

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Commande des systèmes**

Présenté par :

AOUDIA Akila

AIT ABDELKADER Djouher

Thème

Commande linéaire quadratique d'un système incertain.

Mémoire soutenu publiquement le 09/07/2017 devant le jury composé de :

Mr R.KARA

Pr, Ummto, Président

Mr A.MAIDI

Pr, Ummto, Encadreur

Mr M.TOUAT

MCB, Ummto, Examineur

Mr A.AKKOUCHE

MCB, Ummto, Examineur

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mr A.MAIDI, pour l'orientation, la confiance et la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

A nos parents pour nous 'avoir encouragées durant toutes ces années d'études Sans eux, on ne sera pas là aujourd'hui.

A nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce travail à ma mère ; ma grande source de bonheur et de motivation durant ce long parcours.

À mon père ; lui qui m'a toujours fait confiance dans les études.

Mon frère Zohier.

Mon frère Samir ainsi que son épouse Lila.

Mes sœurs : Djamila, Nadira, Djidji.

Mes neveux et nièces : Anis, Mahfoud, Oramdane, Julie, Mélissa, Zahra.

À ma chère cousine Samira et sa petite famille.

À la mémoire de mes grands parents ; que Dieu ait leurs âmes.

À ma binôme Akila.

À mes ami(s/es).

Djouher

Dédicaces

Je dédie ce modeste travail et ma profonde gratitude à ma mère et mon père pour l'éducation qu'ils m'ont prodigué; avec tout les moyens et aux pris de toutes les sacrifices qu'ils ont consentis à mon égard pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.

A mes frères et sœurs que j'aime beaucoup.

A ma grande famille.

Mes chers ami (e) s, et enseignants.

Tous ceux qui ont collaboré de près ou de loin à l'élaboration de ce travail.

Que Dieu leur accorde santé et prospérité.

A ma binôme :Djouher.

Akila.



Sommaire

Sommaire

Remerciements

Liste des figures

Liste des tableaux

Liste des symboles

Introduction générale.....1

Chapitre 1 : Commande quadratique

1.1	Introduction.....	2
1.2	Formulation du problème.....	2
1.3	Principe du minimum.....	4
1.4	Equation algébrique de Riccati.....	7
1.4.1	Résolution de l'équation de Riccati.....	9
1.4.1.1	Solution analytique de l'équation de Riccati.....	9
1.4.1.2	Cas de l'horizon infini.....	10
1.4.2	Exemple d'application.....	14
1.5	Conclusion.....	16

Chapitre 2 : Optimisation classique

2.1	Introduction.....	17
2.2	Optimisation sans contraintes.....	17
2.2.1	Résolution d'un problème d'optimisation sans contraintes.....	17
2.2.1.1	Méthodes analytique.....	18
2.2.1.2	Définitions.....	19
2.2.1.3	Méthode de résolution numérique.....	20
2.2.1.3.1	Méthode du Newton.....	20
2.2.1.3.2	Méthode du gradient (pas de décente).....	22
2.3	Exemple d'application.....	24
2.4	Les différentes méthodes d'optimisation.....	27

2.4.1	Méthodes déterministes.....	27
2.4.2	Méthodes stochastiques.....	28
2.4.2.1	Méthode du recuit simulé.....	28
2.4.2.2	Algorithme génétique.....	31
2.4.3	Méthodes hybrides.....	35
2.5.	Conclusion.....	39

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

3.1	Introduction.....	40
3.2	Description des incertitudes.....	40
3.2.1	Structuration d'incertitudes.....	40
3.2.2	Représentations des incertitudes.....	42
3.3	Contrôleur minimax en boucle fermée.....	44
3.4	Conclusion	52

Chapitre 4 : Commande LQ d'un système incertain

4.1	Introduction	54
4.2	Optimisation min-max dans le cas de la commande LQ.....	54
4.3	Commande LQ d'un système incertain.....	57
4.3.1	Modélisation du système :	58
4.3.2	Commande LQ.....	61
4.3.3	Solution du problem.....	61
4.3.4	Résultats de simulation	64
4.4	Conclusion	69
	Conclusion générale	70

Liste des figures :

Figure 2.1 : Interprétation géométrique.....	22
Figure 2.2 : Méthode de Tunnelier.....	37
Figure 3.1 : Incertitudes additives.....	42
Figure 3.2 : Incertitudes multiplicatives en entrée.....	43
Figure 3.3 : Incertitudes multiplicatives en sortie.....	43
Figure 3.4 : Système masse ressort amortisseur.....	48
Figure 3.5 : Critère du coût quadratique.....	51
Figure 3.6 : Inverse du négatif de J.....	52
Figure 4.1 : Cube incertain.....	55
Figure 4.2 : Système deux masses ressort amortisseur.....	57
Figure 4.3 : Méthode des intégrateurs.....	60
Figure 4.4 : Position de la masse m1.	65
Figure 4.5 : Vitesse de la masse m1.....	66
Figure 4.6 : Position de la masse m2.	66
Figure 4.7 : Vitesse de la masse m2.	67
Figure 4.8 : Position de la masse m1.	67
Figure 4.9 : Vitesse de la masse m1.....	68
Figure 4.10 : Position de la masse m2.	68
Figure 4.11 : Vitesse de la masse m2.....	69

Liste des symboles :

\mathbb{R} : Ensemble des nombre réels.

$\lambda(t)$: Vecteur des variables adjointes.

t : Variable du temps.

t_0 : Instant initial.

t_f : Instant final.

$x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$: Vecteur d'état.

x_0 : État initial.

x_f : État final.

$u = (u_1, u, \dots, u_m)^T \in \mathbb{R}^m$: Vecteur de commande.

x^T : Transposée de x .

$\dot{x} = \frac{dx}{dt}$: Dérivée temporelle.

$\frac{\partial x}{\partial t}$: Dérivée partielle du premier ordre par rapport à t .

H : Fonction hamiltonienne.

| Ou t.q : Tel que.

max: Maximum.

min : Minimum.

lim : Limite.

inf : Borne inférieure.

sup : Borne supérieure.

$\| \cdot \|$: Norme euclidienne classique sur \mathbb{R}^n .

$K(t)$: Matrice des gains.

tr : Trace.

J : Critère quadratique.

J_u : Critère de performance lié à l'écart sur la commande.

J_x : Critère de performance lié à l'écart état-consigne.

δJ : Variation de J.

$[a, b]$: Intervalle fermé de \mathbb{R} d'extrémités a et b .

$f(x)$: Fonction dans \mathbb{R} .

$\nabla_x f(x)$: Le gradient de f par rapport à x .

$\nabla_x^2 f(x)$: Le gradient carré de f par rapport à x .

K^{nom} : Gain de retour d'état nominal.

K_{r_opt} : Gain de retour d'état optimal.

Liste des tableaux :

Tableau 4.1 : Les différentes possibilités pour les incertitudes.....63

Tableau 4.2 : Gain du retour d'état et valeur du critère pour chaque incertitude.....64



Introduction générale

De nos jours, l'automatique répond à plusieurs exigences de l'être humain, elle est présente dans la totalité des domaines. Son rythme de progression ne cesse d'évaluer de jours en jours, jusqu'à ce qu'elle fait partie de la vie quotidienne de l'homme.

De nombreux progrès ont été accomplis récemment dans le domaine de l'automatique des systèmes linéaires. La plupart des techniques d'analyse et de synthèse sont basées sur des modèles linéaires des procédés commandés, les techniques de commande avancée deviennent l'un des domaines de recherche les plus actifs. Notre travail consiste à l'étude et la conception d'une commande linéaire quadratique d'un système incertain en le formulant comme un problème d'optimisation classique du type min-max.

Le contenu du mémoire est subdivisé en quatre chapitres :

Le premier chapitre traite de la commande linéaire quadratique.

Dans le deuxième chapitre nous présentons des généralités sur l'optimisation classique.

Dans le troisième chapitre, nous présentons quelques notions sur les systèmes incertains et la formulation d'un problème de commande optimale sous forme d'optimisation classique du type min-max.

Le quatrième chapitre présente une méthode de synthèse d'un retour d'état optimal pour un système incertain ; illustrée par une application à un système mécanique.

Enfin nous terminons le mémoire par une conclusion générale.



Chapitre 1

1.1. Introduction

En automatique la commande linéaire quadratique dite commande LQ, est une commande optimale par retour d'état. Cette méthode permet de calculer la matrice de gain assurant une poursuite et un rejet de perturbation. L'idée consiste à minimiser un critère de performances quadratique traduisant les objectifs de poursuite et de l'optimisation de l'énergie. La commande LQ permet de réaliser un placement optimal des pôles en boucle fermée.

Ce chapitre est consacré à la commande LQ.

1.2. Formulation du problème

L'évolution du processus est décrite par la représentation d'état suivante :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (1.2)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (1.3)$$

Où \mathbf{u} est le vecteur de commande de dimension m_u , \mathbf{x} le vecteur d'état de dimension n , \mathbf{y} le vecteur de sortie de dimension n_y . \mathbf{A} , \mathbf{B} , \mathbf{C} sont des matrices de tailles respectives $n \times n$, $n \times m_u$, $n_y \times n$, et \mathbf{D} une matrice de transmission directe ou de couplage de dimension $n_y \times m_u$.

L'existence d'une matrice \mathbf{D} , non nulle correspond à une transmission directe d'information de l'entrée vers la sortie et pour beaucoup de modèle de processus $\mathbf{D} = 0$.

Le critère à optimiser dans le cas d'une commande LQ est donné comme suit :

$$\begin{aligned} J = & \frac{1}{2} x^T(t_f) Q_f x(t_f) \\ & + \frac{1}{2} \int_{t_0}^{t_f} \left[(x^d(t) - x(t))^T Q (x^d(t) - x(t)) \right. \\ & \left. + u^T(t) R u(t) \right] dt \end{aligned} \quad (1.4)$$

$$= J_x + J_u + J_f \quad (1.5)$$

Où

$$J_x = \frac{1}{2} \int_{t_0}^{t_f} (x^d(t) - x(t))^T Q (x^d(t) - x(t)) dt \quad (1.6)$$

La relation (1.6) est liée à la nature du régime transitoire obtenu par la minimisation de la distance existant entre la consigne $x^d(t)$ et l'état $x(t)$.

$$J_u = \frac{1}{2} \int_{t_0}^{t_f} u^T(t) R u(t) dt \quad (1.7)$$

L'équation (1.7) est liée à la minimisation de l'énergie de commande.

$$J_f = \frac{1}{2} x^T(t_f) Q_f x(t_f) \quad (1.8)$$

L'équation (1.8) permet de réduire l'écart final consigne-sortie[1].

Posons $x^d(t) = 0$

La résolution du problème LQ consiste à rechercher la matrice des gains $K(t)$, telle que la commande par retour d'état $u(t) = -K(t)x(t)$ stabilise le système et minimise le critère quadratique $J(u)$, c'est-à-dire

$$\min_{\mathbf{u}(t) \in \mathbb{R}^m} \mathcal{J}(\mathbf{u}) = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{Q}_f \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} [\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t)] dt \quad (1.9)$$

Où \mathbf{Q} et \mathbf{Q}_f sont des matrices symétriques semi-définies positives tandis que \mathbf{R} est une matrice symétrique définie positive.

Remarque 1 :

Le problème de la commande linéaire quadratique à horizon infini défini par la minimisation du critère :

$$\min_{\mathbf{u}(t) \in \mathbb{R}^m} \mathcal{J}(\mathbf{u}) = \frac{1}{2} \int_{t_0}^{\infty} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) dt \quad (1.10)$$

$$t_f \rightarrow \infty : \text{pas de coût terminal } \lim_{t_f \rightarrow \infty} \mathbf{x}^T(t_f) \mathbf{Q} \mathbf{x}(t_f) = \mathbf{0} \blacksquare$$

La théorie de contrôle moderne se présente comme la synthèse de la matrice de gain du contrôle par retour d'état. Le contrôle intervient dans l'équation d'état comme une donnée.

Afin de déterminer la solution d'un problème du contrôle optimal, il existe plusieurs approches dont deux sont principales, la première, constituée de la théorie de Hamilton-Jacobi, consiste à établir des conditions suffisantes d'optimalité global et la deuxième se base sur le principe de minimum de Pontriaguine qui est un principe de variation, le contrôle optimal doit minimiser une fonction appelée Hamiltonien[2].

1.3. Principe du minimum

Le principe du minimum de Pontriaguine représente le résultat central de la théorie du contrôle optimal qui est une généralisation du calcul des variations.

Considérons un système défini par les équations d'état :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{u} \in \mathbf{U} \end{cases} \quad (1.11)$$

Chapitre 1 : Commande quadratique

L'objectif est de trouver la commande u qui minimise une fonction coût, c'est-à-dire

$$\min_{u(t)} \mathcal{J}(u(t)) = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} \psi(x(t), u(t), t) dt \quad (1.12)$$

Sujet à :

$$\dot{x}(t) = f(x(t), u(t), t)$$

$$x(t_0) = x_0$$

$$x(t_f) = x_f \text{ ou libre}$$

On définit l'Hamiltonien du système :

$$H(x(t), u(t), \lambda(t), t) = \psi(x(t), u(t), t) + \lambda(t)^T f(x(t), u(t), t) \quad (1.13)$$

Où $\lambda(t)$ est le vecteur des variables adjointes de dimension n .

Le principe du minimum de Pontriaguine énonce que la trajectoire optimale minimise l'Hamiltonien du système.

Le critère quadratique à minimiser devient :

$$\mathcal{J}(u(t)) = \varphi(x(t_f), t_f) + \int_{t_0}^{t_f} H(x(t), u(t), \lambda(t), t) - \lambda^T(t) \dot{x}(t) dt \quad (1.14)$$

Le long de la trajectoire optimale, on dispose d'un certain nombre d'équations permettant de résoudre le problème de commande optimale. Ces équations sont généralement établies en utilisant le calcul des variations [3]. Le but est d'avoir la variation de J est nulle, i.e.

$$\delta J = 0$$

Pour calculer δJ , on doit calculer la variation :

$$\begin{aligned}\Delta J &= J(x + \delta x, u + \delta u, \lambda + \delta \lambda, t) - J(x, u, \lambda, t) \\ &= \delta J \delta X + g u \delta X u\end{aligned}\tag{1.15}$$

Tel que :

$$X = (\delta x, \delta u, \delta \lambda)$$

En annulant δJ , on obtient les conditions d'optimalités suivantes :

$$\nabla_u H(x(t), u(t), \lambda(t), t) = 0\tag{1.16}$$

$$\dot{x}(t) = \nabla_x H(x(t), u(t), \lambda(t), t)\tag{1.17}$$

$$\dot{\lambda}(t) = -\nabla_x H(x(t), u(t), \lambda(t), t)\tag{1.18}$$

Pour définir la nature de la solution, on étudie la matrice Hessienne :

$$\nabla_x^2 H(x(t), u(t), \lambda(t), t)\tag{1.19}$$

Pour résoudre les équations différentielles (1.17) et (1.18), on doit disposer des conditions aux limites. Ainsi, en plus de la condition initiale $x(t_0)$, on doit utiliser la condition suivante

$$\delta^T x(t_f) \left[\lambda(t_f) - \frac{\partial \varphi(x(t_f), t_f)}{\partial x(t_f)} \right] = 0 \quad (1.20)$$

L'équation (1.20) est dite équation de transversalité[4].

1.4. Équation algébrique de Riccati

On considère le cas de la commande linéaire quadratique lorsque le temps d'optimisation devient infini (commande à horizon infini).

Soit un système d'équations différentielles à coefficients constants

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ x(t_0) = x_0 \end{cases} \quad (1.21)$$

Où $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ et $u \in \mathbb{R}^m$. On définit sur un horizon infini, une fonction coût purement quadratique

$$J(u) = \frac{1}{2} x^T(t_f) K_f x(t_f) + \frac{1}{2} \int_{t_0}^{\infty} x^T(t) Q x(t) + u^T(t) R u(t) dt \quad (1.22)$$

Où $x(t_f) \in \mathbb{R}^n$ est fixé, K_f et Q sont des matrices constantes symétriques d'ordre $n \times n$ et à coefficients réels avec R semi-définie positive [5].

