictement positive. Pour toute matrice S de ce type-là, on aura :

$$\{p \mid \exists \delta_{ij} \in [-1, 1], p = \Delta C v_1\} \subset \{p \mid p^T S p \leq v_2^T B S B^T v_2\}$$

D'où

$$\begin{aligned} & \forall v_1 \in \mathbb{R}, \quad \forall v_2 \in \mathbb{R}^N, \\ & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T \\ A^{nom}\hat{x} - b & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + p^T C \hat{x} v_1 + v_1^T \hat{x}^T C^T p \geq 0 \\ & \text{avec } p \in \{p \mid \exists \delta_{ij} \in [-1, 1], p = \Delta B^T v_2\} \\ & \quad \Uparrow \\ & \forall v_1 \in \mathbb{R}, \quad \forall v_2 \in \mathbb{R}^N, \\ & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T \\ A^{nom}\hat{x} - b & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + p^T C \hat{x} v_1 + v_1^T \hat{x}^T C^T p \geq 0 \\ & \text{avec } p \in \{p \mid p^T S p \leq v_2^T B S B^T v_2\} \end{aligned}$$

D'après la \mathcal{S} procédure, on a alors la condition suffisante suivante :

$$\forall v_1 \in \mathbb{R}, \forall v_2 \in \mathbb{R}^N, \forall p \in \mathbb{R}^k,$$

$$\begin{bmatrix} v_1 \\ v_2 \\ p \end{bmatrix}^T \begin{bmatrix} \mu & (A^{nom}\hat{x} - b)^T & \hat{x}^T C^T \\ A^{nom}\hat{x} - b & I - BSB^T & 0 \\ C\hat{x} & 0 & S \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ p \end{bmatrix} \geq 0$$

Que dire du problème d'optimisation :

$$\min_{\mu \in \mathbb{R}, \xi \in \mathbb{R}^m, S} \mu$$

$$\begin{bmatrix} \hat{x} \geq 0 \\ \mu & (A^{nom}\hat{x} - b)^T & \hat{x}^T C^T \\ A^{nom}\hat{x} - b & I - BSB^T & 0 \\ C\hat{x} & 0 & S \end{bmatrix} \geq 0$$

par rapport au problème d'optimisation (4.6) ?

4.2 Réglage d'un correcteur Proportionnel Dérivé sur un moteur à courant continu (exemple)

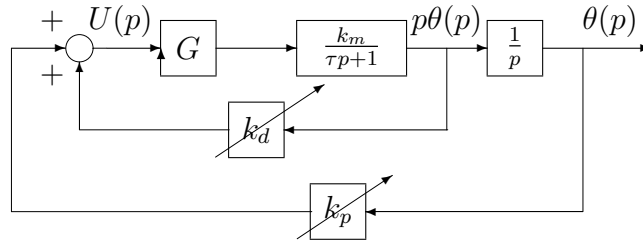


FIG. 4.2 – Commande d'un moteur à courant continu

On considère un moteur à courant continu. La position $\theta(t)$ et la vitesse angulaire $\dot{\theta}(t)$ de l'arbre du moteur sont mesurées. Au temps $t = 0$, l'arbre du moteur à courant continu est à un angle $\theta_0 \neq 0$. On désire déterminer la tension à appliquer aux bornes du moteur pour ramener la position de l'angle du moteur à 0 avec une certaine rapidité. La rapidité sera mesurée par un taux de décroissance α défini ci-après. La tension sera recherchée telle que

$$U(p) = (k_d p + k_p)\theta(p) \quad \text{soit} \quad u(t) = k_d \dot{\theta}(t) + k_p \theta(t) \quad (4.7)$$

avec k_d et k_p deux gains dont on cherche à déterminer la valeur pour assurer la rapidité désirée. Le moteur à courant continu peut être représenté par l'équation suivante :

$$p\theta(p) = \frac{Gk_m}{\tau p + 1} U(p) \quad \text{soit} \quad \tau \frac{d^2\theta(t)}{dt^2} + \frac{d\theta(t)}{dt} = Gk_m u(t) \quad (4.8)$$

En éliminant $u(t)$ entre l'équation (4.7) et l'équation (4.8), on obtient :

$$\tau \frac{d^2\theta(t)}{dt^2} = (Gk_mk_d - 1) \frac{d\theta(t)}{dt} + Gk_mk_p\theta(t).$$

