

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – Biskra
Faculté des Sciences et de la technologie
Département : Génie Electrique
Ref :.....



جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم: هندسة كهربائية
المرجع:.....

Mémoire présenté en vue de l'obtention
du diplôme de
Magister en : Automatique

Option : Commande et Identification des Systèmes Dynamiques

**Commande Multi-Objectifs En Utilisant Les Inégalités Matricielles
Linéaires (LMIs) Et Les Algorithmes Génétiques**

Présenté par :
Houam Yahia

Soutenu publiquement le: 02/05/2013

Devant le jury composé de :

M.S.Mimoune	Professeur	Président	Université de Biskra
M. Boumehraz	Maitre de Conférences	Rapporteur	Université de Biskra
A. Allag	Professeur	Examineur	Université de Biskra
A.Betka	Professeur	Examineur	Université de Biskra

Introduction générale

Un procédé réel est sous l'influence de plusieurs sources de signaux comme le bruit, le signal de commande, le signal de consigne, etc. L'objectif d'une loi de commande est de produire un signal de commande, en utilisant des signaux de mesure pour obtenir une certaine performance désirée comme par exemple le rejet de perturbations ou la poursuite désirée. Ces performances sont souvent représentées par des signaux d'erreur et l'objectif de la loi de commande est décrit comme la minimisation d'une certaine norme du signal d'erreur. Des exemples de normes sont l'énergie de l'erreur, l'amplitude de l'erreur ou la variance de l'erreur, l'objectif est de trouver une loi de commande stabilisante qui minimise l'effet de bruit ou de consigne dans le signal d'erreur. Pour minimiser l'effet du bruit ou du signal de consigne dans le signal d'erreur, le choix d'une norme des signaux est important. Par exemple, si l'objectif de commande est de minimiser le dépassement ou de rejeter l'effet de la perturbation dans le signal d'erreur, la norme l_1 est préférable. Dans ce cas une approche pire des cas pour le problème de commande est de trouver un correcteur tel que la norme l_1 de système entre la sortie et l'entrée soit minimale. Cependant, pour minimiser l'énergie du signal d'erreur, la minimisation de la norme H_∞ de la fonction de transfert nécessaire. Pour minimiser la variance du signal d'erreur (quand le signal de bruit est blanc) il est préférable de minimiser la norme H_2 de transfert entre la sortie et l'entrée qui minimise également l'énergie de la réponse impulsionnelle de cette fonction de transfert. L'approche mono-objectif apparaît bien dans la commande H_∞ , où le critère de performance de la norme H_∞ qui assure la stabilité robuste du système en boucle fermée, est indépendant de les autres performances, et aussi dans la commande H_2 qui assure la performance optimale de stabilité sur un critère unique dépend de la sortie de système étudié. Mais lorsque nous mixons ces objectifs concurrents, nous trouvons nouvelles approches traitent plusieurs objectifs et spécifications en même temps, appelée la commande multiobjectifs H_2 / H_∞ .

Le problème de contrôle multi-objectifs peut être défini comme un problème d'optimisation mixte de norme. Après avoir développer un certain nombre de synthèses mono-formelles : H_∞ , H_2 , LQG, l_1 ... etc, les automaticiens se sont intéressés aux méthodes mixant plusieurs formalismes. En effet pour décrire un problème possédant plusieurs contraintes, parfois antagonistes, le mono-formel se révèle bien souvent impuissant. Le contrôle multiobjectif

tend donc à intégrer le plus grand nombre de normes possible dans sa résolution. Nous intéressons dans ce chapitre par les deux types des normes les plus populaires : la norme H_2 (cas LQG) et la norme H_∞ , utilisés pour définir des objectifs concurrents.

Les méthodes multiobjectif peuvent être divisés en deux familles : les méthodes fréquentielles et les méthodes mixtes fréquentiel-temporel. On retrouve essentiellement dans les méthodes multiobjectifs fréquentielles le contrôle : H_2/H_∞ . Ce problème est majoritairement résolu par l'utilisation de formulations LMIs, et dans la catégorie des méthodes mixtes fréquentiel-temporel on retrouve les synthèses: l_1/H_∞ $l_1/H_2/H_\infty$, $l_\infty/H_2/H_\infty$...etc. Cette fois, encore on rencontre une majeure partie des formulations en LMIs, alors que les résultats spécifiques de la théorie des approches mixant H_2/H_∞ exposent des difficultés d'unification de ces objectifs. Les recherches montraient qu'on peut réinterpréter ce type de problèmes à un problème d'optimisation convexe, depuis des dix années passées l'optimisation convexe pénètre à tous les nouveaux domaines d'automatique, elle est connue par la programmation semi définie (SDP), qui considère comme un problème d'optimisation convexe utilise les formulations des inégalités matricielles linéaires (LMIs) pour contrôler plusieurs spécifications et contraintes convexes. De cette manière et à grâce les concepts des LMIs l'unification des contraintes d'optimisation multiobjectifs est devient possible.

Avec le développement de recherche sur les méthodes de résolution d'un problème d'optimisation multiobjectifs, il apparaît une famille des algorithmes appelés algorithmes évolutionnaires, parmi ces algorithmes nous intéresserons dans ce travail par :

« *Les Algorithmes génétiques multiobjectifs* » qui donnent des meilleures solutions pour ce type des problèmes. Les algorithmes génétiques sont des méthodes stochastiques basées sur une analogie avec des systèmes biologiques. Ils ont été introduits par Holland (1975) pour des problèmes d'optimisation complexe. Contrairement aux méthodes d'optimisation classique, ces algorithmes sont caractérisés par une grande robustesse et possèdent la capacité d'éviter les minimums locaux pour effectuer une recherche globale.

L'objectif de ce travail est traité le problème de commande multiobjectif en utilisant la technique des inégalités matricielles linéaires (LMIs) et les algorithmes génétiques.

Cette mémoire est organisée comme suivant : une introduction générale, cinq chapitres et une conclusion générale:

Le premier chapitre est concerné l'étude des inégalités matricielles linéaires (LMIs) et leurs applications en commande des systèmes. Dans le deuxième chapitre, nous présentons Les concepts de l'optimisation multicritères et ses différentes méthodes de résolution.

Le troisième chapitre, montre l'approche des algorithmes génétiques multiobjectifs qui résoudre le problème de commande multiobjectifs et le quatrième chapitre montre le contrôle multi-objectif H_2/H_∞ en utilisant l'approche LMI. Enfin nous renforçons notre étude par un exemple de simulation et une conclusion résume notre travail.

Etude des inégalités matricielles linéaires LMIs

1. Introduction :

Les inégalités matricielles linéaires sont utilisées pour résoudre plusieurs problèmes d'automatique, (problèmes d'optimisation en théorie du contrôle, identification de système,...) qui sont généralement difficiles à résoudre de façon analytique. L'intérêt des méthodes basées sur les LMIs vient du fait que ces dernières peuvent être résolues en utilisant la programmation convexe. Avec cette approche, on n'est plus limité aux problèmes ayant une solution analytique. En résolvant ces inégalités, on obtient un domaine de solutions faisables, c'est-à-dire de solutions satisfaisant ces LMIs, plus vaste que celui généré par la recherche de solutions analytiques. En utilisant le fait qu'une inégalité possède davantage de solutions qu'une équation, il est possible d'employer les degrés de liberté supplémentaires pour inclure d'autres objectifs que ceux initialement retenus.

Les notions des LMIs se retrouvent dans plusieurs travaux depuis de nombreuses années. Ainsi Lyapunov a conditionné la stabilité d'un système par LMI. Plus tard, Kalman, Yakubovich et Popov ont généralisé le résultat de stabilité proposé par Lyapunov. La terminologie des LMIs a été utilisée par Willems en 1971. En 1994, Nesterov et Nemirovski ont trouvé une solution pour résoudre les LMIs de manière efficace en utilisant des méthodes basées sur les points intérieurs.

Ce chapitre présente un ensemble des notions et propriétés concernant les inégalités matricielles linéaires LMIs et leurs applications en commande des systèmes.

2. Définition :(inégalité matricielle linéaire LMI) :

On appelle une inégalité matricielle linéaire notée (LMI) le problème suivant : étant données les matrices réelles, carrées et symétriques: $F_i = F_i^T \in \mathfrak{R}^{n \times n}$, $i = 0 \dots m$ et $x \in \mathfrak{R}^m$ telles que :

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (\text{I-1})$$

L'inégalité (I-1) implique que: $F(x)$ est une matrice définie positive c'est-à-dire :

$\forall z \in \mathfrak{R}^n$ et $z \neq 0$: $z^T F(x) z > 0$ De manière équivalente, la valeur propre la plus petit de $F(x)$ est positive.