La résolution du problème linéaire quadratique revient à déterminer $u(t)$ qui minimise le critère (1.22)

On appliquant le principe du minimum

$$H = \frac{1}{2} \left(x^T(t) Q x(t) + u^T(t) R u(t) \right) + \lambda^T(t) (Ax(t) + Bu(t)) \quad (1.23)$$

La condition d'optimalité (1.16) donne :

$$Ru(t) + B^T \lambda(t) = 0$$

D'où l'expression de $u(t)$:

$$u(t) = -R^{-1}B^T\lambda(t) \quad (1.24)$$

La condition (1.17) donne l'équation d'état du système en boucle fermée :

$$\dot{x}(t) = Ax(t) - BR^{-1}B^T\lambda(t) \quad (1.25)$$

Et la condition (1.18) qui donne le système adjoint :

$$\dot{\lambda}(t) = -(Qx(t) + A^T\lambda(t)) \quad (1.26)$$

En rassemblant l'ensemble de ces résultats, le système à résoudre devient :

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \quad (1.27)$$

Pour concevoir un retour d'état, on doit prendre :

$$\lambda(t) = K(t)x(t) - V(t) \quad (1.28)$$

Le problème revient à déterminer $K(t)$ et $V(t)$

En décrivant $\lambda(t)$, il devient :

$$\dot{\lambda}(t) = \dot{K}(t)x(t) + K(t)\dot{x}(t) - \dot{V}(t) \quad (1.29)$$

D'après (1.25), on a :

$$\dot{x}(t) = Ax(t) - BR^{-1}B^T(K(t)x(t) - V(t))$$

$$\dot{x}(t) = (A - BR^{-1}B^TK(t))x(t) + BR^{-1}B^TV \quad (1.30)$$

Chapitre 1 : Commande quadratique

L'équation (1.26) peut s'écrire comme suit :

$$\dot{\lambda}(t) = -(A^T K(t) + Q)x(t) + A^T V \quad (1.31)$$

D'après les expressions (1.29), (1.30) on obtient :

$$\dot{\lambda}(t) = \dot{K}(t)x(t) + K(t) \left(A - BR^{-1}B^T K(t) \right) x + K(t)BR^{-1}B^T V(t) - \dot{V}(t) \quad (1.32)$$

En égalisant les deux équations (1.31) = (1.32), on obtient :

$$\begin{aligned} \dot{K}(t)x(t) + K(t)Ax(t) - K(t)BR^{-1}B^T K(t)x(t) + K(t)BR^{-1}B^T V(t) - \dot{V}(t) \\ = -Qx(t) - A^T K(t)x(t) + A^T V(t) \end{aligned} \quad (1.33)$$

$$\begin{aligned} [\dot{K}(t) + K(t)A + A^T K(t) - K(t)BR^{-1}B^T K(t) + Q]x(t) - \dot{V}(t) \\ + (K(t)BR^{-1}B^T - A^T)V(t) = 0 \end{aligned} \quad (1.34)$$

Pour que cette équation soit vérifiée $\forall x(t)$, on doit imposer :

$$\dot{K}(t) + K(t)A + A^T K(t) - K(t)BR^{-1}B^T K(t) + Q = 0 \quad (1.35)$$

$$-\dot{V}(t) + (K(t)BR^{-1}B^T - A^T)V(t) = 0 \quad (1.36)$$

Nous proposons d'étudier ce problème lorsque le temps d'optimisation devient infini.

Pour tout $K_f \geq 0$, $K(t_f) \rightarrow K$ lorsque $t_f \rightarrow \infty$, ou K est l'unique solution semi-définie positive de l'équation algébrique du type Riccati [4]

$$KA + A^T K - KBR^{-1}B^T K + Q = 0 \quad (1.37)$$

1.4.1. Résolution de l'équation de Riccati

Chapitre 1 : Commande quadratique

L'équation matricielle de Riccati peut être résolue, soit analytiquement dans les cas les plus simples, soit par les méthodes d'intégrations [6].

1.4.1.1. Solution analytique de l'équation de Riccati :

Soit l'équation différentielle suivante

$$\dot{X} = AX + XB + XCX + D \quad (1.38)$$

La résolution de l'équation matricielle (1.38) dans laquelle les matrices A, B, C et D peuvent être variables, et équivalente à celle du système linéaire [6]

$$\dot{Y} = -BY - CZ \quad (1.39)$$

$$\dot{Z} = DY + AZ \quad (1.40)$$

Avec

$$X = ZY^{-1} \quad (1.41)$$

On le vérifie facilement par simple substitution. Nous avons en premier lieu :

$$\dot{X} = \dot{Z}Y^{-1} + Z\dot{Y}^{-1}$$

Mais

$$\dot{Y}^{-1} = -Y^{-1}\dot{Y}Y^{-1}$$

D'où

$$\begin{aligned} \dot{X} &= (DY + AZ)Y^{-1} + Z(-Y^{-1}\dot{Y}Y^{-1})(-BY - CZ)Y^{-1} \\ &= D + AX + XB + XCX \end{aligned}$$

1.4.1.2. Cas de l'horizon infini

Soit le problème défini par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \quad (1.42)$$

$$J = \frac{1}{2} \int_0^{t_f} (\eta x_1^2 + \mu x_2^2 + \lambda u^2) dt \quad (1.43)$$

L'équation matricielle d'état s'écrit :

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (1.44)$$

Les matrices dont nous avons besoin sont les suivantes :

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1.45)$$

$$2q = \eta x_1^2 + \mu x_2^2 + \lambda u^2 \rightarrow M = \begin{bmatrix} \eta & 0 \\ 0 & \mu \end{bmatrix} \quad (1.46)$$

$$R = \lambda, \quad S = BR^{-1}B^T \quad (1.47)$$

D'après l'équation (1.47) on obtient :

$$S = \frac{BB^T}{\lambda} = \frac{1}{\lambda} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.48)$$

Posons :

$$K = \begin{bmatrix} a & b \\ b & d \end{bmatrix} \quad (1.49)$$

Calculons alors :

$$KA = \begin{bmatrix} 0 & a \\ 0 & b \end{bmatrix} \quad (1.50)$$

$$A^T K = (KA)^T = \begin{bmatrix} 0 & 0 \\ a & b \end{bmatrix} \quad (1.51)$$

$$KSK = \frac{1}{\lambda} \begin{bmatrix} b^2 & bd \\ bd & d^2 \end{bmatrix} \quad (1.52)$$

L'équation de Riccati s'écrit donc :

$$\begin{bmatrix} \dot{a} & \dot{b} \\ \dot{b} & \dot{d} \end{bmatrix} + \begin{bmatrix} 0 & a \\ 0 & b \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ a & b \end{bmatrix} - \frac{1}{\lambda} \begin{bmatrix} b^2 & bd \\ bd & d^2 \end{bmatrix} + \begin{bmatrix} \eta & 0 \\ 0 & \mu \end{bmatrix} = 0 \quad (1.53)$$

Chapitre 1 : Commande quadratique

Elle est équivalente aux trois équations différentielles suivantes :

$$\dot{a} - \frac{b^2}{\lambda} + \eta = 0 \quad (1.54)$$

$$\dot{b} + a - \frac{bd}{\lambda} = 0 \quad (1.55)$$

$$\dot{d} + 2b - \frac{d^2}{\lambda} + \mu = 0 \quad (1.56)$$

Le calcul se présente ensuite de manière différente selon que l'horizon est fini ou infini, mais dans notre cas on s'intéresse à l'horizon infini pour résoudre l'équation algébrique.

Nous avons alors $\dot{a} = \dot{b} = \dot{d} = 0$. Le système algébrique qui en résulte a pour solution :

$$b^2 = \lambda\eta \quad (1.57)$$

$$d^2 = \lambda(2\sqrt{\lambda\eta} + \mu) \quad (1.58)$$

$$a^2 = \eta(2\sqrt{\lambda\eta} + \mu) \quad (1.59)$$

La commande optimale est donnée par :

$$u = -R^{-1}B^TKx \quad (1.60)$$

Or

$$B^T K = [0 \quad 1] \begin{bmatrix} a & b \\ b & d \end{bmatrix} = [b \quad d] \quad (1.61)$$

D'où

$$u = -\frac{1}{\lambda}(bx_1 + dx_2) \quad (1.62)$$

1.4.2. Exemple d'application

Soit le problème de commande optimale suivant [4] :

$$\min_{u(t)} J(u(t)) = \frac{1}{2} \int_0^{\infty} [\rho^2 x_1^2(t) + u^2(t)] dt, \text{ avec } \rho > 0$$

Sujet à :

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = u(t)$$

Donner et résoudre l'équation de Riccati correspondante :

Comme $t \rightarrow \infty$ (*horizon infini*), alors $\dot{K} = 0$

$$A^T K + KA - KSK + Q = 0, \quad S = BR^{-1}B^T$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, Q = \begin{bmatrix} \rho^2 & 0 \\ 0 & 0 \end{bmatrix}, R = 1$$

$$S = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, K = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} + \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} + \begin{bmatrix} \rho^2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ a & b \end{bmatrix} + \begin{bmatrix} 0 & a \\ 0 & b \end{bmatrix} - \begin{bmatrix} 0 & b \\ 0 & c \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} + \begin{bmatrix} \rho^2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & a \\ a & 2b \end{bmatrix} - \begin{bmatrix} b^2 & bc \\ bc & c^2 \end{bmatrix} + \begin{bmatrix} \rho^2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$-b^2 + \rho^2 = 0 \tag{1.63}$$

$$a - bc = 0 \tag{1.64}$$

$$2b - c^2 = 0 \tag{1.65}$$

$$(1.63) \Rightarrow b^2 = \rho^2 \Rightarrow b = \rho \quad (\rho > 0)$$

$$(1.65) \Rightarrow c^2 = 2b = 2\rho \Rightarrow c = \pm\sqrt{2\rho}$$

$$(1.64) \Rightarrow a = bc \Rightarrow a = \rho\sqrt{2\rho}, (K > 0 \Rightarrow a > 0)$$

La solution de l'équation de Riccati est :

$$K = \begin{bmatrix} \rho\sqrt{2\rho} & \rho \\ \rho & \sqrt{2\rho} \end{bmatrix}$$

1.5 Conclusion

Dans ce chapitre, nous avons présenté la commande linéaire quadratique. Cette méthode permet de minimiser une fonction coût traduisant deux objectifs : poursuite et énergie. Pour déterminer la solution, nous avons appliqué le principe du minimum de Pontriaguine qui nous a conduits à l'équation algébrique de Riccati.

Après avoir mis en évidence la commande quadratique, nous présentons dans le chapitre suivant quelques notions sur l'optimisation classique.



Chapitre 2

Chapitre 2 : Optimisation classique

2.1 Introduction :

En générale optimiser signifie le fait de chercher un certains points particuliers d'une fonction. Une fonction $f(x)$ continue et définie dans l'intervalle $[a, b]$ admet certains points particuliers appelés **optimum**. Un optimum peut être un minimum ou un maximum. Pour chaque optimum, on distingue un optimum global (absolu) et un optimum local (relatif).

Dans ce chapitre, nous allons présenter des généralités sur l'optimisation classique.

2.2 Optimisation sans contraintes :

Soit une fonction $f(x)$ supposée continue et différentiable, le problème d'optimisation de manière générale s'écrit sous la forme suivante [4]:

$$\text{Opt}_x f(x)$$

Sujet à :

$$g_i(x) = 0, \quad i = 1 \dots p$$

$$h_j(x) \leq 0, \quad j = 1 \dots q$$

Dans le cas d'un problème d'optimisation sans contraintes on a $p = q = 0$. Donc on cherche à résoudre le problème d'optimisation sans contraintes.

2.2.1 Résolution d'un problème d'optimisation sans contraintes :

Pour résoudre un problème d'optimisation sans contraintes, on utilise deux méthodes:

Chapitre 2 : Optimisation classique

2.2.1.1 Méthodes analytique :

Lemme 1 (conditions du premier ordre pour la présence d'un extremum).

Si la fonction présente un extremum en \mathbf{x}_0 , alors

$$\nabla_x f(\mathbf{x}) = \mathbf{0}. \quad (2.1)$$

Cette condition de premier ordre n'est pas suffisante, car la connaissance des dérivés d'ordre supérieurs sont nécessaires pour conclure à la présence d'un extremum et la détermination du type d'extrémum (minimum ou maximum).

Lemme 2 :(conditions suffisantes d'existence d'un extrémum).

Pour conclure l'existence et la nature d'un point critique, on doit étudier la matrice Hessienne représentée sous la forme suivante :

$$\nabla_x^2 f(\mathbf{x}) = \frac{\partial^2 f}{\partial \mathbf{x}^2}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \dots & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & \dots & \dots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{pmatrix} \quad (2.2)$$

Pour étudier la nature du point critique, on a quatre cas possibles :

❖ Cas où $\nabla_x f(\mathbf{x}) = \mathbf{0}$ et $\nabla_x^2 f(\mathbf{x}) > 0$.

Si toutes les valeurs propres sont positives ou tout les déterminants pris en chaine de la matrice hessienne sont positifs alors le point critique c'est un minimum.

Chapitre 2 : Optimisation classique

❖ Cas où $\nabla_x f(x) = \mathbf{0}$ et $\nabla_x^2 f(x) < \mathbf{0}$.

Si toutes les valeurs propres sont négatives et tous les déterminants pris en chaîne sont négatifs alors le point critique est un maximum.

❖ Cas où $\nabla_x f(x) = \mathbf{0}$ et $\nabla_x^2 f(x) \leq \mathbf{0}$.

Si une des valeurs des déterminants égales à zéro, et les autres sont négatives ou positives alors le point critique est un point singulier.

❖ Dans le cas où on trouve des valeurs négatives et des valeurs positives le point critique est un point selle [4].

2.2.1.2 Définitions :

On a $x_0 \in S$.

- On dit que x_0 est un minimum local si :

$$\exists V \in V(x_0) \text{ tel que } \forall x \in V, f(x) \geq f(x_0)$$

- On dit que x_0 est un minimum local strict si :

$$\exists V \in V(x_0) \text{ tel que } \forall x \in V, f(x) > f(x_0)$$

- On dit que x_0 est un minimum global si :

$$\forall x \in S, f(x) \geq f(x_0)$$

- On dit que x_0 est un minimum global strict si :

$$\forall x \in S, f(x) > f(x_0)$$

Chapitre 2 : Optimisation classique

- Si $f(x)$ admet plusieurs minimums, alors le minimum global est celui qui correspond à la valeur minimale de la fonction.
- Un problème de minimisation peut être ramené à un problème de maximisation.

$$\min f(x) = -\max[-f(x)]$$

- Si $\forall x \in D, \nabla_x^2 f(x) > 0$, alors $f(x)$ est convexe.
- Si $\forall x \in D, \nabla_x^2 f(x) < 0$, alors $f(x)$ est concave [7].

2.2.1.3 Méthode de résolution numérique :

Dans le cas où on ne peut pas résoudre les problèmes d'optimisation par la méthode analytique on fait appel à des méthodes numériques.

2.2.1.3.1 Méthode du Newton :

La méthode de Newton n'est pas une méthode d'optimisation à proprement parler. C'est en réalité une méthode utilisée pour résoudre des équations non linéaires de la forme $F(x) = 0$ où F est une fonction \mathbb{R}^n dans \mathbb{R}^n . Nous allons d'abord la décrire puis montrer comment on peut l'appliquer à la recherche du minimum [8]

$$x_{n+1} = x_n - \frac{\nabla x_n f(x_n)}{\nabla_x^2 f(x_n)}, \quad \forall n \geq 0 \quad (2.3)$$

Chapitre 2 : Optimisation classique

Algorithme de la méthode de Newton :

i) Etant donné, ε , un critère d'arrêt.

Où $\varepsilon \in \mathbb{R}^+$ représentent des erreurs d'approximation caractérisant la qualité de la solution numérique. Dans tout les cas il se peut que le critère d'arrêt soit vérifié en des points ne correspondant pas à des solutions de l'équation à résoudre.

ii) Etant donné N , le nombre maximal d'itérations.

iii) Etant donné x_0 , une valeur initiale de la solution.

iv) Effectuer :
$$x_{(n+1)} = x_n - \frac{\nabla x_n f(x_n)}{\nabla^2_{x_n} f(x_n)}$$

v) Si $\left| \frac{x_{(n+1)} - x_n}{x_{n+1}} \right| < \varepsilon :$

- Convergence atteinte.
- Ecrire la solution x_{n+1} .
- Arrêt.

vi) Si le nombre d'itérations N est atteint.

- Convergence non atteinte en N itérations.
- Arrêt.

vii) Retour a l'étape (iv).