On a alors :

$$\frac{d}{dt} \left(\underbrace{\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}}_{x(t)} \right) = \left(\underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix}}_A + \underbrace{\begin{bmatrix} 0 \\ -\frac{Gk_m}{\tau} \end{bmatrix}}_B \underbrace{\begin{bmatrix} k_p & k_d \end{bmatrix}}_K \right) \underbrace{\begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}}_{x(t)}$$

soit

$$\frac{dx(t)}{dt} = (A + BK)x(t)$$

Rappel Soit le système :

$$\dot{x}(t) = \bar{A}x(t)$$

Pour un système (asymptotiquement) stable, le taux de décroissance exponentiel se définit comme le plus grand $\alpha > 0$ 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 \bar{A} . Dans le cas où \bar{A} est diagonalisable, en notant $\lambda_i = \sigma_i + j\omega_i$ les valeurs propres de la matrice \bar{A} , on a la i^{ieme} composante du vecteur d'état qui est définie par :

$$x_i(t) = \sum_i e^{\sigma_i t} (a_i \cos(\omega_i t) + b_i \sin(\omega_i t))$$

où a_i et b_i dépendent de la condition initiale x_0 . On aura alors $\sigma_i \leq -\alpha$. α est une borne inférieure sur le taux de décroissance si et seulement s'il existe une matrice \underline{P} telle que :

$$\begin{aligned} \underline{P} &> 0 \\ \bar{A}^T \underline{P} + \underline{P} \bar{A} + 2\alpha \underline{P} &< 0 \end{aligned}$$

Mise en œuvre sur le moteur Trouver \underline{K} et \underline{P} telles que

$$\begin{aligned} \forall x_1 \in \mathbb{R}^n \quad x_1^T \underline{P} x_1 &> 0 \\ \forall x_2 \in \mathbb{R}^n \quad x_2^T ((A + BK)^T \underline{P} + \underline{P}(A + BK) + 2\alpha \underline{P}) x_2 &< 0 \end{aligned}$$

Problème : seconde inégalité 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,

$$\begin{aligned} \forall \tilde{x}_1 \in \mathbb{R}^n \quad \tilde{x}_1^T \underline{P}^{-1} \tilde{x}_1 &> 0 \\ \forall \tilde{x}_2 \in \mathbb{R}^n \quad \tilde{x}_2^T (A \underline{P}^{-1} + \underline{P}^{-1} A^T + BK \underline{P}^{-1} + \underline{P}^{-1} K^T B^T + 2\alpha \underline{P}^{-1}) \tilde{x}_2 &< 0 \end{aligned} \tag{4.9}$$

Les deux inégalités obtenues 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 :

$$\begin{aligned} \forall \tilde{x}_1 \in \mathbb{R}^n \quad \tilde{x}_1^T \underline{Q} \tilde{x}_1 &> 0 \\ \forall \tilde{x}_2 \in \mathbb{R}^n \quad \tilde{x}_2^T (A \underline{Q} + \underline{Q} A^T + B \underline{Y} + \underline{Y}^T B^T + 2\alpha \underline{Q}) \tilde{x}_2 &< 0 \end{aligned} \tag{4.10}$$

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 (4.10) sont bien affines en les variables \underline{Q} et \underline{Y} . Par suite, la recherche d'une loi de commande $u(t) = \underline{K}x(t)$ assurant un taux de décroissance supérieur à α peut s'effectuer de la façon suivante :

1. Trouver \underline{Q} et \underline{Y} tels que les inégalités (4.10) soient satisfaites (problème de faisabilité) ;
2. $\underline{P} = \underline{Q}^{-1}$ et $\underline{K} = \underline{Y}\underline{P}$.

4.3 Retour de sortie statique ou la difficulté de chercher des lois de commande simples (exercice)

Soit le système dynamique, linéaire stationnaire d'équation :

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

où $x(t) \in \mathbb{R}^n$ est l'état du système, $u(t) \in \mathbb{R}^p$ l'entrée de commande et $y(t) \in \mathbb{R}^m$ la sortie mesurée du système (avec $p < n$ et $m < n$). On recherche une loi de commande statique, par retour de sortie, telle que le système bouclé soit stable. La loi de commande est de la forme $u(t) = \underline{K}y(t)$ avec $\underline{K} \in \mathbb{R}^{p \times m}$, la matrice à déterminer. Ecrire la condition de stabilité de Lyapunov pour le système bouclé. Constater que contrairement au cas de la commande par retour d'état (c'est-à-dire une loi de commande de la forme $u(t) = \underline{K}x(t)$), on ne peut pas mettre ce problème sous forme d'un problème d'optimisation LMI, en utilisant la même approche que pour le retour d'état.