Les matrices symétriques F_i sont fixées (connues) et $x = [x_1, x_2, \dots, x_m]^T$ est un vecteur de valeurs inconnues (variables). On dit que $F(x) > 0$ est une LMI affine des éléments de x .

Remarque :

L'inégalité (I-1) est une LMI stricte si $F(x)$ est seulement définie positive (non négative) autrement LMI est dite non stricte.

Le succès des LMIs vient du développement des méthodes dites *du point intérieur* qui permettent de résoudre ces problèmes de manière efficace [1].

3. Problème de faisabilité :

Le problème de faisabilité d'une LMI est le problème de trouver l'ensemble des points: $x \in C$ où $C = \{x \in \mathfrak{R}^n / F(x) > 0\}$ qui vérifient LMI: $F(x) > 0$ alors le problème $F(x) > 0$ est dit faisable (ou réalisable) et ces points appelées points faisables [2].

Exemple :

Les LMIs ne se présentent pas souvent directement sous la forme (I-1) prenons un exemple classique de l'automatique: la stabilité au sens de Lyapunov pour un système linéaire $\dot{x}(t) = Ax(t)$. Il s'agit de trouver une matrice réelle $P = P^T > 0$ de même dimensions que A telle que: $A^T P + PA < 0$

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

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (\text{I-2})$$

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

$$P = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix} \quad (\text{I-3})$$

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

$$x_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} > 0 \quad (\text{I-4})$$

L'inégalité de Lyapunov $A^T P + PA < 0$, peut se réécrire sous la forme suivante:

$$x_1 \begin{bmatrix} 2a_1 & a_2 \\ a_3 & 0 \end{bmatrix} + x_2 \begin{bmatrix} a_2 + a_3 & a_1 + a_4 \\ a_1 + a_4 & a_2 + a_3 \end{bmatrix} + x_3 \begin{bmatrix} 0 & a_2 \\ a_3 & 2a_4 \end{bmatrix} < 0 \quad (\text{I-5})$$

Cette inégalité est une LMI affine des éléments: x_1, x_2, x_3 [3].

4. Propriétés:

Parmi les propriétés les plus importantes des inégalités matricielles linéaires, on peut mentionner:

4.1. Propriété 1 : (LMIs multiples peuvent être écrites comme une seule LMI)

Parmi les propriétés remarquables des LMIs, la possibilité de regrouper plusieurs LMIs

$F_1(x) > 0, F_2(x) > 0, \dots, F_n(x) > 0$ en une seule LMI bloc diagonale [1]:

$$\begin{bmatrix} F_1(x) & 0 & 0 & \cdot & 0 \\ 0 & F_2(x) & 0 & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 0 & F_n(x) \end{bmatrix} > 0 \tag{I-6}$$

4.2. Propriété 2 : (La convexité)

La convexité est une propriété géométrique importante, qu'on trouve dans la théorie d'optimisation globale.

4.2.1. Définition : (Un ensemble convexe):

Un ensemble C est dit convexe si pour toutes les points : $(x_1, x_2) \in C$ et $0 < \lambda < 1$, alors:

$$(\lambda x_1 + (1 - \lambda)x_2) \in C$$

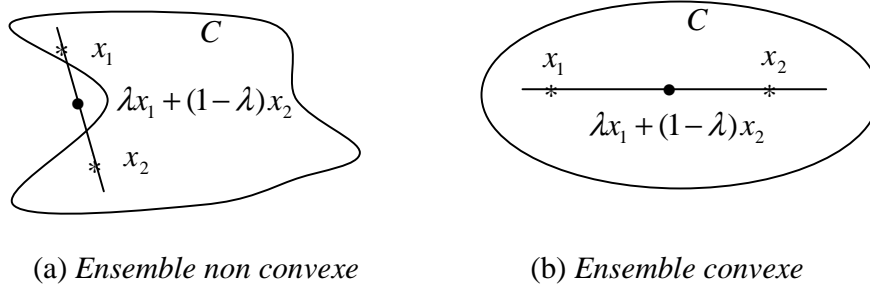


Figure I.1 Représentation d'un ensemble convexe et non convexe.

4.2.2. Définition : (Fonction convexe)

Soit une fonction f avec : $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ la fonction f est convexe si:

$\forall (x, y) \in \mathfrak{R}^n, 0 < \lambda < 1$ Alors :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \tag{I-7}$$

La fonction f est convexe si pour tous les pairs (x, y) : $f(\lambda x + (1 - \lambda)y)$ est toujours en bas de $\lambda f(x) + (1 - \lambda)f(y)$,

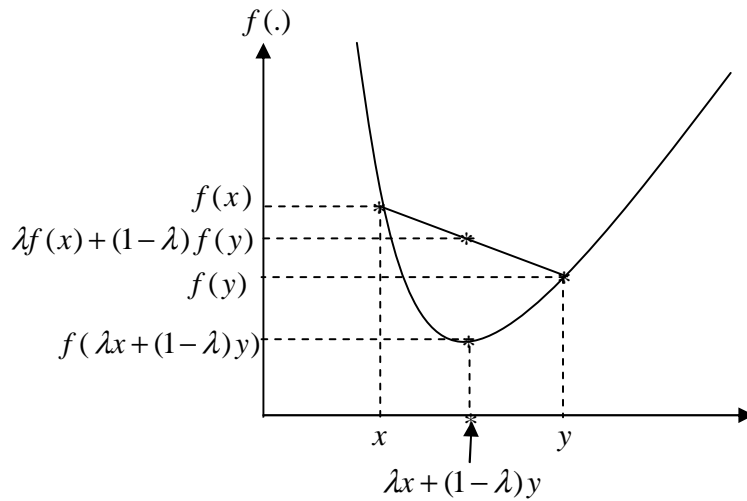


Figure I.2 Graphe d'une fonction convexe.

4.2.3. Définition: (contrainte LMI convexe)

Une importante propriété des LMIs est que l'ensemble : $C = \{x : F(x) > 0\}$ est convexe. C'est à dire LMI (I-1) définit un ensemble convexe sur la variable x [4].

Prenons : x et y deux vecteurs, avec: $F(x) > 0$, $F(y) > 0$ et $0 < \lambda < 1$ si:

$$F(\lambda x + (1-\lambda)y) > 0 \tag{I-8}$$

Donc on peut écrire [5] : $\lambda F(x) + (1-\lambda)F(y) > 0 \tag{I-9}$

4.3. Propriété 3 :(l'intersection de deux ensembles convexes)

Soit: $F(x) > 0$ et $G(x) > 0$ deux LMIs, liées respectivement avec les deux ensembles convexes suivants: $C_1 = \{x \in \mathfrak{R}^m / F(x) > 0\}$ et $C_2 = \{x \in \mathfrak{R}^m / G(x) > 0\}$

Alors l'intersection de C_1 et C_2 est définie par l'ensemble convexe suivant:

$$C_1 \cap C_2 = \left\{ x \in \mathfrak{R}^m / \begin{bmatrix} F(x) & 0 \\ 0 & G(x) \end{bmatrix} > 0 \right\} \tag{I-10}$$

Alors l'intersection de deux ensembles convexes donne un ensemble convexe [6].

5. Lemme du complément de Schur :

Le lemme du complément de Schur converti une classe des inégalités non linéaires à des inégalités matricielles linéaires LMIs convexes qui apparaissent régulièrement dans les problèmes de commande.

Les inégalités non linéaires convexes sont:

$$\begin{aligned} R(x) &> 0 \\ Q(x) - S(x)R^{-1}(x)S^T(x) &> 0 \end{aligned} \quad (\text{I-11})$$

Où: $Q(x) = Q^T(x)$, $R(x) = R^T(x)$ et $S(x)$ dépend d'une manière affine de x .

Le lemme du complément de Schur converti ces inégalités non linéaires convexes à une LMI équivalente [5] :

$$\begin{bmatrix} Q(x) & S(x) \\ S^T(x) & R(x) \end{bmatrix} > 0 \quad (\text{I-12})$$

Exemple :

Soit l'inégalité matricielle quadratique suivante :

$$\begin{aligned} R &> 0 \\ A^T P + PA + PBR^{-1}B^T P + Q &< 0 \end{aligned} \quad (\text{I-13})$$

Où: A , B , $Q = Q^T$, $R = R^T > 0$ sont des matrices données et $P = P^T$ est la variable.