Remarquons qu'il faut non seulement assurer le convergence de la suite x_n vers la solution x_* , mais aussi monter que cette suite est bien définie, c'est-à-dire montrer que $f'(x_n) \neq \mathbf{0}$ [10].

Chapitre 2 : Optimisation classique

Cette méthode est aussi appelée **méthode de la tangente**. En effet chaque itéré x_{n+1} est obtenu à partir du précédent en traçant la tangente à la courbe de f au point $(x_n, f(x_n))$ et en prenant son intersection avec l'axe des abscisses (voir la Figure 2.1) [8].

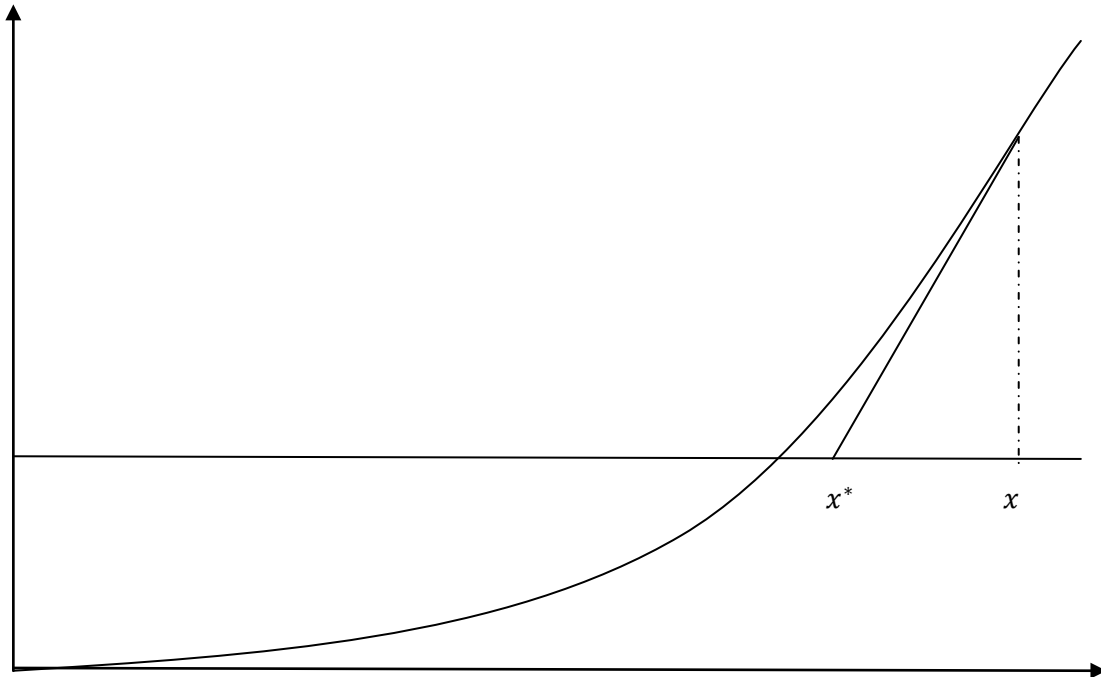


Figure 2.1 Interprétation géométrique de la méthode de Newton.

L'inconvénient majeur de cette méthode est sa sensibilité au choix du point de départ : la convergence est **locale**. Si le point est mal choisi (« trop loin » de la solution), la méthode peut diverger ou converger vers une autre solution [8].

2.2.1.3.2 Méthode du gradient (pas de décente):

La méthode du gradient fait partie d'une classe plus grande de méthode numérique appelée **méthode de décente**. Expliquons rapidement l'idée directrice de ces méthodes

On veut minimiser une fonction f . Pour cela on se donne un point de départ arbitraire x_0 . Pour construire l'itéré suivant x_1 , il faut penser qu'on veut se rapprocher du minimum de f ;

Chapitre 2 : Optimisation classique

on veut donc que $f(x_1) < f(x_0)$. On cherche alors x_1 sous la forme $x_1 = x_0 - \alpha_0 \nabla f(x_0)$ [3].

Les algorithmes d'optimisation sont généralement écrits pour minimiser une fonction. Si l'on désire maximiser une fonction, il suffira de minimiser son opposé [9].

L'algorithme :

Soient \mathbb{E} un espace hilbertien (produit scalaire noté $\langle \cdot, \cdot \rangle$ et norme associée notée $\| \cdot \|$) et $x \in \mathbb{E} \mapsto f(x) \in \mathbb{R}$ une fonction différentiable. On note $f'(x)$ et $\nabla f(x)$, la dérivée et le gradient de f en x , si bien que pour tout $d \in \mathbb{E}$, $f'(x) \cdot d = \langle \nabla f(x), d \rangle$.

Algorithme du gradient :

On se donne un point/ itéré initial $x_0 \in \mathbb{E}$ et un seuil de tolérance $\varepsilon \geq 0$. L'algorithme du gradient définit une suite d'itérés $x_1, x_2, \dots \in \mathbb{E}$ jusqu'à ce qu'un test d'arrêt soit satisfait. Il passe de x_k à x_{k+1} par les étapes suivantes [9]:

- i) Simulation : calcul de $\nabla f(x_k)$.
- ii) Test d'arrêt : si $\|\nabla f(x_k)\| \leq \varepsilon$, arrêt.
- iii) Calcul de pas $\alpha_k > 0$ par une règle de recherche linéaire sur f en x_k le long de la direction $-\nabla f(x_k)$.
- iv) Nouvel itéré :
$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{2.4}$$

- L'algorithme peut nécessiter de nombreuses itérations pour converger vers un minimum local, notamment si la courbure est très différente dans des directions différentes
- La recherche du pas α optimal, généralement effectuée par une recherche linéaire, peut se révéler très longue. Inversement, utiliser un pas α fixe peut conduire à de mauvais résultats. Des méthodes comme la méthode de Newton et l'inversion de la matrice hessienne en complément des techniques de gradient conjugué offrent souvent de meilleurs résultats.

2.3. Exemple d'application :

Méthode analytique :

On a $f(x) = x_1^2 + x_2^2$

-les points critiques de cette fonction sont :

$$\nabla_x f(x) = \mathbf{0} \Rightarrow \nabla_x f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = \mathbf{0}$$

$$2x_1 = 0$$

$$2x_2 = 0 \Rightarrow x_1 = 0 \text{ et } x_2 = 0$$

$x = (0, 0)$ est le seul point critique

- **Nature du point :**

$$\nabla_x^2 f(x) = \frac{\partial^2 f}{\partial x^2}(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \dots & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \dots & \dots & \frac{\partial^2 f}{\partial x_2^2}(x) \end{pmatrix}$$

$$\nabla_x^2 f(x) = \begin{pmatrix} \frac{\partial(2x_1)}{\partial x_1} & \frac{\partial(2x_1)}{\partial x_2} \\ \frac{\partial(2x_2)}{\partial x_1} & \frac{\partial(2x_2)}{\partial x_2} \end{pmatrix}$$

$$\nabla_x^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Méthode des déterminants pris en chaîne :

$$\nabla_x^2 f(x) = \begin{bmatrix} |2| & \mathbf{0} \\ \mathbf{0} & |2| \end{bmatrix} \Rightarrow |2| = 2 > 0 \quad \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0$$

Alors le point critique est un minimum [4].

Méthodes numérique

Exemple : on a la fonction suivante [4]:

$$f(x) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

Pour déterminer le minimum de cette fonction on propose d'utiliser la méthode de Newton.

En considérant l'estimé de départ :

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Méthode de newton

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[\nabla_{\mathbf{x}}^2 (f(\mathbf{x}^{(k)})) \right]^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}^{(k)})$$

Gradient de la fonction $f(\mathbf{x})$:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{bmatrix}$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} 1 + 4x_1^{(k)} + 2x_2^{(k)} \\ -1 + 2x_1^{(k)} + 2x_2^{(k)} \end{bmatrix}$$

Matrice Hésienne

$$\nabla_{\mathbf{x}}^2 f(\mathbf{x}) = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

Avant d'appliquer la méthode de Newton, il est judicieux d'étudier la nature de la fonction $f(\mathbf{x})$. Ainsi le problème admet une solution si la matrice Hésienne est définie positive ($f(\mathbf{x})$ convexe). On constate que la matrice Hésienne ne dépend pas de \mathbf{x} alors la fonction $f(\mathbf{x})$ peut être convexe ou concave.

$$|4| = 4 > 0$$

Chapitre 2 : Optimisation classique

$$\begin{vmatrix} 4 & 2 \\ 2 & 2 \end{vmatrix} = 4 > 0$$

$\nabla_x f(x)$ est définie positive quelque soit x , alors la fonction $f(x)$ est convexe et admet un seul minimum.

- Application de la méthode de Newton :

$$\left[\nabla_x^2 (f(x^k)) \right]^{-1} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$$

k	$x^{(k)}$	$\left[\nabla_x^2 (f(x^k)) \right]^{-1}$	$\nabla_x f(x^{(k)})$	$x^{(k+1)}$
0	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$
1	$\begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$	$\begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$

On remarque que $\nabla_x f(x^{(1)}) = \mathbf{0}$, alors le minimum est atteint donc la solution obtenue par la méthode de Newton est [4] :

$$x_1 = -1, x_2 = 1.5$$

2.4 Les différentes méthodes d'optimisation :

2.4.1 Méthodes déterministes :

Nous allons présenter quelques algorithmes permettant de calculer (de manière approchée) le minimum de la fonction (f) de départ. Nous présentons les méthodes « de base » les plus classiques. Toutefois, la plupart de ces algorithmes exploitent les conditions d'optimalité dont on a vu qu'elles permettaient (au mieux) de déterminer des minima **locaux** [8].

Chapitre 2 : Optimisation classique

Les deux méthodes les plus utilisées sont la méthode de Newton et la méthode du Gradient, que nous avons bien présenté dans les méthodes numérique.

2.4.2 Méthode stochastique :

Les recherches d'un minimum par exploration aléatoire du domaine de définition \mathbb{E} sont souvent simples à mettre en place pour des fonctions aux propriétés variées, qu'elles soient peu nécessairement, dotées d'une multitude de minima locaux. La vitesse de convergence est souvent difficile à estimer mais dans la plupart le temps généralement est faible (d'autant plus faible que l'approximation \mathbf{x}_k est proche du minimum \mathbf{x}_0), au contraire de certaines méthodes utilisant la différentiabilité (par exemple la méthode de Newton). La programmation est aisée et rapide, basée uniquement sur la fonction f (pas de calcul de gradient, exact ou approché, ni a fortiori de hessienne) et la simulation de variables aléatoires $\mathbf{x}_1, \dots, \mathbf{x}_k \dots$ convergent vers un point de minimum et telles que la suite des valeurs $f \circ \mathbf{x}_1, \dots, f \circ \mathbf{x}_k$ converge en décroissant vers la valeur minimum f_* qui sera le minimum global. Cet algorithme n'a pas une grande garantie d'efficacité ou de précision dû à l'absence de test d'arrêt pour la suite de point (\mathbf{x}_n) [11].

Dans ce qui suit, on présente deux méthodes d'optimisation stochastique en l'occurrence les algorithmes génétiques et le recuit simulé.

2.4.2.1 Méthode du recuit simulé :

La méthode du recuit simulé entre dans le cadre des méthodes d'optimisation de type « Monte Carlo » amélioré. Dans l'industrie métallurgique, le recuit désigne un procédé au cours duquel un solide, plongé dans un bain chaud, est réchauffé jusqu'à une température maximale à laquelle les particules se réarrangent de façon aléatoire en phase liquide. Cette étape est suivie d'un lent refroidissement du bain chaud et l'on montre que les particules du

Chapitre 2 : Optimisation classique

solide se réarrangent de manière à atteindre un niveau d'énergie plus bas, pourvu que la température maximale soit « assez grande » et le refroidissement suffisamment lent[14].

A chaque palier de température T , le solide atteint un équilibre thermique caractérisé par la probabilité d'avoir un niveau d'énergie. On aboutit à la distribution de Boltzman.

$$\Pr (\text{Energie} = E) = \frac{1}{Z(T)} \exp\left(\frac{-E}{k_B T}\right)$$

Où $Z(T)$ est un coefficient dépendant de T et où le k_B est la constante de Boltzman.

Soit alors à minimiser une fonction coût

$$f(x_1, x_2, \dots, x_n)$$

Définie sur \mathbb{R}^n . A partir d'une configuration $\mathbf{x} = (x_1, \dots, x_n)$ nous allons définir la configuration suivante. Pour cela on choisit « au hasard » une configuration voisine $\mathbf{j} = (j_1, \dots, j_n)$. Désignons par Δf_{ij} la différence $f_j - f_i$. tel que $f_i = f(x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$ et $f_j = f(x_1^{(j)}, x_2^{(j)}, \dots, x_n^{(j)})$ Si $\Delta f_{ij} \leq 0$ alors la probabilité pour que \mathbf{j} soit la prochaine configuration vaut 1. On dit qu'il s'agit de la probabilité d'**acceptation**. Par contre si $\Delta f_{ij} > 0$, la probabilité d'acceptation de \mathbf{j} est posée égale à :

$$\exp\left(\frac{-\Delta f_{ij}}{c}\right)$$

Où c est un paramètre à choisir et qui sera amené à décroître. Dans le processus physique, c correspond à la température du bain. Le critère de choix est appelé critère de Metropolis. L'algorithme se déroule alors comme suit :

Chapitre 2 : Optimisation classique

- Si $\Delta f_{ij} \leq 0$, on remplace i par j et l'on répète le procédé.
- Si $\Delta f_{ij} > 0$, la configuration j sera choisie si

$$\exp\left(\frac{-\Delta f_{ij}}{c}\right) > N(0, 1)$$

Où N désigne la fonction qui choisit de façon aléatoire un nombre sur $[0, 1]$.

- Si le critère n'est pas satisfait, on définit **au hasard** une autre configuration dans le voisinage de i .
- Le processus est poursuivi jusqu'à ce que l'on atteigne un état d'équilibre. Cet état a une probabilité d'acceptation des configurations proche de celle de Boltzman, donnée par :

$$\Pr(\text{Configuration} = i) = \frac{1}{Q(c)} \exp\left(\frac{-f(i)}{c}\right)$$

Où $Q(c)$ est une constante de normalisation (souvent choisie =1) dépendant du paramètre de « contrôle » c .

- On diminue alors le paramètre et l'on réitère le procédé.
- Le processus est arrêté lorsque c atteint une valeur « petite » pour laquelle aucune nouvelle configuration n'est acceptée. La dernière configuration optimale donnant le coût **minimal global**.

D'un point de vue mathématique, le recuit simulé peut être considéré comme un algorithme décrit par une **chaîne de Markov**.

D'un point de vue mathématique, le recuit simulé converge vers un minimum global, avec la probabilité 1.

Cette méthode présente néanmoins de nombreux inconvénients par rapport aux méthodes déterministes. Outre le fait qu'il est délicat de choisir la « décroissance » du paramètre c , nous n'avons pas la possibilité d'obtenir une **estimation théorique** du temps de calcul comme c'est le cas de la méthode du recuit simulé va être grandement tributaire du point initial.

Avantage : - possibilité d'échapper d'un minimum local vers un minimum global [14].

2.4.2.2 Algorithme génétique :

Instances particulières de méthodes dites d'évolution, les algorithmes génétiques [AG] sont inspirés par l'évolution des espèces avec des phases de sélection, croisement et mutation. Ils mettent en place l'évolution d'une population de cardinal N , *i. e.* N trajectoires, itératives, simultanées interagissant entre elles, et laissant espérer une convergence de certains individus de la population vers un point d'optimum de la fonction d'objectif (fonction de coût pour un minimum, fonction d'adaptivité à maximiser pour continuer la métaphore de l'évolution) [14].