4.4 Retour d'état avec coût quadratique garanti pour une condition initiale donnée (exercice)

Soit le système dynamique, linéaire stationnaire d'équation d'état :

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

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

1. le système bouclé est stable ;
2. pour deux matrices données, définies positives $Q = Q^T \in \mathbb{R}^{n \times n}$, $R = R^T \in \mathbb{R}^{p \times p}$, une condition initiale donnée x_0 et un scalaire γ positif donné :

$$\int_0^{+\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt < \gamma.$$

Pour cela, on recherchera une fonction $V(x) = x^T \underline{P} x$, avec $\underline{P} = \underline{P}^T > 0$, telle que

$$V(x_0) < \gamma$$

et telle que la dérivée de $V(x)$ le long des trajectoires du système bouclé $\dot{V}(x)$ vérifie la relation :

$$\dot{V}(x) + x(t)^T Q x(t) + u(t)^T R u(t) < 0. \quad (4.11)$$

Ces conditions assurent :

1. la stabilité car V est fonction de Lyapunov (la relation (4.11) implique que $\dot{V}(x) < 0$ car les matrices Q et R sont définies positives) ;
2. Par intégration de 0 à T , l'inégalité (4.11) donne :

$$V(x(T)) - V(x_0) + \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt < 0$$

le système étant stable, quand T tend vers l'infini, $V(x(T))$ tend vers 0, d'où :

$$\int_0^{+\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt < V(x_0) < \gamma.$$

Mettre la recherche d'un correcteur par retour d'état assurant les propriétés précédentes sous la forme d'un problème de faisabilité sur des contraintes LMIs.

4.5 Simulation temporelle des systèmes (exercice)

Lorsque l'on met au point un système (par exemple, un circuit électrique), il est usuel d'utiliser la simulation numérique par regarder comment vont évoluer ses différentes variables (tensions, intensités, etc) en fonction du temps pour différentes conditions initiales et différentes valeurs de ses paramètres physiques (par exemple résistances, capacités, etc). Il est clair qu'il est impossible de tester par simulation toutes les configurations possibles correspondant à toutes les conditions initiales et à toutes les valeurs des paramètres du système. La méthode qui est ici étudiée va permettre d'estimer l'évolution des variables d'un système, pour toutes les configurations possibles.

Soit x_k un vecteur de \mathbb{R}^n dont les composantes sont les variables du système à l'instant k . Soit x_{k+1} un vecteur de \mathbb{R}^n dont les composantes sont les variables du système à l'instant $k + 1$. On suppose que le lien entre x_{k+1} et x_k est donné par :

$$x_{k+1} = A x_k \tag{4.12}$$

où A est une matrice de $\mathbb{R}^{n \times n}$ que l'on sait simplement appartenir à l'ensemble \mathcal{A} défini par :

$$\mathcal{A} = \left\{ A \mid \exists \lambda_i \geq 0, \quad i \in \{1, \dots, m\}, \quad \sum_{i=1}^m \lambda_i = 1, \quad \text{et } A = \sum_{i=1}^m \lambda_i A_i \right\}$$

où A_1, A_2, \dots, A_m est un ensemble de m matrices données. \mathcal{A} contient en fait les paramètres du système que l'on sait simplement appartenir à un ensemble.

- 1) Supposons que x_k soit donné. Soit l'ellipsoïde de centre c_{k+1} et défini par la matrice symétrique définie positive P_{k+1} :

$$\varepsilon_{P_{k+1}} = \{x \mid (x - c_{k+1})^T P_{k+1}^{-1} (x - c_{k+1}) \leq 1\}.$$

Montrez que x_{k+1} défini par la condition (4.12) appartient à l'ellipsoïde $\varepsilon_{P_{k+1}}$ si et seulement si

$$\forall i \in \{1, \dots, m\}, \quad \begin{bmatrix} 1 & (A_i x_k - c_{k+1})^T \\ (A_i x_k - c_{k+1}) & P_{k+1} \end{bmatrix} > 0$$

Si on mesure la taille de l'ellipsoïde $\varepsilon_{P_{k+1}}$ par la somme des longueurs des demi axes au carré, écrire le problème d'optimisation consistant à trouver le plus petit ellipsoïde (caractérisé par son centre c_{k+1} et la matrice P_{k+1}) contenant x_k . A quelle classe de problèmes d'optimisation appartient-il ?