On peut reformuler cette inégalité matricielle quadratique sous forme d'une inégalité matricielle plus simple, en utilisant le lemme du complément de Schur [7] :

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} > 0 \quad (\text{I-14})$$

6. Les applications des LMIs:

Nous allons donner quelques problèmes qui font appel aux LMIs :

6.1. Analyse de Stabilité au sens de Lyapunov:

La méthode de Lyapunov proposée en 1982 dans le cadre de l'étude de stabilité des systèmes linéaires. Etant donné un système LTI:

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0 \quad (\text{I-15})$$

Ce système est stable s'il existe une fonction $V(x)$ définie positive telle que sa dérivée est définie négative :

$$V(x) = x^T(t)Px(t) \quad (\text{I-16})$$

Où: P est une matrice symétrique et définie positive : $P = P^T > 0$

$$\dot{V}(x) = x^T (A^T P + PA)x \quad (\text{I-17})$$

La condition de stabilité consiste à trouver la matrice P qui vérifie l'inégalité matricielle :

$$A^T P + PA < 0 \quad (\text{I-18})$$

On peut écrire :

$$\begin{cases} V(x) > 0 \\ \dot{V}(x) < 0 \end{cases} \Leftrightarrow \begin{cases} P > 0 \\ A^T P + PA < 0 \end{cases} \quad (\text{I-19})$$

Donc il résulte :

$$\begin{bmatrix} P & 0 \\ 0 & -A^T P - PA \end{bmatrix} > 0 \quad (\text{I-20})$$

Le système (I-16) est stable lorsque l'inégalité matricielle (I-20) est faisable [5]

6.2. Problème de stabilité quadratique des systèmes incertains:

La notion de stabilité quadratique est le prolongement de la notion de stabilité de Lyapunov lorsque l'on considère des systèmes incertains, nous supposons que les matrices incertaines A du modèle d'état appartiennent à des ensembles compacts Ω .

Nous considérons le système linéaire incertain suivant :

$$\dot{x}(t) = A(t).x(t) \quad (\text{I-21})$$

Ce système est dit stable quadratiquement lorsqu'il existe une matrice $P = P^T > 0$ telle que quelle que soit la matrice A appartenant à l'ensemble Ω nous avons $\dot{V}(x) < 0$:

$$x^T (A^T(t)P + PA(t))x < 0 \quad (\text{I-22})$$

Nous pouvons alors montrer qu'une condition nécessaire et suffisante de stabilité quadratique du système incertain (I-21) est :

❖ Dans le cas des incertitudes bornées en norme avec l'ensemble compacts :

$$\Omega = \{A_0 + DF(t)E, \|F(t)\|_2 \leq 1\} \text{ avec : } \|F\|_2 = \lambda_{\max}(F^T F) \text{ où } \lambda_{\max} \text{ représente la valeur propre maximale de } F^T F$$

Qu'il existe une matrice $P = P^T > 0$ telle que l'inégalité matricielle :

$$A_0^T P + PA_0 + PDD^T P + E^T E < 0 \quad (\text{I-23})$$

Soit vérifiée, pour une matrice nominale A_0 et des matrices constantes D, E données.

❖ Dans le cas des incertitudes polytopiques avec : $\Omega = Co\{A_1, \dots, A_n\}$

Co : représente l'ensemble convexe et $\{A_1, \dots, A_n\}$ une série de matrices constantes données.

Qu'il existe une matrice $P = P^T > 0$ telle que l'inégalité matricielle :

$$A_i^T P + PA_i < 0, \quad \forall i = 1, \dots, n \quad (\text{I-24})$$

Soit vérifiée.

Notons que ces conditions sont uniquement suffisantes pour assurer la stabilité robuste du système incertain (I-21), c'est à dire sa stabilité pour toute incertitude admissible. De manière générale la stabilité quadratique implique la stabilité robuste mais l'inverse n'est pas vrai [8].

6.3. Stabilité des modèles Takagi-Sugeno (TS) :

L'étude de la stabilité des modèles Takagi-Sugeno s'effectue principalement en utilisant la méthode directe de Lyapunov. Soit le modèle Takagi-Sugeno continu suivant en régime libre :

$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t))A_i x(t) \quad (\text{I-25})$$

$$h_i(z(t)) \geq 0, \sum_{i=1}^r h_i(z(t)) = 1, \forall i \in \{1, \dots, r\} \quad (\text{I-26})$$

Les matrices $A_i \in \mathfrak{R}^{n \times n}$ représentent un ensemble de r modèles linéaires.

Les fonctions d'appartenance $h_i(\cdot)$ ont la propriété de somme convexe (I-26) et sont fonctions d'un vecteur $z(t) \in \mathfrak{R}^z$ appelé vecteur des prémisses.

La stabilité quadratique du modèle TS (I-25) revient à résoudre le problème suivant :
Trouver une matrice $P > 0$, telle que [9]:

$$A_i^T P + P A_i < 0, \forall z(t) \in \mathfrak{R}^z \quad (\text{I-27})$$

6.4. Problème de Stabilisation :

Les LMIs permettent également de générer des lois de commande stabilisantes. La condition nécessaire et suffisante de stabilisabilité du système commandé :

$$\dot{x} = A.x + Bu \quad (\text{I-28})$$

Par un retour d'état : $u = K.x$ est l'existence des matrices P et K telles que :

$$\begin{aligned} (A + BK)^T P + P(A + BK) &< 0 \\ P = P^T &> 0 \end{aligned} \quad (\text{I-29})$$

Cette inégalité est bilinéaire en P et K mais à l'aide du changement de variables $P = W^{-1}$ et $K = RW^{-1}$ elle devient une LMI en R et W :

$$\begin{aligned} AW + WA^T + BR + R^T B^T &< 0 \\ W = W^T &> 0 \end{aligned} \quad (\text{I-30})$$

6.5. Problèmes d'optimisation sous contraintes LMIs :

Beaucoup de problèmes d'Automatique et particulièrement les problèmes de contrôle des systèmes peuvent se formuler comme des problèmes d'optimisation sous contraintes LMIs. Et plusieurs problèmes sont mieux écrits en termes d'une simple ou multiple fonction objectif avec un ensemble des contraintes LMIs, car les problèmes d'optimisation convexe apparaissent

souvent en pratique. Ceci est la force de l'utilisation des formulations LMIs dans les applications réelles qui concernent les lois de commande des divers systèmes.

L'introduction des contraintes LMIs permet de définir un ensemble de problèmes d'optimisation suivants:

6.5.1. Programmation Semi Définie (SDP):

La programmation semi définie SDP appelé aussi problème d'optimisation LMI, est une généralisation de la programmation linéaire (LP), où l'égalité de contrainte remplace par une LMI, un problème SDP formulé comme suivant:

$$\begin{aligned} \text{Min } (c^T x) \\ F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \end{aligned} \quad (\text{I-31})$$

$x \in \Re^m$: vecteur des variables de décision.

c^T : vecteur ligne donnée

$F(x)$: contrainte LMI [5].

6.5.2. Problèmes des valeurs propres (Eigenvalue Problem EVP):

Un large nombre de propriétés de commande peuvent être calculés comme un problème de valeur propre (EVP) qui est le problème de minimisation de la valeur propre maximale : λ_{\max} d'une matrice: $A(x) > 0$ qui dépend affinement de la variable x , soumise à une contrainte LMI $F(x) > 0$.

Plusieurs tests d'analyse de performance, tel que le calcul de la norme H_{∞} peuvent être écrit sous forme d'un problème des valeurs propres (EVP) avec contrainte LMI.

On écrit la forme générale d'un problème des valeurs propres (EVP) comme suivant [5].

$$\begin{aligned} \text{Min } \lambda \\ \lambda I - A(x) > 0 \\ F(x) > 0 \end{aligned} \quad (\text{I-32})$$

6.5.3. Problèmes des valeurs propres généralisées (GEVP):

Le problème de valeur propre généralisée consiste à minimiser la plus grande valeur propre d'une paire de matrices $F(x)$ et $G(x)$ dépendant linéairement de la variable x , sous contraintes LMIs.

Le problème GEVP est exprimé par:

$$\begin{aligned} & \text{Min } \lambda && \text{(I-33)} \\ & \text{Tel que : } \lambda F(x) - G(x) > 0 \\ & && F(x) > 0 \text{ et } H(x) > 0 \end{aligned}$$

$F(x)$, $G(x)$ et $H(x)$ sont des matrices symétriques [10].

6.5.4. Minimisation du déterminant :

Le problème de minimisation du déterminant est exprimé par:

$$\begin{aligned} & \text{Min } \{ \log [\det(G^{-1}(x))] \} && \text{(I-34)} \\ & \text{Tel que : } F(x) > 0 \text{ et } G(x) > 0 \end{aligned}$$

Où $F(x)$ et $G(x)$ sont des matrices symétriques et affines en x [5].