➤ Principe d'un algorithme génétique :

Un algorithme génétique possède la structure suivante :

1. Initialiser le temps ;
2. Créer une population initiale ;
3. Calculer l'adaptation de chaque individu ;
4. Tant que (il n'y a pas de solution satisfaisante) et (le temps est inférieur au temps limite) faire
 - Incrémentation le temps ;
 - Sélectionner les parents ;
 - Déterminer les gènes des nouveau-nés par recombinaison des gènes parentaux ;
 - Faire subir des mutations aléatoires à la population ;
 - Evaluer l'adaptation de chaque individu ;
 - Sélectionner les survivants ;

Fait;

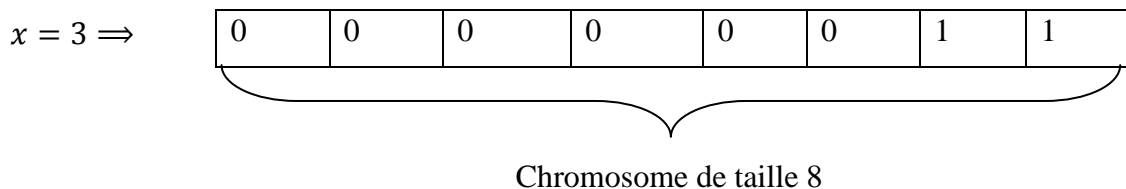
➤ Notions de la théorie des algorithmes génétiques

Chromosomes et schéma de représentation :

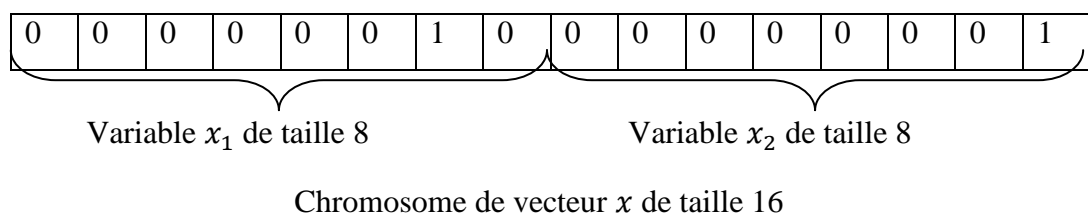
Un algorithme génétique ne travaille pas avec le variable x de la population, mais par le code attribué à cette variable (variable codée). Chaque variable de décision est codée sous forme d'une chaîne de symboles appelé chromosome, la taille de ce dernier est la même pour

Chapitre 2 : Optimisation classique

toutes les variables de la population. Les éléments du chromosome sont choisis à partir d'un ensemble de symboles appelé **alphabet**. Un choix typique de l'alphabet est l'ensemble des éléments binaire, c'est-à-dire $\{0,1\}$. Dans ce cas un chromosome est une chaîne de caractère composée des 1 et 0. Par exemple pour une fonction d'une seule variable :



Dans le cas d'une fonction à plusieurs variables par exemple : $x = (x_1, x_2) = (2,1)$



➤ Création de la population initiale

Si on n'a aucune idée de la solution du problème, la population (ensemble d'individus) est générée aléatoirement. Sinon, on crée des individus qui représentent les solutions dont on dispose. Mais le principal problème est de choisir la taille m de la population. Les biologistes ont introduits le concept de diversité requise pour représenter le fait que pour survivre, une espèce doit être suffisamment hétérogène. Par ailleurs, une population trop grande augmente le temps de calcul. Il faut donc trouver le bon compromis [14].

Chapitre 2 : Optimisation classique

➤ Evaluation de l'adaptation :

Afin de mesurer les performances de chaque individu, on introduit une fonction d'adaptation. Cette fonction correspond au profit à l'utilité de la solution par rapport au problème.

➤ Sélection des parents :

Pour que la génération suivante soit plus performante, on doit faire l'accouplement des meilleurs individus. Chaque individu aura donc une chance proportionnelle à son adaptation de devenir parent. Ainsi, on choisit un nombre $\alpha \in [0, 1]$, puis pour chaque individu $x^{(i)}$ de la population on calcule la quantité suivante :

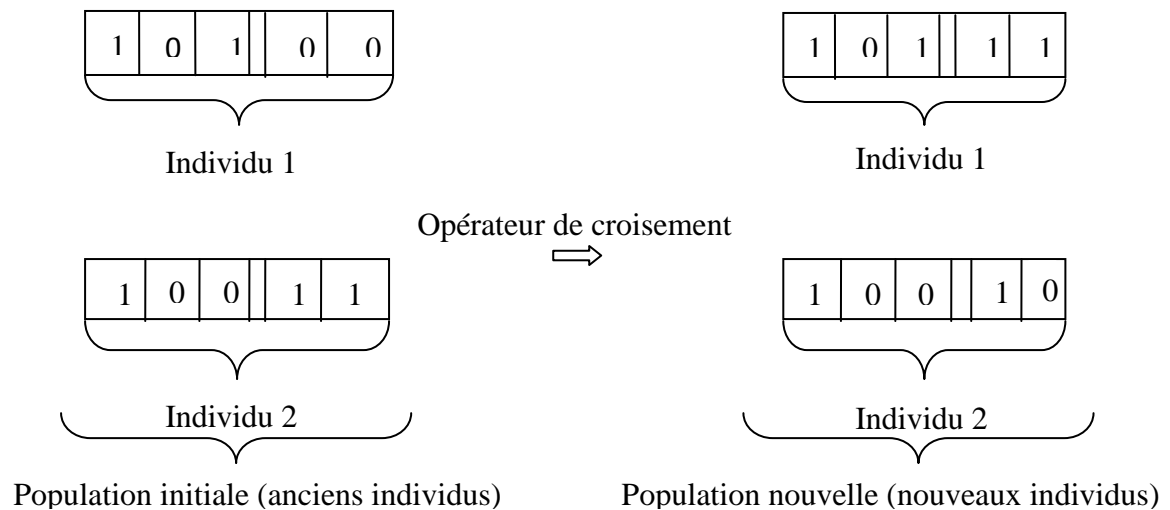
$$P_i = \frac{f(x^{(i)})}{\sum_{j=1}^m f(x^{(j)})}$$

Si $P_i > \alpha$, sélectionner $x^{(i)}$. A partir des éléments sélectionnés, on forme une nouvelle population dont la taille est la même que celle de la population initiale (m).

➤ Opérateurs génétiques :

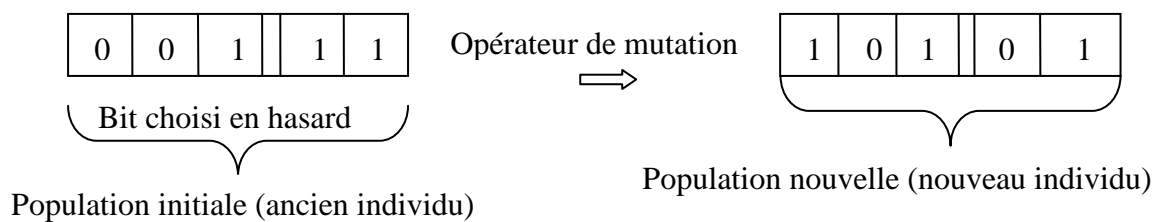
Afin de donner naissance à un individu nouveau, il ne suffit pas de copier le génome de l'un des parents, mais au contraire, de prendre aléatoirement quelques gènes de chacun des parents. Ce phénomène, présent dans la nature, est appelé crossover. Il s'agit d'un phénomène essentiel qui permet d'explorer l'ensemble des solutions possibles.

Chapitre 2 : Optimisation classique



Mutation :

Les mutations sont des modifications aléatoires du génome. Bien sûr il ne faut pas muter tous les gènes d'un individu, sinon il serait complètement déterminé aléatoirement. Il faut au contraire en modifier une petite partie, afin d'apporter quelque chose de nouveau à l'individu. Mais la mutation a un rôle secondaire par rapport à la combinaison, et on lui attribue en général une faible probabilité de l'ordre de 0.001 [14].



➤ Sélection des survivants :

Cette étape consiste à ne garder que les solutions les plus intéressantes, tout en gardant une population assez grande et assez diversifiée. On choisit en général de conserver la taille de la

Chapitre 2 : Optimisation classique

population d'une génération à l'autre. Il y aura donc autant de morts que de nouveaux nés. Parfois, on choisit de garder seulement les enfants. Cela assure la diversité requise et l'évolution de la population.

➤ Avantages et inconvénients :

Le grand avantage des algorithmes génétiques est le fait que pour parvenir au résultat, on n'a pas besoin de connaître les caractéristiques de la solution du problème, mais seulement de déterminer parmi deux solutions quelle est la meilleure. Par contre ce genre d'algorithme est très coûteux en temps de calcul, difficile à programmer (les paramètres comme le taille de la population et la fonction d'évaluation sont difficiles à établir), et il n'a qu'une très faible chance, voir aucune, de trouver la solution idéale, il ne fait qu'en approcher [14].

2.4.3 Méthodes hybrides :

Les méthodes hybrides profitent des avantages des méthodes exactes et stochastiques. Le principe de ces méthodes consiste à coopérer une méthode heuristique et une méthode exacte pour localiser l'optimum global recherché avec une précision déterminée. En général, la recherche de l'optimum global commence par l'utilisation d'une heuristique qui localise un \mathbf{x}_0 appartenant au voisinage de l'optimum, c'est-à-dire $f_\varepsilon(\mathbf{x})$, puis on passe le relais à une méthode d'optimisation exacte qui permet de localiser l'optimum globale \mathbf{x}_0 avec une bonne précision.

L'utilisation des méthodes hybrides permet de combiner les avantages des deux types de méthodes (déterministes et stochastiques), elles peuvent être vu comme la solution parfaite vis-à-vis des désavantages des méthodes locales et des méthodes globales. Au fait, les méthodes locales mènent à une solution locale. Tandis que les méthodes globales consomment énormément de temps, il parait claire qu'un compromis entre exploitation et exploration doit être trouvé d'où l'utilité des méthodes hybrides. Quand l'hybridation est basée sur une vraie maîtrise de l'idée dernière chacune des méthodes candidates, l'augmentation de la précision ainsi que la diminution du temps de calcul est assuré. Dans cet esprit, plusieurs types d'hybridation sont possibles [12].

Deux stratégies peuvent être utilisées pour expliquer le principe d'hybridation :

Chapitre 2 : Optimisation classique

Stratégie 1- On pourra par exemple s'approcher de l'optimum global par une méthode aléatoire puis affiner le résultat en appliquant successivement une méthode locale. Cette hybridation s'appelle l'hybridation bête parce qu'elle applique une méthode après l'autre. Il est clair que le résultat sera meilleur mais au détriment du temps de calcul, on consommera davantage de temps que si on avait appliqué les deux méthodes séparément. Un vrai sens d'hybridation, dans ce cas de figure, peut être d'appliquer une méthode globale et de trouver le bon moment de faire la commutation avec une méthode locale.

Stratégie 2- On pourra aussi essayer de trouver un état de minimum local et de n'y rester que s'il est le meilleur de minimums locaux (global).

Cette stratégie a été utilisée dans trois méthodes d'optimisation globales suivantes :

La méthode des initialisations multiples ou (multistart). On utilise plusieurs fois une technique d'optimisation locale en différents points de C ; on prend le meilleur résultat ; il serait solution de PO.

La méthode de **tunnelier** (tunneling method). On cherche par une technique locale un minimum local x_l ; on utilise une technique (du tunnelier) pour éliminer tous les points qui ont une valeur supérieure à celle de x_l (en effet ils ne pourraient être optimum global). Puis on ré-optimise pour trouver un x_{l+1} et ainsi de suite. La suite des x_k ainsi construite converge vers le minimum global. Voir la figure ci-dessous :

La méthode du **montagnes russes**.

D'abord on cherche un minimum local, puis on cherche cette fois-ci un maximum local ; cela permet d'échapper de l'état de minimum local. Puis on refait une minimisation suivie d'une autre maximisation et ainsi de suite. Le plus petit minimum trouvé à la fin, est candidat pour être le minimum global.

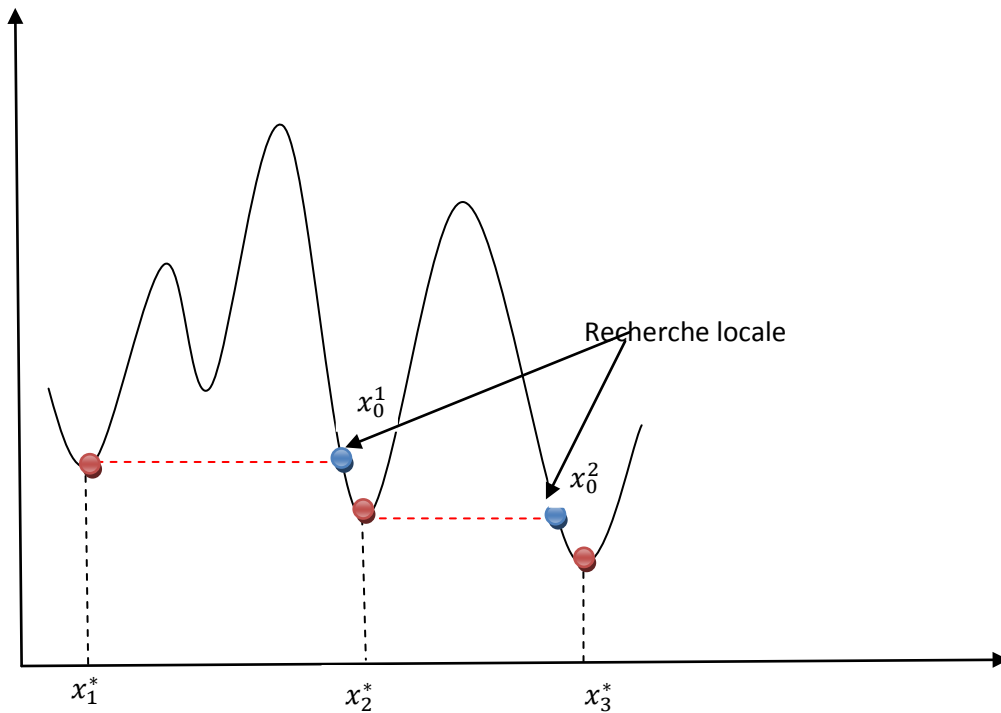


Figure 2.2 : La méthode de Tunnelier.

Choix d'une méthode d'optimisation :

On peut se demander à quoi sert une classification des types de problèmes et de méthodes. Quand un utilisateur est confronté à un problème d'optimisation globale, la première chose à faire est de bien cerner le problème, à savoir :

- Hypothèses sur f (différentiabilité, convexité..)
- Hypothèse sur le domaine de recherche (convexe,concave).
- Cout d'évaluation de la fonction (temps CPU, nombre de sous programmes nécessaire),
- Facilité de l'évolution (accès, formule explicite de f)
- Précision dont on dispose sur les calculs,
- Type de matériel utilisé,
- Temps dont on dispose pour résoudre le problème.

Chapitre 2 : Optimisation classique

Donc connaître une classification des problèmes et des méthodes, peut faciliter la tâche de l'utilisateur et le guider dans son choix ; ce qui lui permet de fixer ses objectifs en conséquence. Car il est préférable d'avoir une solution approchée (avec une précision insuffisante) dans un délai raisonnable qu'une solution exacte (avec la précision souhaitée) hors délais raisonnable. Pour choisir la méthode la plus adaptée à un problème bien précis, les caractéristiques principales prises en compte sont [12] :

La capacité à éviter les minimas locaux. C'est selon la complexité du problème qu'on peut choisir une ou l'autre des méthodes présentées en haut. Par exemple si la fonction objective est convexe il est sans doute recommander de travailler avec une méthode locale qui permet d'atteindre l'optimum rapidement par rapport aux méthodes d'optimisation globale.

Ceci dit, si la fonction objectif est multimodale ce n'est pas la peine d'appliquer une méthode local et le recourt à des méthodes globales devient nécessaire. Quoique là encore plusieurs méthodes existent.

La capacité à traiter des problèmes mono-ou multi-objectif. L'optimisation à objectifs multiples est un domaine de recherche très actif car les enjeux économiques et industriels sont énormes. Les méthodes d'optimisation multi objectif, permettant de fournir au concepteur un ensemble de solutions correspondant à autant de compromis entre les différents objectifs antagonistes du problème. D'où leur importance.

La rapidité de convergence. C'est à dire quel est le nombre de variables à évaluer pour converger vers un optimum global ? La réponse à cette question réside dans le compromis qu'il faut trouver entre exploration et exploitation.

✓ **Exploration et exploitation des algorithmes d'optimisation**

Pour un algorithme d'optimisation, **l'exploration** est sa capacité à explorer le domaine des variables pour chercher la meilleure vallée, c'est-à-dire celle qui contient l'optimum global. A l'inverse, **l'exploitation** est sa capacité à converger rapidement vers le minimum d'une vallée donnée à partir d'un point de départ.

Le succès et l'efficacité d'une technique de résolution dépendent la plupart du temps d'un compromis entre l'exploration l'exploitation.