2) On désire maintenant étudier x_{k+1} pour tous les vecteurs x_k appartenant à l'ellipsoïde de centre c_k et défini par la matrice symétrique définie positive P_k :

$$\varepsilon_{P_k} = \{x \mid (x - c_k)^T P_k^{-1} (x - c_k) \leq 1\}.$$

Montrer que

$$\begin{bmatrix} 1 & (A_i x_k - c_{k+1})^T \\ (A_i x_k - c_{k+1}) & P_{k+1} \end{bmatrix} > 0 \Leftrightarrow \begin{bmatrix} x_k \\ 1 \end{bmatrix}^T \left[\begin{array}{c|c} -A_i^T P_{k+1}^{-1} A_i & A_i^T P_{k+1}^{-1} c_{k+1} \\ \hline c_{k+1}^T P_{k+1}^{-1} A_i & 1 - c_{k+1}^T P_{k+1}^{-1} c_{k+1} \end{array} \right] \begin{bmatrix} x_k \\ 1 \end{bmatrix} > 0$$

Montrer qu'une condition suffisante pour que :

$$\begin{bmatrix} x_k \\ 1 \end{bmatrix}^T \left[\begin{array}{c|c} -A_i^T P_{k+1}^{-1} A_i & A_i^T P_{k+1}^{-1} c_{k+1} \\ \hline c_{k+1}^T P_{k+1}^{-1} A_i & 1 - c_{k+1}^T P_{k+1}^{-1} c_{k+1} \end{array} \right] \begin{bmatrix} x_k \\ 1 \end{bmatrix} > 0$$

pour l'ensemble des vecteurs x_k appartenant à l'ellipsoïde ε_{P_k} est qu'il existe un réel positif τ_i tel que pour tout $x_k \in \mathbb{R}^n$, on a :

$$\begin{bmatrix} x_k \\ 1 \end{bmatrix}^T \left[\begin{array}{c|c} \tau_i P_k - A_i^T P_{k+1}^{-1} A_i & \tau_i P_k c_k + A_i^T P_{k+1}^{-1} c_{k+1} \\ \hline \tau_i c_k^T P_k + c_{k+1}^T P_{k+1}^{-1} A_i & 1 - \tau_i + \tau_i c_k^T P_k^{-1} c_k - c_{k+1}^T P_{k+1}^{-1} c_{k+1} \end{array} \right] \begin{bmatrix} x_k \\ 1 \end{bmatrix} > 0$$

En fait, cette condition est aussi *nécessaire* : ceci est admis. De plus, en admettant que cette condition est équivalente à :

$$\left[\begin{array}{c|c} \tau_i P_k - A_i^T P_{k+1}^{-1} A_i & \tau_i P_k c_k + A_i^T P_{k+1}^{-1} c_{k+1} \\ \hline \tau_i c_k^T P_k + c_{k+1}^T P_{k+1}^{-1} A_i & 1 - \tau_i + \tau_i c_k^T P_k^{-1} c_k - c_{k+1}^T P_{k+1}^{-1} c_{k+1} \end{array} \right] > 0$$

montrer qu'étant donné l'ellipsoïde ε_{P_k} , rechercher le plus petit ellipsoïde $\varepsilon_{P_{k+1}}$ (au sens précédemment défini) contenant x_{k+1} est un problème d'optimisation convexe sous contraintes LMI.

Chapitre 5

Annexe A : rappels sur les matrices

5.1 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 5.1.1 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 5.1.2 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 5.1.3 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 5.1.4 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 5.1.5 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 :

$$A = U\Sigma V^* \tag{5.1}$$

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 $n > m$.

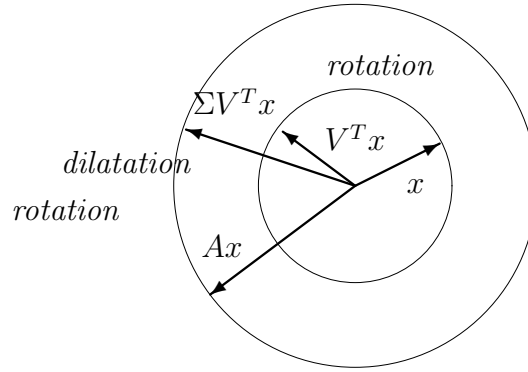


FIG. 5.1 – Transformation de v par A quand $\sigma_1 = 2$ et $\sigma_2 = 1$

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).