Ces problèmes d'optimisation peuvent être résolus par différents types de méthodes :

- méthode des points intérieurs (méthode des centres, méthode primale-duale, méthode projective de Nemirovskii).
- Méthode des plans sécants.
- Méthode de l'ellipsoïde.
- Méthode du type simplex [10].

6.6. Problème de commande Linéaire Quadratique (LQ):

On parle de commande linéaire quadratique : LQ ou LQR (linear quadratic regulator). Le système est linéaire et la commande est quadratique.

Dans le cas d'un système linéaire à temps invariant LTI suivant :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad \text{(I-35)}$$

On écrit la loi de commande optimale à horizon infinie ($t_{final} \rightarrow +\infty$) par retour d'état comme suivant : $u(t) = -K.x(t)$ où: $K = -R^{-1}B^T P$ avec les matrices: $R = R^T > 0$ et $P = P^T > 0$ vérifie l'équation algébrique de Riccati (ARE) suivante :

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (\text{I-36})$$

Q : est une matrice symétrique fixée et définie positives [3].

Les équations algébriques de Riccati (ARE) sont excessivement utilisées dans la commande optimale, un résultat nécessitant une ARE peut être remplacé par un résultat équivalent où l'égalité est remplacée par une inégalité. Plus précisément ces contrôleurs optimaux peuvent être construits en calculant une matrice P symétrique définie positive qui satisfait l'inégalité algébrique de Riccati (ARI) :

$$A^T P + PA - PBR^{-1}B^T P + Q < 0 \quad (\text{I-37})$$

L'ARI est quadratique en P mais elle peut être exprimée comme une LMI en appliquant le lemme du complément de Schur [5]:

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} > 0 \quad (\text{I-38})$$

6.7. Problème de commande LQG (Quadratique Linéaire Gaussienne) :

Le problème d'atténuation des perturbations est traité par les méthodes de la commande optimale où une certaine mesure de l'amplitude de la sortie est minimisée soumise à des suppositions sur les perturbations. Une procédure standard connue sous le nom du problème de la commande LQG (quadratique linéaire Gaussienne) est souvent utilisée, où la somme des variances de sortie est minimisée soumise à la supposition que les perturbations sont caractérisées comme des processus stochastiques [5].

Soit le système LTI stable suivant :

$$G(s) : \begin{cases} \dot{x}(t) = Ax(t) + Bw(t) \\ y(t) = Cx(t) \end{cases} \quad (\text{I-39})$$

Où: w est un bruit blanc gaussien unitaire.

La commande LQG peut se mettre sous une forme particulière dite forme standard. Il s'agit alors de synthétiser un correcteur minimisant une norme H_2 sur les signaux de transfert.

La fonction du coût minimisée est donnée par :

$$J = \|G\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |G(j\omega)|^2 d\omega \quad (\text{I-40})$$

La norme H_2 est une mesure de la moyenne du carré de gain pris sur toutes les fréquences.

Voici la formulation LMI de la norme H_2 :

Soit la solution $P_0 = P_0^T \leq 0$ qui vérifiant l'équation de Lyapunov :

$$AP_0 + P_0A^T + BB^T = 0 \quad (\text{I-41})$$

Alors toute matrice $P > P_0$ vérifiant :

$$AP + PA^T + BB^T < 0 \quad (\text{I-42})$$

Le système $G(s)$ stable, vérifié : $\|G\|_2^2 < \gamma_2$ si et seulement s'il existe une matrice symétrique positive $P > 0$ vérifiant LMI (I-42) et :

$$\text{Trace}(CPC^T) < \gamma_2 \quad (\text{I-43})$$

L'ensemble des inégalités (I-42) et (I-43) constitue un système LMI qui peuvent formuler le problème LQG (H_2) comme suivant [3]:

$$\begin{aligned} \text{Min } \{ & \text{Trace}(CPC^T) \} \\ & AP + PA^T + BB^T < 0 \end{aligned} \quad (\text{I-44})$$

6.8. Commande Prédictive Robuste:

La commande prédictive du modèle est devenue la méthode la plus populaire des méthodes de conception des contrôleurs multivariables à cause de sa capacité de traiter les contraintes linéaires des processus variables. Les formulations de la commande prédictive en programmation standard linéaire et quadratique peuvent être écrites en termes des LMIs. Le principe de la commande prédictive du modèle est l'utilisation d'une fonction quadratique définie positive de

l'état pour borner ou limiter la fonction objective de performance elle est basée sur l'utilisation d'un modèle pour prédire par le comportement future du système sur un horizon du temps fini et puis utiliser le lemme du complément de Schur pour changer ces contraintes à une contrainte LMI [5].

Considérons le système linéaire variant dans le temps représenté par les équations d'état
Suivantes:

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) \end{aligned} \quad (\text{I-45})$$

Où chaque matrice d'état est arbitraire et se trouve dans un polytope $[A(k) \ B(k)] \in \Omega$. En considérant que les incertitudes sont définies par:

$\Omega = Co\{[A_1, B_1], [A_2, B_2], \dots, [A_L, B_L]\}$, avec : $[A(k) \ B(k)] \in \Omega$ Peut être exprimé comme:

$$[A, B] = \sum_{i=1}^L \lambda_i [A_i, B_i], \quad \forall \lambda_i \geq 0, \quad \sum_{i=1}^L \lambda_i = 1 \quad (\text{I-46})$$

Définissons $x(k|k)$ comme l'état d'un système incertain mesuré à l'instant d'échantillonnage k , $x(k+i|k)$ comme l'état du système à l'instant $k+i$ prédit à l'instant k , $u(k+i|k)$ comme la commande à l'instant $k+i$ calculée à l'instant k et W , et R sont des matrices de pondération définies positives. Pour ce problème de commande, l'objectif est de calculer la matrice F du retour d'état :

$$u(k+i|k) = Fx(k+i|k) \quad (\text{I-47})$$

Pour minimiser la borne supérieure d'une fonction objective quadratique à horizon infini

$$J(k) = \sum_{i=0}^{\infty} [x(k+i|k)^T W x(k+i|k) + u(k+i|k)^T R u(k+i|k)] \quad (\text{I-48})$$

On définit une fonction quadratique $V(x) = x^T(t) P x(t)$ où: $P = P^T > 0$ et γ sa borne supérieure.

$$\begin{aligned} & \underset{u(k+i|k), i \geq 0}{\text{Min}} \quad \{\gamma\} \\ & V(x(k/k)) \leq 0 \\ & \underset{[A(k+i) \ B(k+i)] \in \Omega, i \geq 0}{\text{Max}} \quad J(k) \leq V(x(k/k)) \end{aligned} \quad (\text{I-49})$$

Le système (I-45) est asymptotiquement stabilisé par un contrôleur de retour d'état, tout en minimisant une borne supérieure du coût quadratique $J(k)$ sous contraintes, si à chaque instant d'échantillonnage ($k = 0, 1, \dots, \infty$) il existe une matrice symétrique définie positive Q et une matrice Y solutions du problème suivant:

$$F = YQ^{-1} \quad (I-50)$$

Le problème de contrôle prédictif robuste peut être résolu à chaque itération par le problème LMI suivant [1]:

$$\left. \begin{array}{l} \text{Min}_{\gamma, Q, Y} \{ \gamma \} \end{array} \right\} \quad (I-51)$$

soumis à :

$$\begin{bmatrix} 1 & x^T(k|k) \\ x(k|k) & Q \end{bmatrix} \geq 0 \quad (I-52)$$

$$\begin{bmatrix} Q & QA_i^T + Y^T B_i^T & QW^{1/2} & Y^T R^{1/2} \\ A_i Q + B_i Y & Q & 0 & 0 \\ W^{1/2} Q & 0 & \gamma I & 0 \\ R^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \quad \forall i = 1, \dots, L. \quad (I-53)$$

6.9. Séquencement de gain :

Une nouvelle approche de conception des contrôleurs de séquencement de gain est de représenter le processus comme étant linéaire à paramètres variants (LPV) :

$$\begin{aligned} x(k+1) &= A(p(k))x(k) + B(p(k))u(k) \\ y(k) &= C(p(k))x(k) + D(p(k))u(k) \end{aligned} \quad (I-54)$$

Où les matrices de l'espace d'état sont des fonctions explicites d'un vecteur $p(k)$ de paramètres variants dans le temps. Nous supposons que le vecteur de paramètres est mesuré en temps réel et que ses composantes appartiennent a priori à des intervalles donnés.