Chapitre 2 : Optimisation classique

Tout algorithme d'optimisation doit utiliser ces deux stratégies pour trouver l'optimum global : l'exploration pour la recherche de régions inexplorées de l'espace de recherche et l'exploitation pour exploiter la connaissance acquise aux points déjà visités et ainsi trouver des points meilleurs.

2.5 Conclusion :

Ce chapitre a été consacré à des généralités sur l'optimisation des fonctions mathématiques, et les différentes méthodes utilisées pour déterminer l'optimum global avec précision. Malheureusement aucune méthode d'optimisation n'est capable de traiter efficacement tout les cas. Dans le chapitre suivant, on s'intéressera à la formulation des problèmes de commande optimale sous forme d'optimisation du type min-max.



Chapitre 3

3.1 Introduction

Généralement, la formulation des problèmes de commande pour les systèmes incertains conduit à un problème d'optimisation classique du type min-max. Dans le présent chapitre on s'intéresse à la formulation d'un problème de commande optimale, d'un système incertain, sous forme d'un problème optimisation min-max.

3.2 Description des incertitudes

Dans le cadre de la commande d'un système incertain, il est nécessaire de préciser l'ensemble des modèles possibles représentant le système à commander. Un tel ensemble est défini par un modèle nominal et un domaine d'incertitude [13].

3.2.1 Structuration d'incertitudes

Généralement, on distingue deux types d'incertitudes :

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

- **Incertitudes structurées**

Il y a deux types d'incertitudes structurées :

- Les incertitudes structurées à structuration forte : les incertitudes sont données sur les paramètres physiques ($\mathbf{m} = \mathbf{m}_0 \pm \Delta\mathbf{m}$).
- Les incertitudes structurées à structuration faible : les incertitudes sont données sur les paramètres du modèle [13].

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

Exemples :

Fonction de transfert :

$$G(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + 1} \quad (3.1)$$

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

Modèle d'état :

Le modèle d'état d'un système incertain est donné comme suit :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.2)$$

$$z(t) = Cx(t) \quad (3.3)$$

avec

$$A = A_0 + \Delta A \quad (3.4)$$

$$B = B_0 + \Delta B \quad (3.5)$$

$$C = C_0 + \Delta C \quad (3.6)$$

- **Incertitudes non structurées**

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

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

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

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

3.2.2 Représentations des incertitudes

Pour représenter ces incertitudes, on utilise les formes additives et les formes multiplicatives :

- **Forme additive :**

$$G(s) = G_0(s) + \Delta_a(s) \quad (3.7)$$

avec

$$\sigma[\Delta_a(jw)] \leq \delta_a(w), \forall w > 0 \quad (3.8)$$

Cette incertitude est représentée par la Figure 3.1.

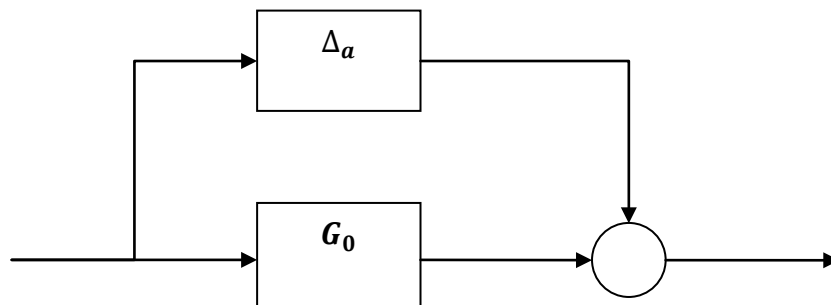


Figure 3.1 : Incertitudes additives.

- **Forme multiplicative en entrée :**

Cette incertitude est donnée comme suit :

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$G(s) = G_0(s)[I + \Delta_e(s)] \quad (3.9)$$

avec

$$\sigma[\Delta_e(jw)] \leq \delta_e(w), \forall w > 0 \quad (3.10)$$

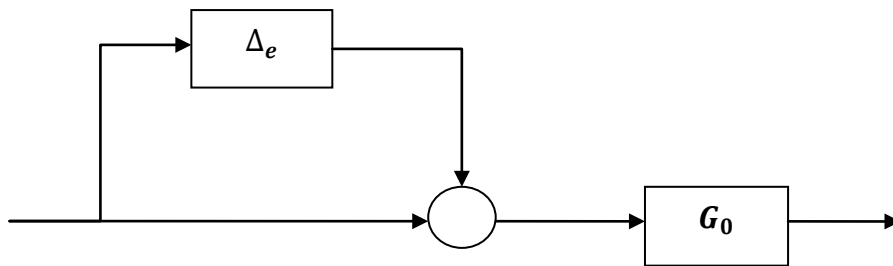


Figure 3.2 : Incertitudes multiplicatives en entrée.

- **Forme multiplicative en sortie :**

Mathématiquement, cette incertitude est donnée comme suit :

$$G(s) = [I + \Delta_s(s)]G_0(s) \quad (3.11)$$

avec

$$\sigma[\Delta_s(jw)] \leq \delta_s(w), \forall w > 0 \quad (3.12)$$

Cette incertitude est schématisée sur la Figure 3.3.

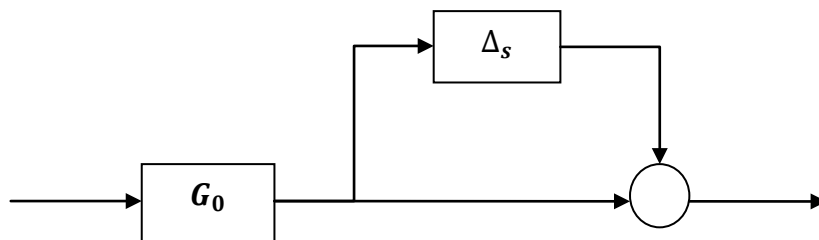


Figure 3.3 : Incertitudes multiplicatives en sortie.

3.3 Contrôleur minimax en boucle fermée

Nous examinerons d'abord la conception du contrôleur de rétroaction pour les systèmes incertains où la connaissance de la gamme et la répartition des incertitudes sont connues. Un problème d'optimisation min-max est formulé de sorte à déterminer des gains de la rétroaction pour garantir les performances même pour les pires cas d'incertitude [15]. La fonction de coût à minimiser est l'intégrale d'une fonction quadratique des états et du contrôle donnée comme suit

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (3.13)$$

Les conditions initiales du système sont supposées du type Gaussiennes à moyenne nulle et de covariance Σ , c'est-à-dire

$$\Sigma = E[x(0)x(0)^T] \quad (3.14)$$

Considérez l'ensemble des lois de commande sous forme de retour d'état

$$u = -Kx \quad (3.15)$$

Ainsi, la fonction coût (3.13) peut s'écrire comme suit :

$$J = \int_0^{\infty} x^T (Q + K^T R K) x dt \quad (3.16)$$

et la dynamique en boucle fermée est:

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$\dot{x} = Ax + Bu = (A - BK)x, \quad (3.17)$$

d'où la solution est :

$$x(t) = e^{(A-BK)t}x(0) \quad (3.18)$$

La fonction de coût qui en résulte est :

$$J = \int_0^{\infty} x(0)^T e^{((A-BK)t)^T} (Q + K^T RK) e^{(A-BK)t} x(0) dt \quad (3.19)$$

On remarque qu'elle dépend des conditions initiales incertaines et de la matrice du retour d'état K.

Posons

$$P = \int_0^{\infty} e^{((A-BK)t)^T} (Q + K^T RK) e^{(A-BK)t} dt \quad (3.20)$$

Ce qui permet de représenter la fonction coût comme suit:

$$J = x(0)^T P x(0) \quad (3.21)$$

En utilisant la propriété suivante :

$$\frac{d}{dt} \int_{t_0}^t \phi(t, \tau) d\tau = \phi(t, t) + \int_{t_0}^t \frac{\partial \phi}{\partial t}(t, \tau) d\tau \quad (3.22)$$

On calcule la dérivée de P par rapport au temps comme suit :

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$\begin{aligned}
 \frac{dP}{dt} &= e^{((A-BK)0)^T} (Q + K^T R K) e^{(A-BK)0} \\
 &+ \int_0^\infty \frac{\partial}{\partial t} \left(e^{((A-BK)t)^T} \right) (Q + K^T R K) e^{(A-BK)t} dt \\
 &+ \int_0^\infty e^{((A-BK)t)^T} (Q + K^T R K) \frac{\partial}{\partial t} \left(e^{(A-BK)t} \right) dt
 \end{aligned} \tag{3.23}$$

Qui peut être simplifié comme suit :

$$\begin{aligned}
 \frac{dP}{dt} &= (Q + K^T R K) + (A - BK)^T \int_0^\infty e^{((A-BK)t)^T} (Q + K^T R K) e^{(A-BK)t} dt \\
 &+ \int_0^\infty e^{((A-BK)t)^T} (Q + K^T R K) e^{(A-BK)t} dt (A - BK)
 \end{aligned} \tag{3.24}$$

d'où

$$\frac{dP}{dt} = (Q + K^T R K) + (A - BK)^T P + P(A - BK) \tag{3.25}$$

Puisque, pour les systèmes stables P tend à une matrice constante lorsque le temps tend vers l'infini, alors

$$P(A - BK) + (A - BK)^T P = -(Q + K^T R K) \tag{3.26}$$

Par conséquent, la fonction coût peut maintenant être réécrite comme suit :

$$\begin{aligned}
 J &= \int_0^\infty x^T (Q + K^T R K) x dt \\
 &= - \int_0^\infty x^T (P(A - BK) + (A - BK)^T P) x dt \\
 &= - \int_0^\infty x^T P \dot{x} + \dot{x}^T P x dt = - \int_0^\infty \frac{d}{dt} x^T P x dt
 \end{aligned} \tag{3.27}$$

Qui peut être simplifié comme suit :

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$J = -(x^T P x|_{t_f} - x^T P x|_{t_0}) = x(0)^T P x(0) \quad (3.28)$$

où $x(0)$ est le vecteur des conditions initiales aléatoires. Le coût peut être réécrit comme :

$$J = E[x(0)^T P x(0)] = tr(E[x(0)x(0)^T P]) = tr(\Sigma P) \quad (3.29)$$

Cette fonction doit être maximisée par rapport au vecteur de paramètres θ du système incertain et minimiser par rapport à K (gains du retour d'état). Donc, on a problème d'optimisation du type min-max donné comme suit :

$$\min_K \max_{\theta} J = tr(\Sigma P)$$

Où $tr(\)$ représente la trace de la matrice. Ainsi, par la résolution de l'équation (3.26), on peut calculer le coût quadratique intégral pour toute matrice de gain de rétroaction [15]. Cette technique simple peut être exploitée pour optimiser la matrice des gains de rétroaction afin de minimiser la fonction coût (3.13).

En général, minimiser la fonction coût (3.29) est très difficile car elle présente des propriétés mathématiques particulières (variations rapides, discontinuité, ...). Alors, au lieu de s'intéresser à l'optimisation de la fonction (3.29), il est préférable de considérer la fonction suivante :

$$\tilde{J} = -\frac{1}{tr(\Sigma P)} \quad (3.30)$$

Considérons le système masse-ressort-amortisseur donné par la figure 3.4.

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

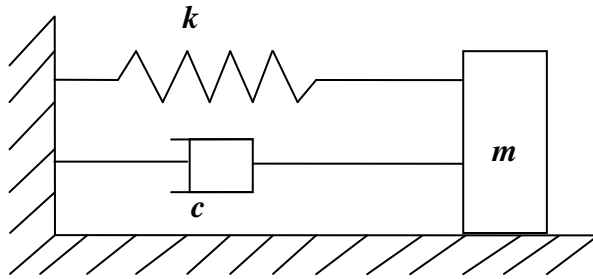


Figure 3.4 : Système mass ressort amortisseur.

L'équation différentielle régissant le déplacement de la masse m est donnée comme suit [15] :

$$\ddot{y} + c\dot{y} + ky = 0 \quad (3.31)$$

Où c et k représentent respectivement le coefficient de frottement et la raideur du ressort.

La fonction de coût est :

$$J = tr(\Sigma P) \quad (3.32)$$

où P est solution de l'équation (3.26), et la matrice de covariance Σ est une matrice identité.

Après changement de variables le modèle du système sous forme est donné comme suit :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.33)$$

Où $x_1 = y$.

Supposons que l'objectif est de minimiser le critère suivant :

$$J = \int_0^{\infty} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} dt \quad (3.34)$$

D'après les données du problème, on identifie les matrices suivantes :

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$A = \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix} \quad (3.35)$$

$$B = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.36)$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.37)$$

$$R = 0 \quad (3.38)$$

D'où l'équation suivante

$$PA + A^T P = -Q \quad (3.39)$$

Posons $P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}$

D'après l'équation (3.39) on obtient :

$$\begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix}^T \times \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -kP_2 & P_1 - cP_2 \\ -kP_3 & P_2 - cP_3 \end{bmatrix} + \begin{bmatrix} -kP_2 & -kP_3 \\ P_1 - cP_2 & P_2 - cP_3 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -2kP_2 & P_1 - cP_2 - kP_3 \\ P_1 - cP_2 - kP_3 & 2(P_2 - cP_3) \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

D'où

$$1 - 2kP_2 = 0$$

$$P_1 - cP_2 - kP_3 = 0$$

$$2(P_2 - cP_3) + 1 = 0$$

La solution de ce système d'équations algébriques donne :

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$P_1 = \frac{(c^2 + k^2 + k)}{(2 \times c \times k)}$$

$$P_2 = \frac{1}{(2 \times k)}$$

$$P_3 = \frac{(k + 1)}{(2 \times c \times k)}$$

Posons

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Calculons alors :

$$J = \text{tr}(\Sigma P)$$

$$\Sigma P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \frac{(c^2 + k^2 + k)}{(2 \times c \times k)} & \frac{1}{(2 \times k)} \\ \frac{1}{(2 \times k)} & \frac{(k + 1)}{(2 \times c \times k)} \end{bmatrix}$$

$$\Sigma P = \begin{bmatrix} \frac{(c^2 + k^2 + k)}{(2 \times c \times k)} & \frac{1}{(2 \times k)} \\ \frac{1}{(2 \times k)} & \frac{(k + 1)}{(2 \times c \times k)} \end{bmatrix}$$

$$J = \text{tr} \left(\begin{bmatrix} \frac{(c^2 + k^2 + k)}{(2 \times c \times k)} & \frac{1}{(2 \times k)} \\ \frac{1}{(2 \times k)} & \frac{(k + 1)}{(2 \times c \times k)} \end{bmatrix} \right)$$

$$J = \frac{(c^2 + k^2 + 2k + 1)}{(2 \times c \times k)}$$

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

$$\tilde{J} = -\frac{1}{\text{tr}(\Sigma P)}$$

$$\tilde{J} = -\frac{(2 \times c \times k)}{(c^2 + k^2 + 2k + 1)}$$

D'où les résultats de simulation suivant :

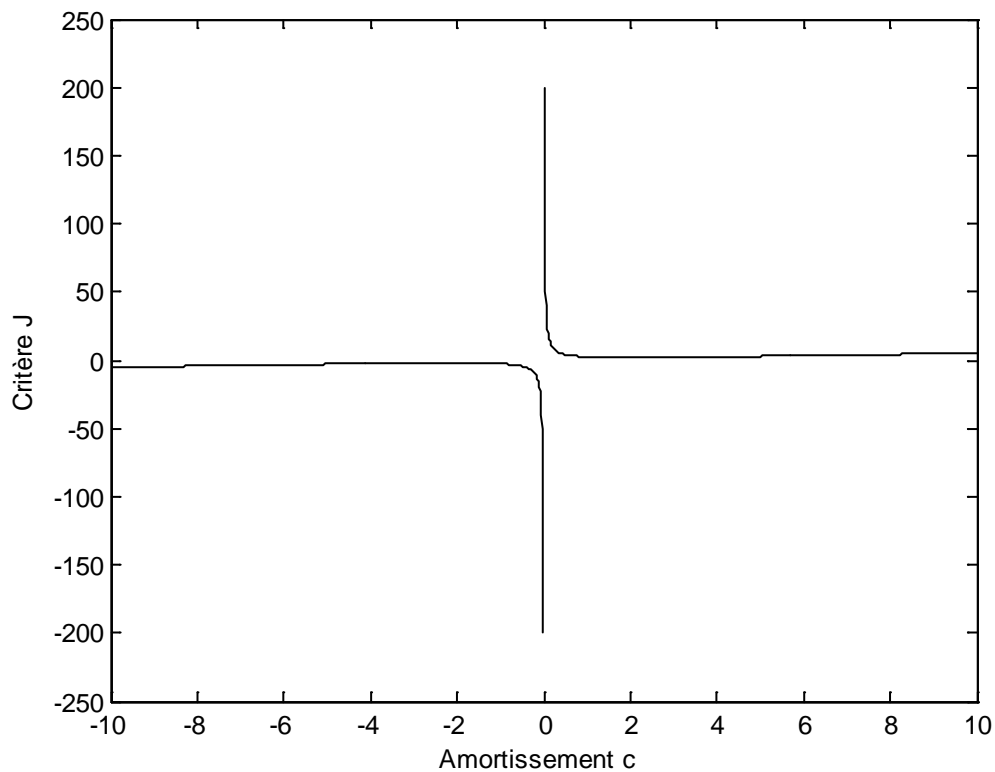


Figure 3.5 : Critère du coût quadratique.