5.1.1 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 5.1.6 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 5.1.7 (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 5.1.8 (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 5.1.9 (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 6

Annexe B : Ensembles convexes particuliers

6.1 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 6.1 correspond à un ellipsoïde défini par la matrice :

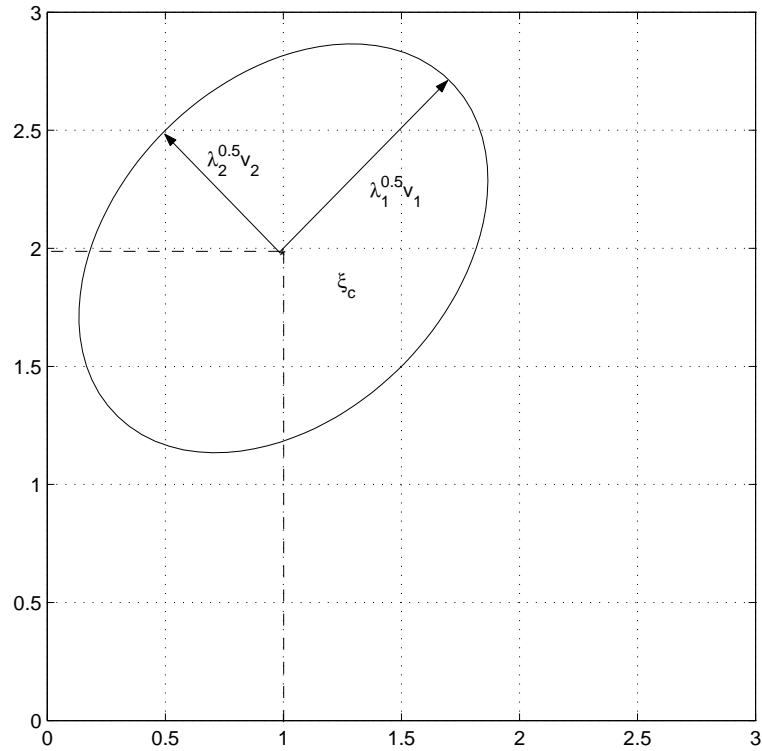
$$P = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

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. 6.1 – 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,

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

6.2 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 :

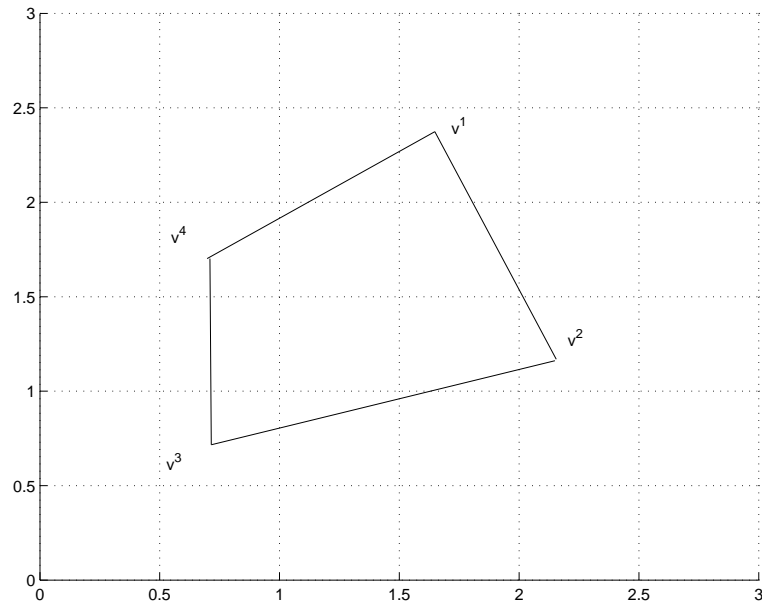
$$\mathcal{P} = \{x \mid a_j^T x \leq b_j, \quad j = 1, \dots, m\}$$

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. 6.2 – 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 7