Un processus LPV se réduit à un processus linéaire à temps variant pour une trajectoire donnée et il se réduit à un système linéaire à temps invariant pour un vecteur $p(k)$ de paramètres constants. Cette représentation du modèle forme la base pour un cadre théorique solide de

conception de contrôleurs à séquençement de gains utilisant les LMIs. C'est une pratique commune de supposer que les matrices de l'espace d'état sont des fonctions affines de $p(k)$ et que le paramètre variant en temps $p(k)$ varie dans un polytope. Alors le contrôleur à séquençement de gains (LPV) a la forme

$$\begin{aligned}\hat{x}(k+1) &= \hat{A}(p(k))\hat{x}(k) + \hat{B}(p(k))y(k) \\ u(k) &= \hat{C}(p(k))\hat{x}(k) + \hat{D}(p(k))y(k)\end{aligned}\tag{I-55}$$

Similaire à celle du processus. Le processus est supposé être capable de mesurer ou d'estimer $p(k)$ en ligne, pour que cette information soit utilisée par le contrôleur pour produire une performance améliorée par rapport aux contrôleurs qui n'exploitent pas une telle information. Les matrices du contrôleur qui garantissent la stabilité asymptotique globale et minimisent une norme 2 induite de la fonction objective de performance peuvent être calculées comme une EVP. Les LMIs sont développées utilisant une fonction quadratique de Lyapunov et une généralisation du lemme réel borné [5].

6.10. Problème de Commande H_∞ :

La synthèse H_∞ est un problème d'atténuation de perturbation, il consiste à minimiser l'effet d'une perturbation $w(t)$ sur le comportement du système.

Ce problème est représenté schématiquement par la figure (I.5) suivante :

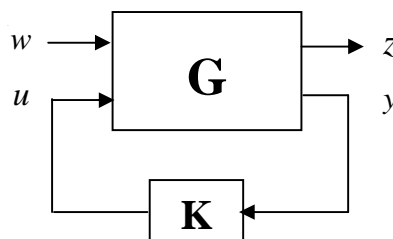


Figure I.3 Problème H_∞ standard

Soit $G(s)$ un système LTI représenté par les équations d'état

$$G(s) : \begin{cases} \dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) = C_2 x(t) + D_{21} w(t) \end{cases} \quad (\text{I-56})$$

$$x \in \mathfrak{R}^n, w \in \mathfrak{R}^{nw}, u \in \mathfrak{R}^{nu}, z \in \mathfrak{R}^{nz}, y \in \mathfrak{R}^{ny}$$

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

L'objectif d'un problème de commande H_∞ est de trouver un contrôleur $K(s)$ pour synthétiser une loi de commande qui stabilise le processus $G(s)$.

Le contrôleur $K(s)$ défini par :

$$K(s) : \begin{cases} \dot{x}_K(t) = A_K x_K(t) + B_K y(t) \\ u(t) = C_K x_K(t) + D_K y(t) \end{cases} \quad (\text{I-57})$$

Avec $G(s)$ et $K(s)$ définis ci dessus, la boucle fermée $N(s)$ admet la réalisation

$$N(s) : \begin{cases} \dot{x}_{cl}(t) = A_{cl} x_{cl}(t) + B_{cl} w(t) \\ z(t) = C_{cl} x_{cl}(t) + D_{cl} w(t) \end{cases} \quad (\text{I-58})$$

Le but est celui de trouver des matrices A_K, B_K, C_K et D_K telles que la norme H_∞ de la boucle fermée soit la plus petite possible, c'est à dire:

$$\gamma_{opt} = \text{Min} \gamma \quad (\text{I-59})$$

$$\text{tel que: } \|N(s)\|_\infty < \gamma$$

La résolution du problème H_∞ par LMIs, fondée sur l'utilisation des lemmes suivants [5] :

Lemme 1 (Valeur singulière maximale):

La valeur singulière maximale mesure le gain maximal d'un système multivariable, où les amplitudes des vecteurs d'entrée et de sortie sont quantifiées par la norme Euclidienne.

La valeur singulière maximale d'une matrice A qui dépend d'une manière affine de x est dénotée par $\bar{\sigma}(A(x))$ qui est la racine carrée de la plus large valeur propre de: $A(x)A^T(x)$

L'inégalité $\bar{\sigma}(A(x)) < 1$ est une contrainte convexe non linéaire de x , elle peut être écrite comme une LMI utilisant le lemme du complément de Schur :

$$\bar{\sigma}(A(x)) < 1 \Leftrightarrow A(x)A^T(x) < I \quad (\text{I-60})$$

$$\Leftrightarrow I - A(x)A^T(x) > 0 \quad (\text{I-61})$$

$$\Leftrightarrow \begin{bmatrix} I & A(x) \\ A^T(x) & I \end{bmatrix} > 0 \quad (\text{I-62})$$

Lemme 2 (systèmes réels bornés):

Le lemme des réels bornés peut être appliqué à la commande des systèmes linéaires et non linéaires, le résultat actuel est basé sur la représentation en variables d'état d'un système linéaire suivant:

$$\begin{cases} \dot{x} = Ax + Bu, & x(0) = 0 \\ y = Cx + Du \end{cases} \quad (\text{I-63})$$

Où: $A \in \mathfrak{R}^{n \times n}$, $B \in \mathfrak{R}^{n \times p}$, $C \in \mathfrak{R}^{p \times n}$ et $D \in \mathfrak{R}^{p \times p}$ sont données.

Supposons que A est stable et que (A, B, C) est minimal. La fonction de transfert matricielle est :

$$G(s) = C(sI - A)^{-1}B + D \quad (\text{I-64})$$

La performance du pire cas d'un système mesurée en termes de l'intégrale des erreurs carrées de l'entrée et de la sortie est quantifiée par la norme H_∞ :

$$\|G(s)\|_\infty = \sup_{\text{Re}(s) > 0} \{\bar{\sigma}(G(s))\} = \sup_{\omega \in \mathfrak{R}} \{\bar{\sigma}(G(j\omega))\} \quad (\text{I-65})$$

La norme H_∞ peut être écrite en termes d'une LMI. Pour cela on emprunte un résultat de la littérature de la commande robuste que la norme $\|G(s)\|_\infty \leq \gamma$ si et seulement si la valeur singulière maximale de D inférieure ou égale γ : $D^T D < \gamma^2 I$ et qu'il existe $P = P^T > 0$ tel que :

$$(A^T P + PA + C^T C) + (PB + C^T D)(\gamma^2 I - D^T D)(B^T P + D^T C) < 0 \quad (\text{I-66})$$

Le lemme du complément de Schur implique que cette inégalité algébrique de Riccati (ARI) est équivalente à l'existence de $P = P^T > 0$ tel que LMI suivante est valide [5].

$$\begin{bmatrix} A^T P + PA + C^T C & PB + C^T D \\ B^T P + D^T C & D^T D - \gamma^2 I \end{bmatrix} < 0 \quad (\text{I-67})$$

Donc $N(s)$ est stable de manière interne et de norme $H_\infty < \gamma$ si et seulement s'il existe une matrice symétrique $P > 0$ telle que [5]:

$$\begin{pmatrix} A_{cl}^T P + PA_{cl} & PB_{cl} & C_{cl}^T \\ B_{cl}^T P & -\mathcal{A} & D_{cl}^T \\ C_{cl} & D_{cl} & -\mathcal{A} \end{pmatrix} < 0 \quad (\text{I-68})$$

7. Conclusion :

Ce chapitre a été contenu quelques rappels sur les propriétés générales des inégalités matricielles linéaires LMIs, avec certain problèmes dépendent essentiellement par les formulations LMIs comme le problème des valeurs propres EVP et la programmation semi définie SDP, ainsi nous avons présenté aussi quelques applications des LMIs en commande et analyse des systèmes.

Nous nous intéressons dans ce mémoire à deux techniques importantes pour résoudre un problème de commande multiobjectifs, la première technique est les formulations LMIs qui apparaît depuis 1892, à grâce les études de Lyapunov sur l'analyse de stabilité asymptotique d'un Système linéaire, les formulations LMIs utilisent dans plusieurs applications tels que les lois de commande des divers systèmes ainsi, ils formulent les contraintes de problème d'optimisation multicritère, et la seconde technique est les Algorithmes Génétiques (AGs).

Etude des méthodes d'optimisation Multicritères

1. Introduction :

Les ingénieurs se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers. Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un problème d'optimisation, dans lequel on définit une fonction objectif, ou fonction Coût, que l'on cherche à minimiser (ou maximiser) par rapport à tous les paramètres concernés.