La figure 3.5 illustre la variation de J en fonction de c. puisque le système n'est stable que pour c supérieure à 0, nous ne intéressons qu'à la partie du graphe où $c > 0$. On remarque que le graphe présente une discontinuité autour de l'origine qui correspond au passage de la zone stable à la région instable. De plus, il est très difficile de localiser l'optimum global car la variation de la fonction est très lente d'où la difficulté d'application des méthodes analytiques.

Alors pour surmonter cette difficulté, nous allons considérer la fonction coût suivante :

$$\tilde{J} = \frac{1}{\text{tr}(\Sigma P)} \quad (3.40)$$

$$\tilde{J} = -\frac{(2 \times c \times k)}{(c^2 + k^2 + 2k + 1)}$$

La figure 3.6 donne le graph de la fonction (3.40). Il est clair que la fonction admet un seul minimum qui coïncide avec celui de J et qui est facile à localiser.

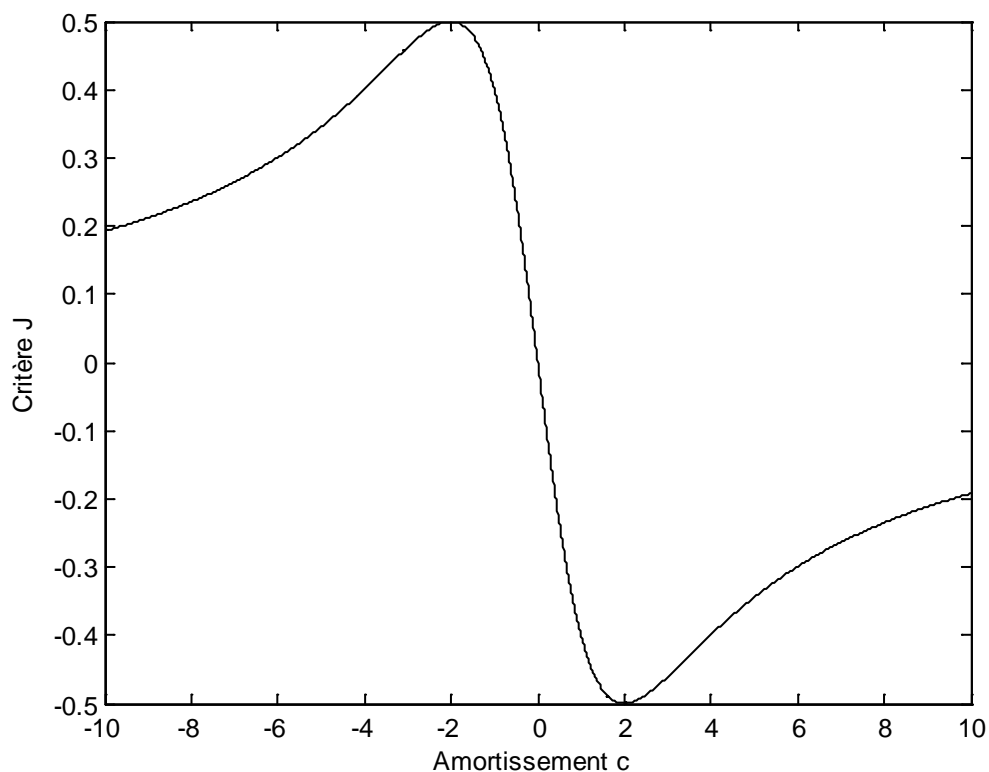


Figure 3.6 : l'inverse du négatif de J.

3.4 Conclusion :

Dans ce chapitre, nous avons formulé la synthèse d'un retour d'état pour un système incertain sous forme d'un problème d'optimisation du type min-max. On a montré que la fonction à optimiser peut avoir des propriétés mathématiques rendant la recherche de l'optimum très difficile. Pour cela, nous avons présenté une autre alternative consistant à optimiser l'inverse de cette fonction.

Chapitre 3 : Formulation d'un problème de commande optimale sous forme d'optimisation minimax

Un exemple d'application concernant le système masse-ressort-amortisseur est présenté pour montrer cette difficulté. Dans le chapitre suivant, on s'intéresse à la détermination de la matrice des gains du retour d'état en présence des incertitudes.



Chapitre 4

4.1 Introduction :

Dans ce chapitre, nous allons présenter une méthode pour la synthèse d'un retour d'état optimal en présence des incertitudes paramétriques. Comme exemple d'application on considère un système mécanique constitué de deux masses reliées avec un amortisseur et un ressort.

4.2 Optimisation min-max dans le cas de la commande LQ

L'exemple étudié au chapitre 3 a démontré l'utilité de l'utilisation de la fonction inverse du critère à optimiser. Cette approche permet d'éviter les problèmes liés au calcul du gradient numériques, puisque la fonction objectif présente des variations rapides, c'est-à-dire la fonction objectif n'est pas lisse.

Rappelons que dans le cas d'un système à paramètres incertains, la synthèse de correcteur revient à résoudre le problème d'optimisation min-max suivant

$$\min_K \max_P J(K, P)$$

Où K est la matrice des gains du retour d'état et P est le vecteur des paramètres incertains du système. Mais pour résoudre ce problème, pour des considérations numériques, on minimise la fonction objectif suivante [15] :

$$J(K, P) = -\frac{1}{\text{tr}(\Sigma P)} \quad (4.1)$$

Chapitre 4: commande LQ d'un système incertain

En présence des incertitudes, il est impossible de prendre en considération tous les cas possibles pour les paramètres du système, car nombre de cas est infini. Pour déterminer la matrice K , le problème doit être relaxé en utilisant la méthode des sommets. En effet, en supposant que le domaine des incertitudes est convexe et comme le critère à optimiser est convexe, l'optimum ne peut être qu'un des sommets du domaine.

Par exemple dans le cas d'un système à quatre paramètres incertains, le domaine est donné sous forme d'un cube. La solution du problème est l'un des points représentés sur la Figure 4.1. Donc, il suffit de résoudre le problème de commande en considérant les seize sommets et de prendre comme solution celui qui minimise la fonction objectif [15].

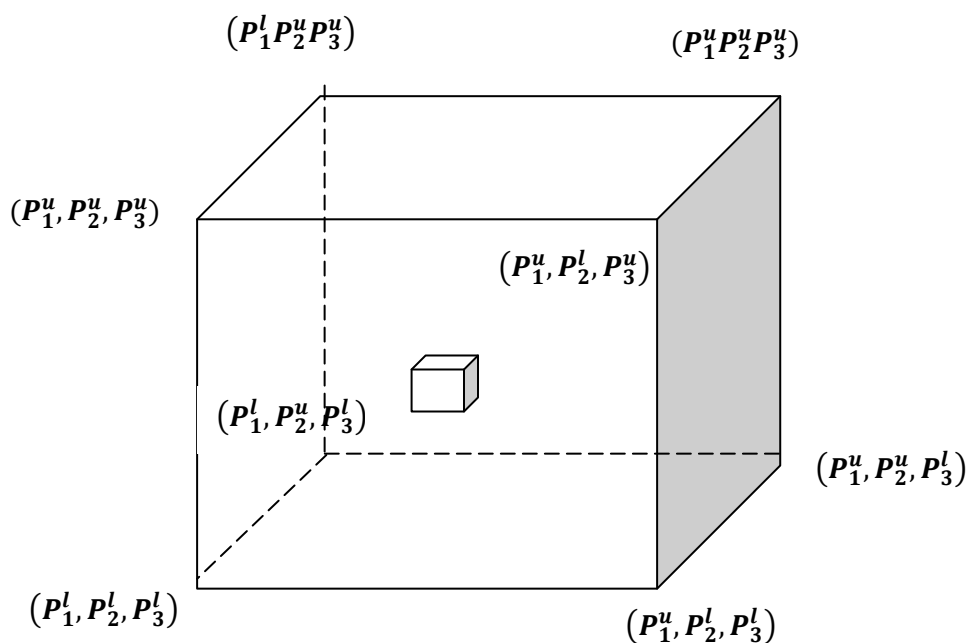


Figure 4.1 : Cube incertain.

Le petit cube au centre du domaine d'incertitudes représente le cas des paramètres du modèle nominale. Les sommets représentent les cas d'incertitudes importantes. Ainsi, en se rapprochant du centre, alors on se rapproche du modèle nominal [15].

Chapitre 4: commande LQ d'un système incertain

Ainsi, des algorithmes d'optimisation à base du gradient peuvent être utilisés pour résoudre le problème de min-max. La fonction de coût à optimiser est donnée comme suit [15] :

$$J = -\frac{1}{\text{tr}\Sigma P_i} \quad (4.2)$$

Où P_i est la solution de l'équation Lyapunov correspondant au i -ème sommet du cube donnée comme suit :

$$P_i(A - BK) + (A - BK)^T P_i = -(Q + K^T RK) \quad (4.3)$$

La dérivée de l'équation (4.3) par rapport au gain de rétroaction k_i est

$$\frac{dP_i}{dk_i}(A - BK) + (A - BK)^T \frac{dP_i}{dk_i} = P_i B \frac{dK}{dk_i} + \frac{dK^T}{dk_i} B^T P_i - K^T R \frac{dK}{dk_i} - \frac{dK^T}{dk_i} R K \quad (4.4)$$

Ainsi, le calcul du gradient du coût J par rapport au gain k_i du retour d'état est :

$$\frac{dJ}{dk_i} = \frac{1}{(\text{tr}(\Sigma P_i))^2} \text{tr} \left(\Sigma \frac{dP_i}{dk_i} \right) \quad (4.5)$$

Qui peut être exploité dans l'algorithme d'optimisation.

4.3 Commande LQ d'un système incertain

Considérons le système mécanique de la figure. 4.2. Les masses, la raideur du ressort et la constante d'amortissement sont toutes incertaines.

Le domaine d'incertitude est défini par les inéquations suivantes :

$$0.7 \begin{Bmatrix} m_1^{nom} \\ m_2^{nom} \\ k^{nom} \\ c^{nom} \end{Bmatrix} \leq \begin{Bmatrix} m_1 \\ m_2 \\ k \\ c \end{Bmatrix} \leq 1.3 \begin{Bmatrix} m_1^{nom} \\ m_2^{nom} \\ k^{nom} \\ c^{nom} \end{Bmatrix}$$

Où $(.)^{nom}$ désigne la valeur nominale du paramètre incertain.

Le nombre des paramètres du modèle est égal à 4, donc le nombre de sommets est égal à 16. Donc, on doit résoudre le problème de commande LQ pour les différents cas et de prendre comme solution, celle qui minimise la fonction de coût à optimiser.

On suppose que les conditions initiales ont la propriété suivante [15] :

$$E[x(0)x(0)^T] = \Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

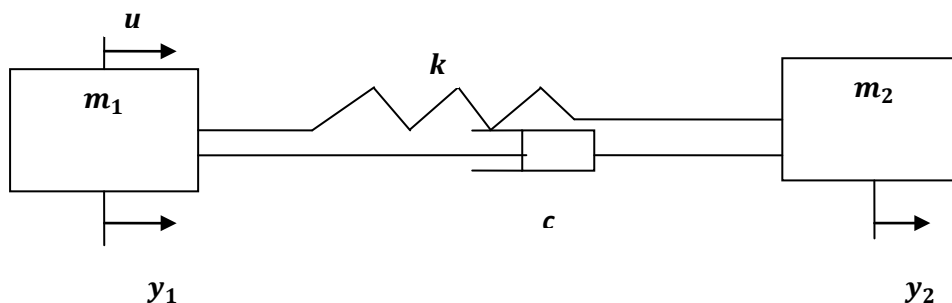


Figure 4.2 : système deux masses ressort amortisseur

4.3.1 Modélisation du système :

Pour formuler le problème de commande LQ à résoudre, on commence d'abord par la modélisation du système. Pour ce faire, on utilise le formalisme de Lagrange basée sur le calcul des énergies cinétique et potentielle totale du système.

L'énergie cinétique totale du système est :

$$E_c = \frac{1}{2} m_1 \dot{y}_1^2 + \frac{1}{2} m_2 \dot{y}_2^2 \quad (4.6)$$

L'énergie potentielle totale du système est :

$$E_p = \frac{1}{2} k (y_2 - y_1)^2 \quad (4.7)$$

Et l'énergie dissipée est :

$$D = \frac{1}{2} c (\dot{y}_2 - \dot{y}_1)^2 \quad (4.8)$$

Le seul effort externe est la force appliquée sur la masse m_1

Appliquons maintenant le formalisme de Lagrange pour déterminer le modèle mathématique du système. L'équation différentielle régissant la dynamique de chaque masse est donnée par l'équation suivante :

$$\frac{d}{dt} \left(\frac{\partial E_c}{\partial \dot{y}_1} \right) - \frac{\partial E_c}{\partial y_1} + \frac{\partial E_p}{\partial y_1} + \frac{\partial E_d}{\partial \dot{y}_1} = \sum F_i \quad (4.9)$$

Où F_i est la i -ième force appliquée sur la masse m_i .

Appliquons cette équation pour les différentes masses :

Masse m_1 :

$$\frac{d}{dt} \left(\frac{\partial E_C}{\partial \dot{y}_1} \right) - \frac{\partial E_C}{\partial y_1} + \frac{\partial E_P}{\partial y_1} + \frac{\partial E_D}{\partial \dot{y}_1} = U \quad (4.10)$$

Il y a une seule force u appliquée à la masse m_1 .

Les calculs conduisent à l'équation suivante :

$$m_1 \ddot{y}_1 + k(y_1 - y_2) + c(\dot{y}_1 - \dot{y}_2) = u \quad (4.11)$$

Masse m_2 :

$$\frac{d}{dt} \left(\frac{\partial E_C}{\partial \dot{y}_2} \right) - \frac{\partial E_C}{\partial y_2} + \frac{\partial E_P}{\partial y_2} + \frac{\partial E_D}{\partial \dot{y}_2} = 0 \quad (4.12)$$

$$m_2 \ddot{y}_2 + k(y_2 - y_1) + c(\dot{y}_2 - \dot{y}_1) = 0 \quad (4.13)$$

Ces deux équations différentielles peuvent être écrites sous la forme matricielle suivante :

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} + \begin{bmatrix} c & -c \\ -c & c \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} + \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \quad (4.14)$$

Qu'on peut mettre sous la forme d'état suivante :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.15)$$

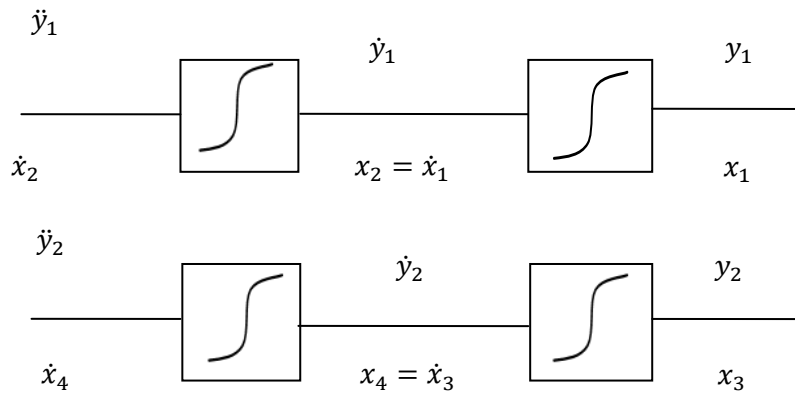


Figure 4.3 : Méthodes des intégrateurs.