Annexe C : Programmes de la section 4.1

7.1 Programme principal

```
%  
% Exemple de resolutions de problemes de type moindre carre  
% avec des matrices A et b generees aleatoirement.  
%  
% min || Ax-b||  
% x  
%  
% G. Scorletti 2002  
  
A = rand(4,3); % on essaiera successivement pour dimensions de A :  
           % (4,3) (5,3) (8,2)  
b = rand(4,1);  
  
N = size(A,1); m = size(A,2);  
  
% min || Ax-b||  
% x  
%  
% xopt = argmin || Ax-b||  
% x  
%  
% solution classique  
  
tic xclass = inv(A'*A)*A'*b; toc  
  
% resolution des moindres carres simples : approche LMI  
  
LMI_moinsre_carres  
  
% On appelle residu la quantite ||Ax-b||. On peut comparer la solution
```

```

% classique et la solution LMI

fprintf('comparaison residus approche classique %f et LMI %f',
        sqrt((A*xclass-b)'*(A*xclass-b)), sqrt(copt));

pause

% On introduit la contrainte que les composantes de x doivent etre positive...
%
% min || Ax-b||
% x tel que
%         pour tout i x(i)>=0
%
% C'est un probleme de programmation quadratique
%

H = 2*A'*A; f = -2*b'*A;

G = -eye(m); h = zeros(m,1);

xpq = quadprog(H,f, G, h); % Matlab version 6
%xpq = qp(H,f, G, h);    % Matlab version 5.2

% On peut aussi le résoudre par programmation LMI

LMI_moindres_carres_positifs

% comparaison des residus obtenus par les deux approches

fprintf('comparaison des residus approche qp %f et LMI %f',
        sqrt((A*xpq -b)'*(A*xpq-b)), sqrt((A*xlmi2-b)'*(A*xlmi2-b)))

pause

% Maintenant, on suppose que l'on a 100*w_incertainite pour cent
% d'incertitudes sur les coefficients de la matrice A. La premiere
% approche est d'utiliser une formulation exacte basee sur la
% representation polytopique de A: A est ecrit comme une combinaison
% lineaire de sommets.

w_incertainite = 0.1;

uncertain_polytopes

% La seconde approche est d'utiliser une condition suffisante basee
% sur la representation LFT de A.

```

```
uncertain_LFT
```

```
% Comparaison des resultats
```

```
fprintf('\n residus avec +/- 10 pour cent incertitude polytope %f et LFT %f \n',
        sqrt(copt3), sqrt(copt4) );
```

7.2 Moindres carrés par LMI

```
% resolution des moindres carres simples : approche LMI
```

```
setlmis([]);
```

```
mu = lmivar(2,[1,1]); % on definit la variable mu
```

```
x = lmivar(2,[m,1]); % on definit la variable x
```

```
%
```

```
% Construction de la contrainte LMI
```

```
%
```

```
lmiterm([-1 1 1 mu],1,1);
```

```
% LMI #1: mu
```

```
lmiterm([-1 2 1 x],A,1);
```

```
% LMI #1: A*x
```

```
lmiterm([-1 2 1 0],-b);
```

```
% LMI #1: -b
```

```
lmiterm([-1 2 2 0],eye(N));
```

```
% LMI #1: eye(N)
```

```
moindre=getlmis;
```

```
%
```

```
% Valeur de la fonction de cout
```

```
%
```

```
c = [1; zeros(m,1)];
```

```
options(5)=0;
```

```
tic [copt,xopt] = mincx(moindre,c,options); toc
```

```
xlmi = xopt(2:m+1)
```

7.3 Moindres carrés avec positivité de la solution par LMI

```
% resolution des moindres carres avec contrainte de positivite des composante de x :
```

```

% approche programmation LMI

setlmis([]);

mu = lmivar(2,[1,1]); x = lmivar(2,[m,1]);

lmiterm([-1 1 1 mu],1,1);           % LMI #1: mu
lmiterm([-1 2 1 x],A,1);           % LMI #1: A*x
lmiterm([-1 2 1 0],-b);            % LMI #1: -b
lmiterm([-1 2 2 0],eye(N));        % LMI #1: eye(n)

for i = 1:m,

    ei = [zeros(1, m)];
    ei(i) = 1;
    lmiterm([-i+1 1 1 x], ei, x, 's');

end

moindre=getlmis;

c = [1; zeros(m,1)]; options(5)=0;

tic [copt2,xopt2] = mincx(moindre,c,options); toc

xlmi2 = xopt2(2:m+1)