Il existe deux types de méthodes d'optimisation, la première est l'optimisation mono-objectif, qui se base sur la minimisation (ou la maximisation) d'une seule fonction objectif où le but est de trouver la meilleure solution appelée solution optimale qu'est facilement définie suivant une seule performance du problème étudié (temps de réponse, temps de montée, la robustesse, taux d'erreur,...etc). D'autre part, l'optimisation multi-objectif optimise simultanément plusieurs fonctions objectives qui sont souvent contradictoires, on cherche à trouver la meilleure solution suivant un ensemble de performance du problème (temps de réponse plus la robustesse, temps de réponse plus temps de montée plus la robustesse,...etc). Où le résultat d'un problème d'optimisation multicritère est généralement un assortiment de solutions, qui se distinguent par différents compromis réalisés entre les objectifs. Cet assortiment est connu par *Pareto-optimal*. Donc le but de l'optimisation multicritère, est d'obtenir les solutions de *Pareto*, par conséquent, à connaître l'ensemble des compromis possibles entre les objectifs. La résolution d'un problème d'optimisation multicritère, a conduit les chercheurs à proposer des méthodes de résolution de plus en plus performantes.

Dans ce chapitre, nous présentons les principes de base de l'optimisation multiobjectif, et nous donnerons aussi une présentation de différents méthodes d'optimisation multiobjectif.

2. Principaux concepts en optimisation:

Tout d'abord nous définissons les notions communes à n'importe quelle méthode d'optimisation multicritère :

2.1. Fonction objectif : Une fonction objectif est une fonction qui modélise le but à atteindre dans le problème d'optimisation sur l'ensemble des critères. Il s'agit de la fonction qui doit être optimisée. Elle est notée $F(x)$ de manière générale $F(x)$ est un vecteur :

$F(x) = [f_1(x), f_2(x), \dots, f_k(x)]$. Elle est aussi appelée : *critère d'optimisation, fonction coût, fonction d'adaptation, ou encore performance* [3].

2.2. Paramètres : Un paramètre du problème d'optimisation, est une variable qui exprime une donnée quantitative ou qualitative sur une dimension du problème: coût, temps, taux d'erreurs,...etc. [3]. Ces paramètres correspondent aux variables de la fonction objectif. Ils sont ajustés pendant le processus d'optimisation, pour obtenir les solutions optimales. On les appelle aussi variables d'optimisation, variables de conception ou de projet.

2.3. Vecteur de décision : Un vecteur de décision est un vecteur correspondant à l'ensemble des variables du problème, il est noté : $\vec{x} = [x_1, x_2, \dots, x_n]^T$ avec : n le nombre de variables ou dimension du problème et x_k la variable sur la dimension K .

2.4. Critère de décision : est un critère sur lequel sont jugés les vecteurs de décision pour déterminer le meilleur vecteur. Un critère peut être une variable du problème ou une combinaison de variables.

2.5. Contraintes : Une contrainte du problème est une condition que doivent respecter les vecteurs de décision du problème. Une contrainte est notée : $g_i(\vec{x})$ avec : $i = 1, \dots, q$
 q : le nombre des contraintes [4].

2.6. Espace de recherche : représentant l'ensemble des valeurs pouvant être prises par les variables.

2.7. Espace réalisable : représentant l'ensemble des valeurs des variables satisfaisant les contraintes. Dans la figure II.1, on a :

\underline{x}_i et \overline{x}_i , représentent la borne inférieure et supérieure pour x_i .

C : représente l'espace de recherche égal au produit cartésien des domaines des variables.

X : représente l'espace réalisable délimité par les contraintes. Cette figure illustre en dimension deux ($C = \mathbb{R}^2$) le fait que : $X = \{\bar{x} / g_i(\bar{x}) \leq 0, \bar{x} \in C\}$

L'espace réalisable X (délimité par des contraintes) est un sous-ensemble de C , l'espace de recherche. [11]

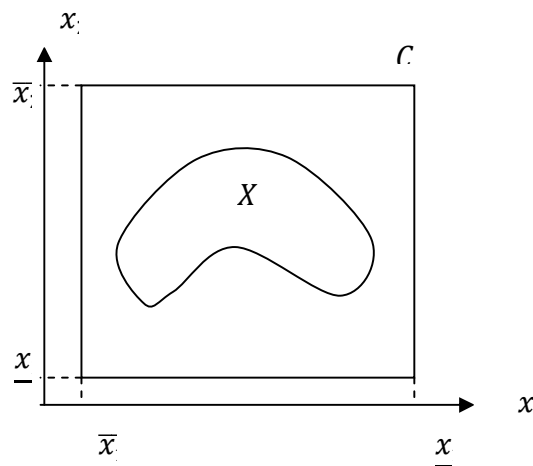


Figure II.1. Espace de recherche et espace réalisable.

2.8. Espace des objectifs : ensemble image de l'espace de recherche, déterminé par toutes les valeurs possibles des fonctions objectifs [4].

2.9. Minimum local : Un point $x \in X$ est un minimum local du problème si et seulement si :

$$\forall x' \in V(X), f(x) \leq f(x') \text{ où } V(X) \text{ définit un voisinage de } x.$$

2.10. Minimum global : Un point $x \in X$ est un minimum global du problème si et seulement si :

$$\forall x' \in X, f(x) \leq f(x') \text{ (il appartient alors à l'ensemble des minimiseurs) [11].}$$

3. Problème d'optimisation multicritère :

Le problème de l'optimisation multicritère a été étudié auparavant par les deux communautés suivantes : la communauté de control (Zadeh, 1963, Gembicki et Haimes 1975 Lin

1976, Giesy 1978, Tabak et al 1979) et la communauté de recherche opérationnelle (Cohon 1978) [4].

Les problèmes réels invoquent souvent de multiples mesures de performance ou objectifs, qui doivent être optimisés simultanément. En pratique, ceci n'est pas toujours possible car les objectifs peuvent être conflictuels, du fait qu'ils mesurent différents aspects de la qualité de la solution. Dans ce cas, la qualité d'un individu est décrite non pas par un scalaire mais par un vecteur. La performance, la fiabilité et le coût sont des exemples d'objectifs conflictuels [12]

Un problème d'optimisation multicritère consiste à trouver le vecteur de décision idéal \vec{x}^* tel que les contraintes $g_i(\vec{x})$ et $h_j(\vec{x})$ soient satisfaites et dont $\vec{F}(\vec{x}^*)$ est optimal [13].

En conséquence, le problème d'optimisation multicritère peut être défini formellement comme suit :

$$\text{Optimiser : } \vec{F}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \quad (m : \text{fonctions à optimiser}) \quad (\text{II-1})$$

$$\text{tel que : } \vec{x} \in X$$

$$X = \left\{ \vec{x} \left| \begin{array}{l} g_i(\vec{x}) \leq 0 \quad i = 1, \dots, q \\ h_j(\vec{x}) = 0 \quad j = 1, \dots, l \\ \vec{x} = [x_1 \dots x_n]^T \end{array} \right. \right\}$$

\vec{x} : est un vecteur de n variables de décisions.

$g_i(\vec{x})$, $h_j(\vec{x})$ sont les fonctions des contraintes qui déterminent le domaine X des solutions.

X : représente le domaine réalisable.

Ces objectifs multiples sont souvent concurrents où l'amélioration de l'un entraîne la détérioration de l'autre ou des autres, Ce conflit entre les objectifs s'explique facilement : de façon générale, des structures de haute performance tendent à avoir un coût élevé, alors que des dispositifs plus simples et usuellement peu coûteux auront des performances moindres. Selon les contraintes du cahier des charges, une solution intermédiaire (performance satisfaisante et coût acceptable) peut être optimale [13]. Dans les problèmes multiobjectifs, l'optimum n'est plus une simple valeur comme pour les problèmes mono-objectif, mais un ensemble de points, appelé l'ensemble des meilleurs compromis ou le front Pareto [12].

Tous les énoncés et définitions seront donnés dans le cadre de problèmes de minimisation. En effet un problème de maximisation peut être aisément transformé en un problème de minimisation en considérant l'équivalence suivante :

$$\text{Maximiser } \vec{F}(\vec{x}) \Leftrightarrow \text{Minimiser } -\vec{F}(\vec{x})$$

4. Méthodes d'optimisation multicritère :

Les approches de résolution des problèmes multiobjectifs peuvent être réparties en classes suivantes :

- ❖ Les méthodes Métaheuristiques.
- ❖ Les méthodes non Pareto.
- ❖ Les méthodes Pareto.
- ❖ Les méthodes hybrides.

4.1. Méthodes Métaheuristiques :

Les métaheuristiques sont un ensemble d'algorithmes d'optimisation visant à résoudre les problèmes d'optimisation difficiles. Elles sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires). Dans la section qui suit, on essayera de donner un bref aperçu sur les méthodes métaheuristiques de base les plus populaires [12].