Ce qui donne les matrices d'état et de commande suivante :

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k}{m_1} & \frac{k}{m_1} & -\frac{c}{m_1} & \frac{c}{m_1} \\ \frac{k}{m_2} & -\frac{k}{m_2} & \frac{c}{m_2} & -\frac{c}{m_2} \end{bmatrix} \quad (4.16)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix} \quad (4.17)$$

4.3.2 Commande LQ

Dans cet exemple on cherche à trouver le gain optimal qui permet d'optimiser le critère quadratique de système à paramètres incertains.

Le critère à quadratique est :

$$J(u(t)) = \int_0^{\infty} x^T(t) Q x(t) + u^2(t) dt \quad (4.18)$$

Avec :

$$R = 1 \quad (4.19)$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

4.3.3 Solution du problème

Pour l'objectif de comparaison, nous allons commencer par la synthèse d'un retour d'état pour le cas d'un modèle nominal. Les paramètres du modèle nominal sont pris comme la moyenne des valeurs minimale et maximale.

$$\begin{Bmatrix} m_1^{nom} \\ m_2^{nom} \\ k^{nom} \\ c^{nom} \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{Bmatrix} \quad (4.21)$$

L'utilisation de la solution de l'équation de Riccati (1.37) nous a conduit à trouver le gain de retour d'état nominal qui minimise le critère quadratique pour les paramètres nominal donnés dans l'équation (4.21)

où

$$K^{nom} = R^{-1} B^T K \quad (4.22)$$

$$K^{nom} = [0.8026 \quad 0.6116 \quad 1.6323 \quad 1.4699] \quad (4.23)$$

Et le coût correspondant est:

$$J = -0.2068$$

Considérons maintenant le cas des paramètres incertains. On a quatre paramètres incertains alors on a seize cas à étudier.

Chapitre 4:commande LQ d'un système incertain

	m1	m2	K	C
Cas 1	0.7	0.7	0.7	0.7
Cas 2	0.7	0.7	1.3	0.7
Cas 3	0.7	1.3	0.7	0.7
Cas 4	0.7	1.3	1.3	0.7
Cas 5	1.3	0.7	0.7	0.7
Cas 6	1.3	0.7	1.3	0.7
Cas 7	1.3	1.3	0.7	0.7
Cas 8	1.3	1.3	1.3	0.7
Cas 9	0.7	0.7	0.7	1.3
Cas 10	0.7	0.7	1.3	1.3
Cas 11	0.7	1.3	0.7	1.3
Cas 12	0.7	1.3	1.3	1.3
Cas 13	1.3	0.7	0.7	1.3
Cas 14	1.3	0.7	1.3	1.3
Cas 15	1.3	1.3	0.7	1.3
Cas 16	1.3	1.3	1.3	1.3

Tableau 4.1 : Les différentes possibilités pour les incertitudes.

Ensuite pour chaque cas, on calcule le gain et la valeur de la fonction objectif comme a été expliqué pour le cas du modèle nominal. Les différents résultats obtenus sont résumés dans le tableau suivant :

Chapitre 4:commande LQ d'un système incertain

	Kr	J
Cas 1	[0.8528 0.5614 1.3727 1.1516]	-0.186483
Cas 2	[0.9034 0.5108 1.3798 1.1220]	-0.186049
Cas 3	[0.7398 0.6744 1.2263 1.5717]	-0.133606
Cas 4	[0.7306 0.6836 1.2204 1.5587]	-0.138171
Cas 5	[1.0075 0.4067 1.8991 1.0175]	-0.137115
Cas 6	[1.1015 0.3127 1.9160 0.9576]	-0.135619
Cas 7	[0.8727 0.5415 1.7154 1.4821]	-0.107885
Cas 8	[0.9052 0.5090 1.7179 1.4307]	-0.112517
Cas 9	[0.7666 0.6476 1.3076 1.1679]	-0.195454
Cas 10	[0.7944 0.6198 1.3102 1.1581]	-0.206753
Cas 11	[0.6690 0.7452 1.1240 1.6067]	-0.146176
Cas 12	[0.6414 0.7728 1.1184 1.6149]	-0.154534
Cas 13	[0.8893 0.5249 1.8455 1.0132]	-0.145039
Cas 14	[0.9707 0.4435 1.8559 0.9828]	-0.150491
Cas 15	[0.7795 0.6347 1.6307 1.4893]	-0.117984
Cas 16	[0.8026 0.6116 1.6323 1.4699]	-0.125007

Tableau 4.2 : Gain du retour d'état et valeur du critère pour chaque incertitude.

D'après les résultats obtenus on constate que le minimum de la fonction objectif obtenu correspondant au Cas 10, alors le retour d'état pour le système incertain est :

$$K_{r_opt} = [0.7944 \quad 0.6198 \quad 1.3102 \quad 1.1581]$$

4.3.4 Résultats de simulation

Pour évaluer les performances du gain de retour d'état, on réalise des simulations en considérant quelques cas d'incertitudes et on compare les résultats au cas nominal.

Cas optimal les résultats sont donnés par la figure 4.4-4.7

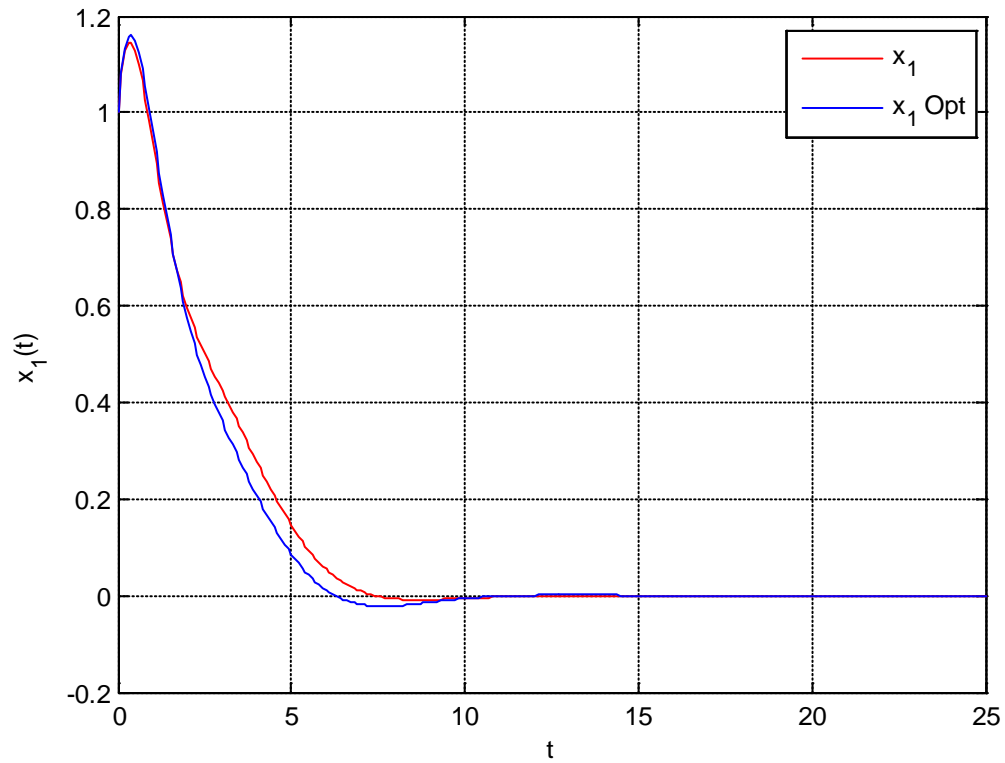


Figure 4.4 : Position de la masse m_1 .

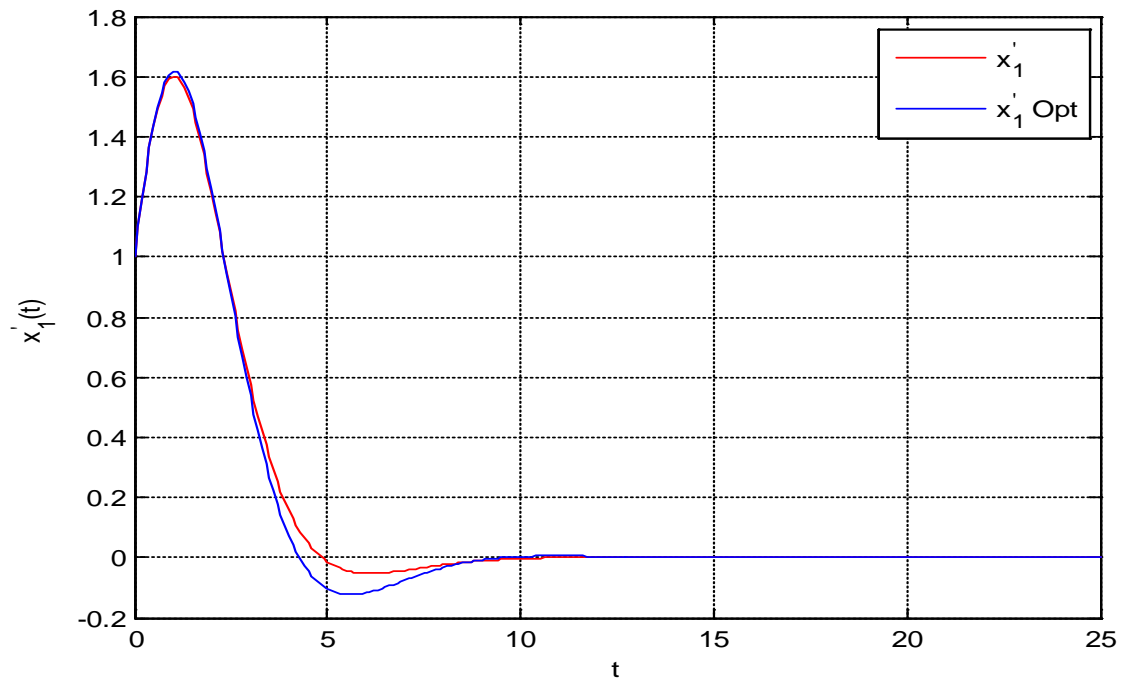


Figure 4.5 : Vitesse de la masse m1.

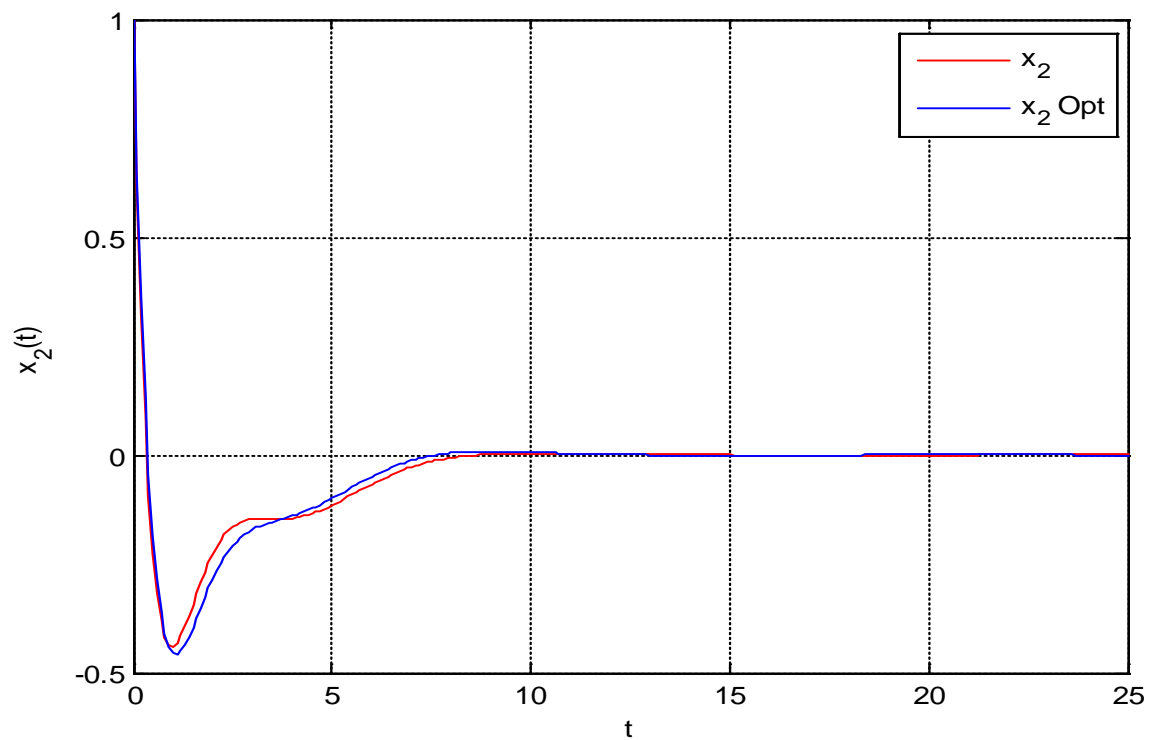


Figure 4.6 : Position de la masse m2.

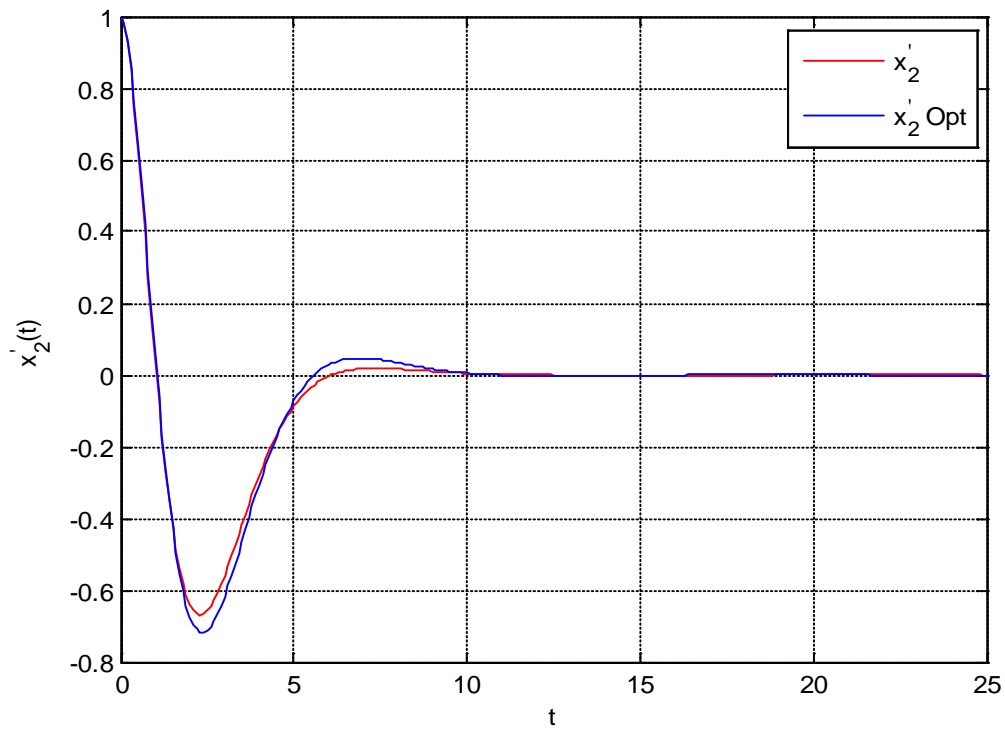


Figure 4.7 : Vitesse de la masse m2.

.Cas 2 : les résultats sont donnés par les figures 4.8-4.11

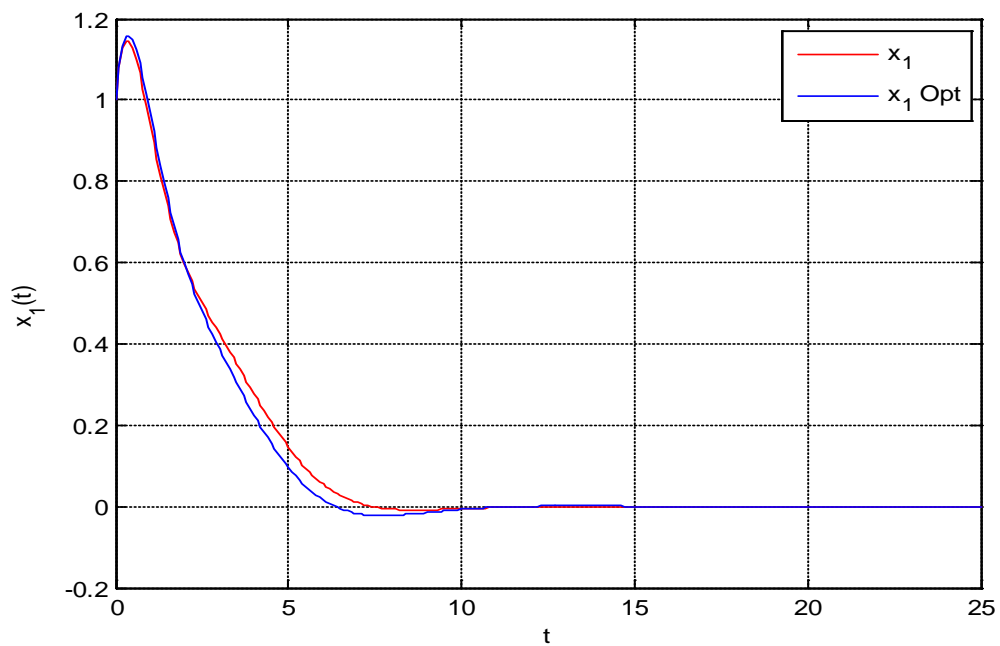


Figure 4.8 : Position de la masse m1.