```

7.4 Moindres carrées, approche polytopique

```

Amin = (1-w_incertitude)*A(:); Amax = (1+w_incertitude)*A(:);

Amin = Amin(:); Amax = Amax(:);

L = 2^(m*N);

A_sommets = Amin;

for i=1:(m*N),

    A_sommets = [A_sommets,A_sommets];
    A_sommets(i,(2^(i-1)+1):(2^i)) = Amax(i)*ones(1,(2^(i-1)));

end

A_sommets = reshape(A_sommets, N, m, L);

```

```

setlmi3([]);

mu = lmivar(2,[1,1]); x = lmivar(2,[m,1]);

tic

for i = 1:m,

    ei = [zeros(1, m)];
    ei(i) = 1;
    lmiterm([-i 1 1 x], ei, x, 's');
end

fprintf('\n    debut LMI construction polytopique    ');

for i = 1:L

    lmiterm([- (i+m) 1 1 mu],                1,1);           % LMI #1: mu
    lmiterm([- (i+m) 2 1 x], A_sommets(:, :, i), 1);         % LMI #1: A*x
    lmiterm([- (i+m) 2 1 0],                -b);             % LMI #1: -b
    lmiterm([- (i+m) 2 2 0],                eye(N));         % LMI #1: eye(n)

    %lmiterm([ (i+m) 1 1 epsilon],           1,1);
    %lmiterm([ (i+m) 2 2 epsilon],           1,1);

    % fprintf(' %d', i);

end

fprintf('\n    fin LMI construction polytopique    ');

moindre=getlmi3;

tic

c = [1; zeros(m,1)];
options(1) = 1e-3;      %precision sur le cout optimal obtenu
options(3) = 10; options(5)=0;

tic [copt3,xopt3] = mincx(moindre,c,options); toc

xlmi3 = xopt3(2:m+1);

% Avec la solution obtenue sans prise en compte explicite des

```

```

% incertitudes, calcul du residu pire cas en presence des
% incertitudes.

residu_pq = 0;

for i = 1:size(A_sommets,3),

    residu_pq = max(residu_pq,
        sqrt((A_sommets(:, :, i)*xpq - b)'*(A_sommets(:, :, i)*xpq-b)));

end

fprintf('residu pire cas avec la solution ne prenant pas en compte les incertitudes :
        avec celle prennant en compte les incertitudes : %f \n', residu_pq, sqrt(copt);

```

7.5 Moindres carrés, approche alternative

```

% notation polycope 2002

Amin = (1-w_incertainitude)*A; Amax = (1+w_incertainitude)*A;

Anom = (Amax+Amin)/2;

B = []; C = [];

for j = 1:m,

    C_tmp = [];
    B_tmp = [];

    for i = 1:N,

        B_tmp = blkdiag( B_tmp, (Amax(i,j)-Amin(i,j))/2 ); % blkdiag
        C_tmp = [C_tmp ;
                1   ];
    end

    B = [B, B_tmp] ;
    C = blkdiag( C, C_tmp );

end

setlmis([]);

```

```

mu      = lmivar(2, [1,1]);
x       = lmivar(2, [m,1]);
[S, ndec] = lmivar(1, [ones(N*m,1), zeros(m*N,1)]);

tic

for i = 1:m,

    ei    = [zeros(1, m)];
    ei(i) = 1;
    lmiterm([-i 1 1 x], ei, x, 's');
end

lmiterm([- (m+1) 1 1 mu],      1 , 1);           % LMI #1: mu
lmiterm([- (m+1) 2 1 x], Anom , 1);           % LMI #1: A*x
lmiterm([- (m+1) 2 1 0],      -b);           % LMI #1: -b
lmiterm([- (m+1) 2 2 0],      1);           % LMI #1: eye(N)
lmiterm([- (m+1) 2 2 S],      -B , B'); lmiterm([- (m+1) 2 3 0],      B
);

lmiterm([- (m+1) 3 1 x],      C , 1);           % LMI #1: A*x
lmiterm([- (m+1) 3 3 S],      1 , 1);

moindre=getlmis;

toc

c = [1; zeros((ndec-1),1)];
options(1) = 1e-3;      %precision sur le cout optimal obtenu
options(3) = 1000; options(5)=0;

tic [copt4,xopt4] = mincx(moindre,c,options); toc

xlmi4 = xopt4(2:m+1);

Sopt = dec2mat(moindre, xopt4, S);

```


Bibliographie