4.1.1. La Méthode de Recuit Simulé :

Cette méthode a été inspirée du processus physique du recuit utilisé en métallurgie. Ce processus consiste en une suite de cycles de refroidissement lent pour obtenir un matériau homogène et de très bonne qualité. En effet, il alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tend à minimiser l'énergie du matériau. Cette méthode s'appuie sur l'algorithme de Metropolis [14] qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction $f(x)$ à minimiser deviendra l'énergie E du système où x est un état donné de la matière. Un paramètre fictif est également introduit : la température T du système. Partant d'une solution donnée nous générons une solution voisine en

utilisant une transformation qui change x en x' . Si celle-ci améliore le critère que l'on cherche à optimiser, c'est-à-dire : $\Delta f = f(x') - f(x) < 0$ on dit alors qu'on a fait baisser l'énergie du système, sinon elle la dégrade. En acceptant une solution améliorant le critère, nous tendons ainsi à chercher l'optimum dans le voisinage de la solution de départ, alors que l'acceptation d'une solution moins bonne, c'est-à-dire $\Delta f = f(x') - f(x) \geq 0$ permet alors d'explorer une plus grande partie de l'espace de solutions et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local. La solution voisine x' est acceptée avec une probabilité p . Si x' n'est pas meilleur que x , alors x' est accepté avec une probabilité suivante : $\exp\left[-\frac{\Delta f}{T}\right]$ [15].

4.1.2. Méthode de recherche Tabou :

La recherche Tabou est une métaheuristique développée par Glover. Elle est basée sur des idées simples, mais elle est néanmoins très efficace. Elle combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes pour surmonter l'obstacle des optima locaux. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire [16]. Dans une première phase, la méthode de recherche Tabou peut être vue comme une généralisation des méthodes d'amélioration locale. En effet, en partant d'une solution quelconque x appartenant à l'ensemble de solutions X , nous nous dirigeons vers une solution x' appartenant au voisinage $V(x)$ de x . L'algorithme explore donc itérativement l'espace de solutions X . Afin de choisir le meilleur voisin x' l'algorithme évalue la fonction objectif F en chaque point x' et retient le voisin qui améliore la valeur de F , ou celui qui la dégrade le moins [15].

4.1.3. Les Algorithmes Génétiques (AGs) :

Les algorithmes génétiques (AGs) ont été introduits par Holland comme un modèle de méthode adaptative. Ils ont été efficacement utilisés pour résoudre plusieurs problèmes d'optimisation multicritère.

L'algorithme génétique est une technique d'optimisation basée sur les concepts de la sélection naturelle de Darwin et la procréation selon les règles de Mendel. La sélection naturelle que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont les mieux adaptés à un environnement. Pour évaluer les individus d'une population on

utilise une *fonction d'évaluation* cette fonction est souvent une transformation de la fonction objectif, appelé aussi la fonction *fitness*, le résultat fournit par cette fonction va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante La sélection est suivie de la procréation, réalisée à l'aide de croisements et de mutations au niveau du patrimoine génétique des individus (ou "génotype") constitué d'un ensemble de gènes. Ainsi, deux individus "parents", qui se croisent transmettent une partie de leur patrimoine génétique à leurs descendants. Le génotype de l'enfant fait que celui-ci est plus ou moins adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, on sélectionne les individus les mieux adaptés, et l'augmentation du nombre de ces individus fait évoluer la population entière.

Dans les algorithmes génétiques, nous simulons le processus d'évolution d'une population. On part d'une population initiale de N solutions du problème représentées par des individus judicieusement choisis. Le degré d'adaptation d'un individu à l'environnement est exprimé par la valeur de la fonction coût $f(x)$, où x est la solution que l'individu représente. On dit qu'un individu est d'autant mieux adapté à son environnement, que le coût de la solution qu'il représente est plus faible ou plus important selon le(s) critère(s) d'optimisation choisi(s). Au sein de cette population, intervient alors la sélection au hasard d'un ou deux parents, qui produisent une nouvelle solution, à travers les opérateurs génétiques, tels que le croisement et la mutation. La nouvelle population, obtenue par le choix de N individus parmi les populations (parents et enfants), est appelée génération suivante. En itérant ce processus, on produit une population plus riche en individus mieux adaptés. [15].

Le chapitre 3 fera l'objet d'une présentation plus approfondie des algorithmes génétiques.

D'autres Métaheuristiques ont été adapté au cas d'optimisation multiobjectif et ont prouvé leur succès :

- ◆ Les colonies de fourmis.
- ◆ L'Optimisation par essaim de particules.
- ◆ Les réseaux de neurones artificiels [16].

4.2 Méthodes non Pareto:

Les méthodes non Pareto ne traitent pas le problème comme un véritable problème multiobjectif. Elles cherchent à ramener le problème initial à un ou plusieurs problèmes mono objectifs. Parmi ces approches nous citons :

4.2.1. Les méthodes Agrégées:

Cette approche, consiste aux nombreuses méthodes de résolution, parmi les méthodes qui utilisent cette approche, nous pouvons citer les méthodes suivantes :

- Méthodes d'agrégation par pondération.
- Méthode ε -contraintes.
- Méthode de but à atteindre.
- Méthode de min-max.
- Méthode de Goal programming.

4.2.1.1. Agrégation par pondération:

Dans cette approche, le but consiste à ramener le problème multicritère à un problème monocritère plus simple à traiter. Cette méthode est la plus simple des méthodes d'optimisation multi-objectif. La transformation que l'on effectue est la suivante :

$$\text{Minimiser } \sum_{i=1}^m w_i \cdot f_i(x) \quad (\text{II-2})$$

$$\text{Avec : } \vec{x} \in X, w_i \in [0, 1] \text{ et } \sum_{k=1}^m w_k = 1$$

X : représente le domaine réalisable.

w_i : Appelé le poids, est une pondération associée au critère, cette pondération permet d'exprimer des préférences sur les critères de décision [4].

Cette méthode est très efficace du point de vue algorithmique, mais elle ne permet pas de trouver les solutions enfermées dans des concavités. La figure II-2 illustre ce cas en dimension 2. En effet pour un vecteur de poids $\vec{w} = [w_1 \ w_2]^T$ fixé la valeur optimale atteignable pour la fonction objectif crée est : p . Les deux points Pareto optimaux trouvés sont A et C. En faisant varier le vecteur \vec{w} il est possible de trouver d'autres points Pareto optimaux. Seulement tous ces points se

trouveront sur les parties convexes de la surface de compromis. Il n'existe pas de valeur possible pour \vec{w} permettant de trouver par exemple le point B [11].

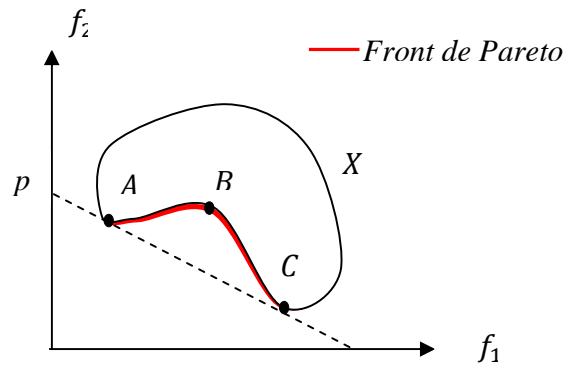


Figure II-2 Interprétation graphique de la méthode d'agrégation par pondération

Il existe une autre méthode d'agrégation appelée « l'Agrégation par L'intégrale de Choquet » c'est une deuxième technique pour l'agrégation de critères, plus sophistiquée que la première. Son intérêt est qu'elle permet de décrire des situations difficilement modélisables. Cette technique utilise une mesure floue (capacité) sur l'ensemble de critères $E_c = \{c_1, c_2, \dots, c_m\}$, qui représente l'importance des sous ensemble de critères dans la prise de décision. Il y a deux notions importantes pour l'analyse sémantique de l'intégrale de Choquet: La notion d'importance globale d'un critère et l'interaction entre les critères [13].

4.2.1.2. Méthode ε -contrainte :

Elle est aussi dite méthode du compromis. Elle transforme un problème d'optimisation multiobjectif en un problème d'optimisation mono-objectif de la façon suivante :

- ◆ Choisir un objectif à optimiser prioritairement.
- ◆ Choisir un vecteur de contraintes initiales ε .
- ◆ transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$$\begin{aligned} & \text{Minimiser } f_1(x) && \text{(II-3)} \\ & \text{Tel que : } f_2(x) \leq \varepsilon_2, \dots, f_m(x) \leq \varepsilon_m \\ & x \in X \end{aligned}$$

Cette approche a l'avantage par rapport à la précédente dans les problèmes non convexes, mais présente plusieurs inconvénients à savoir :

- ◆ la formulation des préférences utilisateur est délicate et nécessite une connaissance approfondie du problème de départ.
- ◆ Les contraintes rajoutées compliquent la résolution du problème [16].