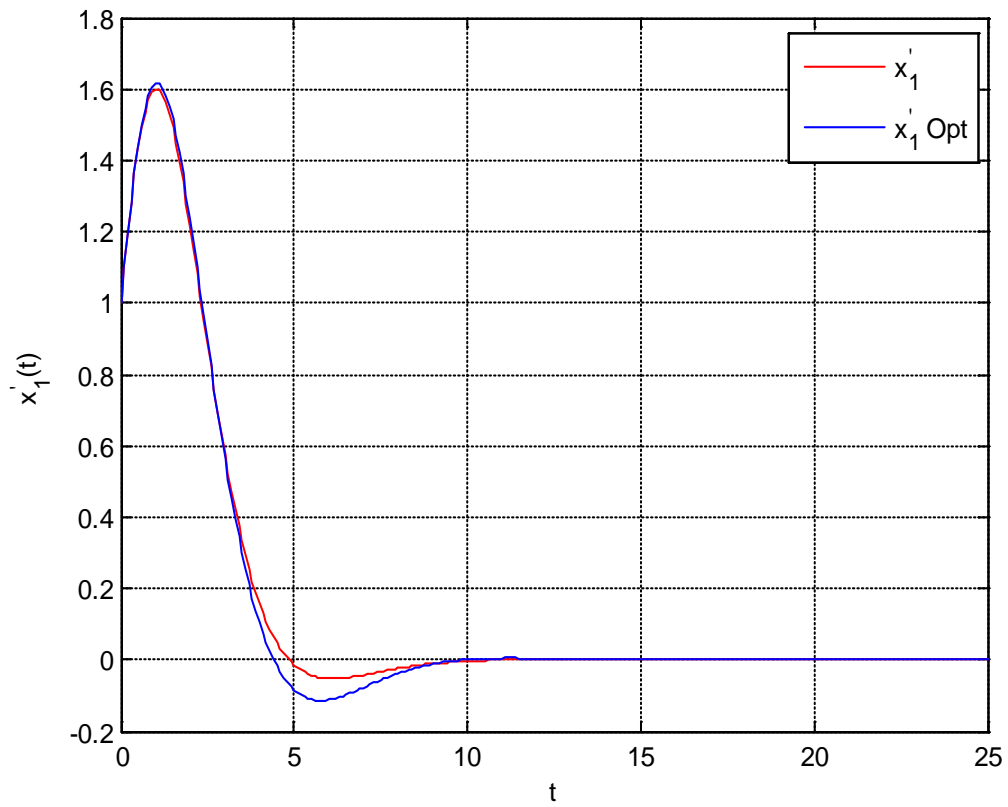


Figure 4.9 : Vitesse de la masse m1.

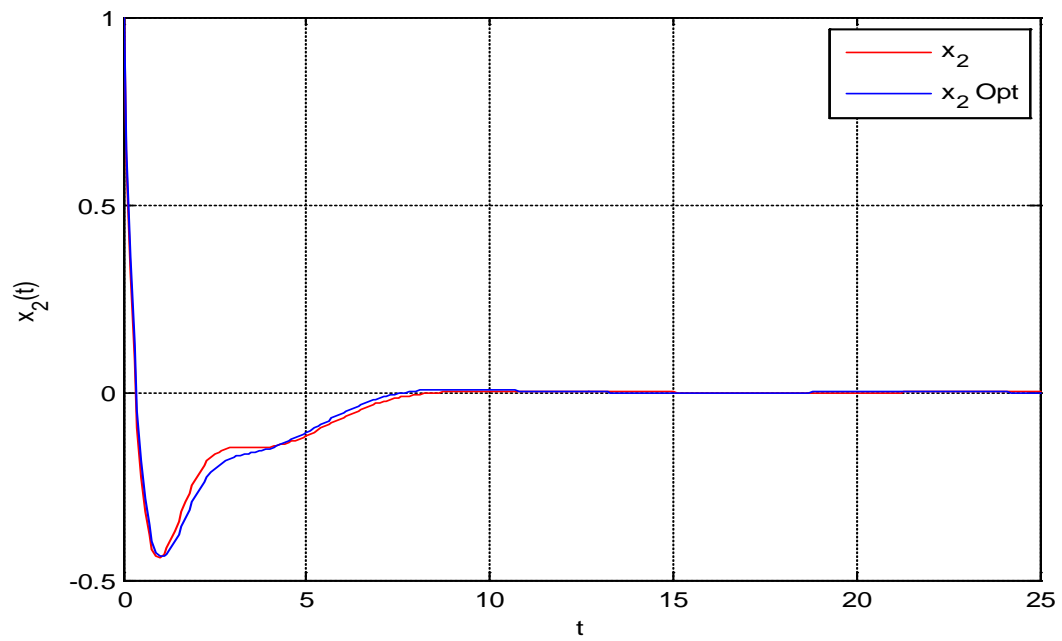


Figure 4.10 : Position de la masse m2.

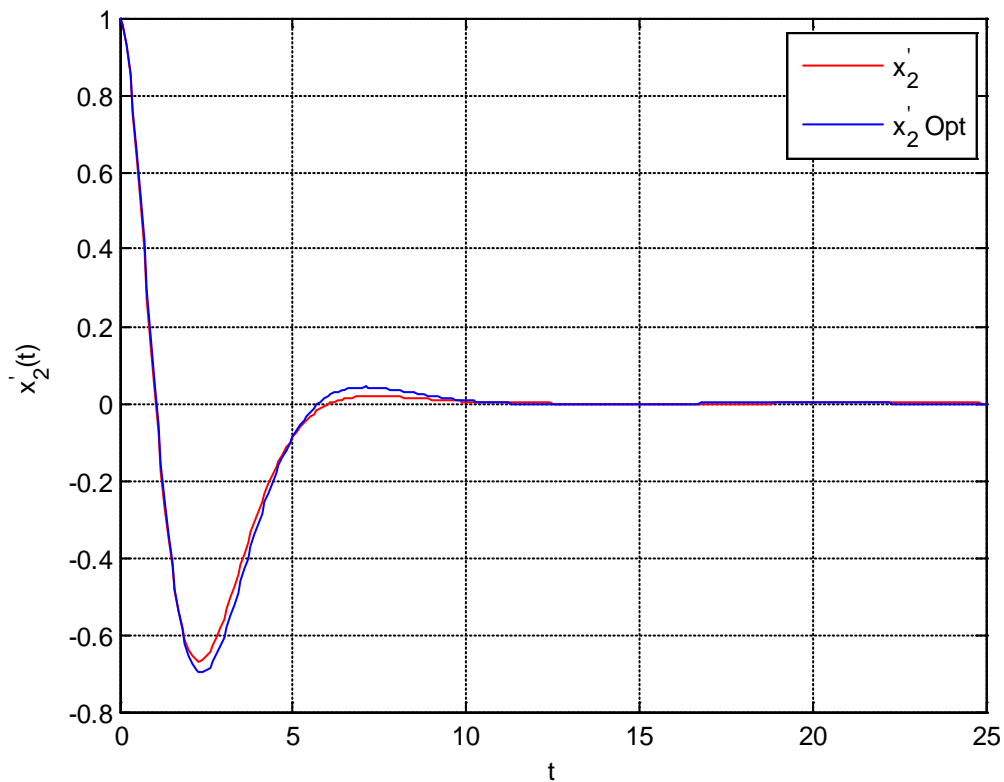
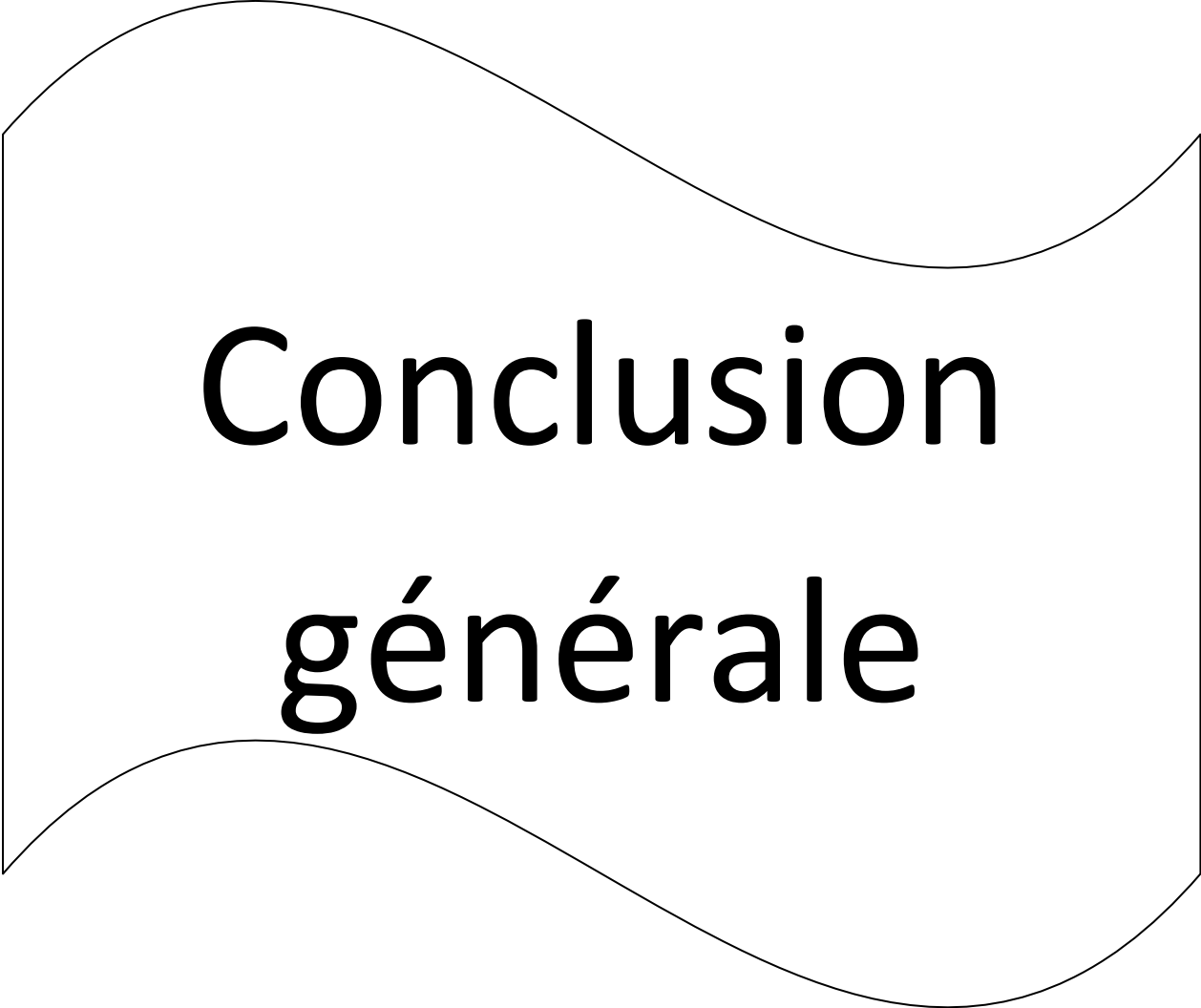


Figure 4.11 : Vitesse de la masse m2.

D'après les résultats obtenus, on constate que malgré les incertitudes, le retour d'état synthétisé assure de bonnes performances puisque la régulation est assurée.

4.4 Conclusion :

Dans ce chapitre, nous avons formulé le problème de commande LQ pour un système mécanique incertain sous forme d'un problème min-max. Nous avons commencé par la présentation d'une méthode de résolution basée sur les sommets du domaine d'incertitude. Cette méthode a été appliquée pour la commande d'un système mécanique caractérisé par 4 paramètres incertains. Après la modélisation du système, nous avons formulé le problème de commande LQ puis synthétisé le retour d'état optimal. Des résultats de simulations ont été présentés pour quelques cas d'incertitudes et comparés au cas nominal.



Conclusion générale

Conclusion générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la commande optimale des systèmes incertains. L'objectif est de concevoir une commande du type LQ pour un système incertain.

Ainsi, après avoir présenté des généralités sur les systèmes incertains, nous avons rappelé des notions de base relatives à l'optimisation classique. Après, nous avons montré que les problèmes de commande des systèmes incertains se ramènent généralement à des problèmes d'optimisation classique du type min-max. Puis, nous avons considéré le cas de la commande linéaire quadratique d'un système incertain. Une méthode de synthèse a été présentée et appliquée pour un système mécanique.

L'étude présentée confirme que la synthèse d'une commande optimale pour un système incertain est généralement très difficile. En effet, la prise en compte de la totalité du domaine d'incertitudes est quasiment impossible. A cet effet, le problème de synthèse peut être formulé sous forme d'un problème min-max qui peut être relaxé en considérant certains cas d'incertitudes seulement. Par exemple, la méthode étudiée dans le cas de la commande LQ limite l'étude aux sommets du domaine d'incertitudes seulement. Cette méthode a permis de concevoir un retour d'état assurant la régulation du système quelques soient les incertitudes.



Bibliographie

Bibliographie

- [1]F .FELLAH. « *Résolution de l'équation de Riccati avec la méthode de perturbation homotopique* », UNIVERSITE Mouloud MAMMERI de TIZI OUZOU département automatique 2016.
- [2]L.HADJOU et M.BELHOCINE. « *Contrôle actif robuste d'une structure flexible* », Université MOULOUD MAMMERI Tizi Ouzou-master en automatique, 2010.
- [3]Edouard Laroche. « *Cours Automatique commande optimale* », ECOLE NATIONALE SUPERIEURE DE PHYSIQUE DE STRASBOURG, 2009/2010.
- [4]A.MAIDI. « *Cours Master 2 commande optimale et optimisation* », UNIVERSTE Mouloud MAMMERI.TIZI OUZOU
- [5]L.CHERFI. « *Systèmes différentiels et algébriques du type Riccati issus de la théorie des jeux* », ECOLE NORMALE SUPERIEURE DE CACHAN, 2005.
- [6]P.NASLIN. « *Théorie de la commande et conduite optimale* », DUNOD, Paris 1969.
- [7]Eric Walter. « *Méthodes numériques et optimisation* », Fluctant Nec Merguntur, Paris, 2015.
- [8]M. Bergounioux. « *Cours optimisation sans contraintes* », Département de Mathématiques Orléans année 2003-2004.
- [9]https://fr.wikipedia.org/wiki/Algorithme_du_gradient 10/03/2017.
- [10]Daya OUIDJA. « *Principe du maximum et méthode de tir* », Mémoire de Magister en mathématique,2011.-
- [11]Laurent Guillopé. « *Cours mathématique thème Optimisation déterministe et stochastique* », Université de Nantes, 2017.
- [12]G. IDIRI. « *Commande prédictive des systèmes non linéaires dynamique* », UNIVERSITE Mouloud MAMMERI TIZI OUZOU, 2011.
- [13]S.HADEF. « *Calcul et robustification d'un contrôleur PID multivariable par approche LMI* », Université MENTOURI DE CONSTANTINE, 2007.
- [14]Y. Cherruault. « *Modèles et méthodes mathématiques pour les sciences du vivant* ». Presses Universitaires de France, 1998.
- [15]T. Singh. « *Optimal Reference Shaping for Dynamical Systems: Theory and Applications* ». CRC Press, Boca Raton, 2010.

Résumé

L'objectif du travail consiste à concevoir une loi de commande linéaire quadratique pour un système incertain. L'idée consiste à formuler le problème de synthèse sous forme d'un problème d'optimisation classique du type min-max. La solution de ce dernier permet de déterminer les gains du retour d'état permettant d'assurer la robustesse en stabilité et en performance.