4.2.1.3. Méthode Min-Max :

Cette méthode consiste à transformer le problème multiobjectif en un problème à un seul objectif où l'on cherche à minimiser l'écart relatif par rapport à un point de référence appelé but fixé par la méthode ou le décideur. Il existe plusieurs manières de caractériser la distance entre un point de référence (le but) et un autre, notamment à l'aide de normes. Une norme est définie de manière suivante :

$$L_r(F(x)) = \left[\sum_{i=1}^m |B_i - f_i(x)|^r \right]^{\frac{1}{r}} \quad (\text{II-4})$$

Les principales normes utilisées sont : $L_1(F(x)) = \sum_{i=1}^m |B_i - f_i(x)|$ la distance classique, et la norme $L_\infty(F(x)) = \max_{i \in \{1, 2, \dots, m\}} (B_i - f_i(x))$. C'est cette dernière qui est utilisée dans l'approche min-max appelée aussi approche de Tchebychev:

$$\left| \begin{array}{l} \text{Minimiser : } \max(B_i - f_i(x)) \\ \text{tel que : } g_q(x) \leq 0 \\ \text{Avec : } x \in \mathfrak{R}^n, g_q(x) \in \mathfrak{R}^q, F(x) \in \mathfrak{R}^m \end{array} \right. \quad (\text{II-5})$$

Dans cette approche, le point de référence joue un rôle fondamental, s'il est mal choisi la recherche peut s'avérer être très laborieuse. La figure II-3 illustre, en dimension 2, le cas d'une recherche avec un but B fixé. Il est clair que l'approche permet de traiter les problèmes non convexes à condition que le point de référence soit choisi judicieusement [15].

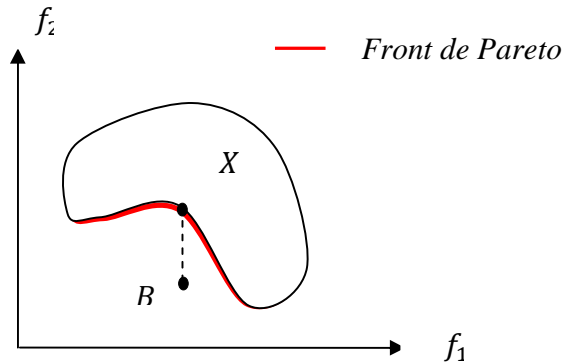


Figure II-3 Interprétation graphique de l'approche Min-Max

4.2.1.4. Méthode de but à atteindre :

On définit un ensemble de buts qu'on espère atteindre pour chaque fonction objectif. L'algorithme tente de minimiser l'écart entre la solution courante et ses buts. Cette approche, comme celle de min-max, utilise un point de référence pour guider la recherche. Mais elle introduit aussi une direction de recherche, si bien que le processus de résolution devra suivre cette direction. A la différence de l'approche min-max, qui utilise des normes pour formaliser la distance au point de référence, l'approche du but à atteindre utilise des contraintes, comme de l'approche ϵ -contrainte, pour déterminer la position du point de référence. L'écart par rapport à ce but est contrôlé grâce à la variable λ introduite à cet effet :

$$\begin{array}{l}
 \text{Minimiser : } \lambda \\
 \text{Tel que :} \\
 \quad f_i(x) - w_i \cdot \lambda \leq B_i \\
 \text{Avec :} \\
 \quad g_q(x) \leq 0 \\
 x \in \mathfrak{R}^n, g_q(x) \in \mathfrak{R}^q, F(x) \in \mathfrak{R}^m
 \end{array}
 \tag{II-6}$$

Ainsi en minimisant λ et en vérifiant toutes les contraintes, la recherche va s'orienter vers le but B et s'arrêter sur le point A faisant partie de la surface de compromis (voir l'illustration en 2 dimensions de la figure II-4). Cette approche permet comme l'approche par ϵ -contrainte et l'approche min-max, de trouver les parties non convexes des fronts Pareto.

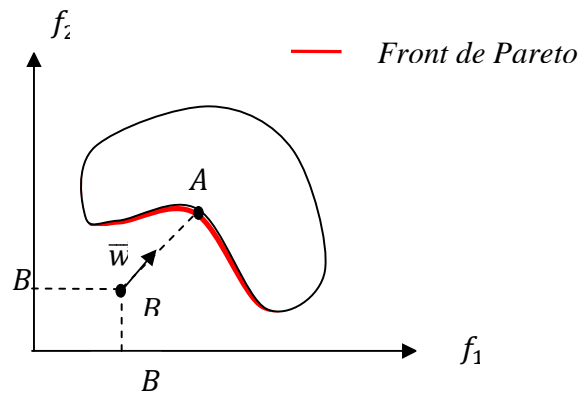


Figure II-4 Interprétation graphique de l'approche but à atteindre

Cependant, cette approche comme les précédentes, doit être itérée plusieurs fois dans le but d'obtenir un ensemble de points Pareto optimaux. Les paramètres \vec{w} et B doivent être bien choisis par l'utilisateur. Bien que ces paramètres permettent une grande flexibilité de la recherche (orientation et but), s'ils sont mal choisis, ils peuvent, dans certains cas extrêmes, donner des résultats non cohérents [15].

4.2.1.5 Méthode de Goal Programming :

Dans cette méthode le décideur fixe un but T_i à atteindre pour chaque objectif f_i . Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre :

$$\min \sum_{i=1}^m |f_i(\vec{x}) - T_i| \quad (\text{II-7})$$

T_i : représente la valeur à atteindre pour le $i^{\text{ème}}$ objectif.

Nous pouvons reprendre la critique faite pour la somme pondérée. La méthode est facile à mettre en œuvre mais l'efficacité de la méthode dépend de la définition des poids et des objectifs à atteindre. Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs but(s) T_i non réalisable(s) [17].

4.2.2. Méthode lexicographique :

Fourman a proposé une méthode dans laquelle les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite, l'optimum est obtenu en minimisant tout d'abord la fonction objectif la plus importante puis la deuxième et ainsi de suite.

Soient les fonctions objectifs f_i avec $i = 1, \dots, k$, supposons un ordre tel que : $f_1 \succ f_2 \succ \dots \succ f_k$ Il faut :

$$\left| \begin{array}{l} \text{Minimiser } f_1(x) \\ \text{avec : } g_j(x) \text{ satisfait : } j = 1, \dots, m \end{array} \right. \quad (\text{II-8})$$

Soit x_1^* , la meilleure solution trouvée avec : $f_1^* = f_1(x_1^*)$ f_1 devient alors une nouvelle contrainte.

L'expression du nouveau problème est donc :

$$\left| \begin{array}{l} \text{Minimiser } f_2(x) \\ \text{avec : } g_j(x) \text{ satisfait : } j = 1, \dots, m \text{ et : } f_1(x) = f_1^* \end{array} \right. \quad (\text{II-9})$$

Soit x_2^* la solution de ce problème. Le $i^{\text{ème}}$ problème sera le suivant :

$$\left| \begin{array}{l} \text{Minimiser } f_i(x) \\ \text{avec : } g_j(x) \text{ satisfait } j = 1, \dots, m, \text{ et : } f_1(x) = f_1^*, f_2(x) = f_2^*, \dots, f_{i-1}(x) = f_{i-1}^* \end{array} \right. \quad (\text{II-10})$$

La procédure est répétée jusqu'à ce que tous les objectifs soient traités. La solution obtenue à l'étape k sera la solution du problème. L'inconvénient essentiel de cette méthode est la grande importance attribuée aux objectifs classés en premier. La meilleure solution f_1^* trouvée pour l'objectif le plus important va faire converger l'algorithme vers une zone restreinte de l'espace d'état et enfermer les points dans une niche (ensemble d'individus situés dans un espace restreint) [17].

4.2.3. Vector evaluated genetic algorithm (VEGA):

En 1985 Schaffer propose une extension d'un algorithme génétique simple pour la résolution d'un problème multiobjectif. Cette méthode est appelée Vector Evaluated Genetic Algorithm. La seule différence avec un algorithme génétique simple est la manière dont s'effectue la sélection. L'idée est simple. Si nous avons K objectifs et une population de N individus, une sélection de N/K individus est effectuée pour chaque objectif. Ainsi K sous-populations vont être créées, chacune d'entre elles contenant les N/K meilleurs individus pour un objectif particulier. Les K sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille N . Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation).

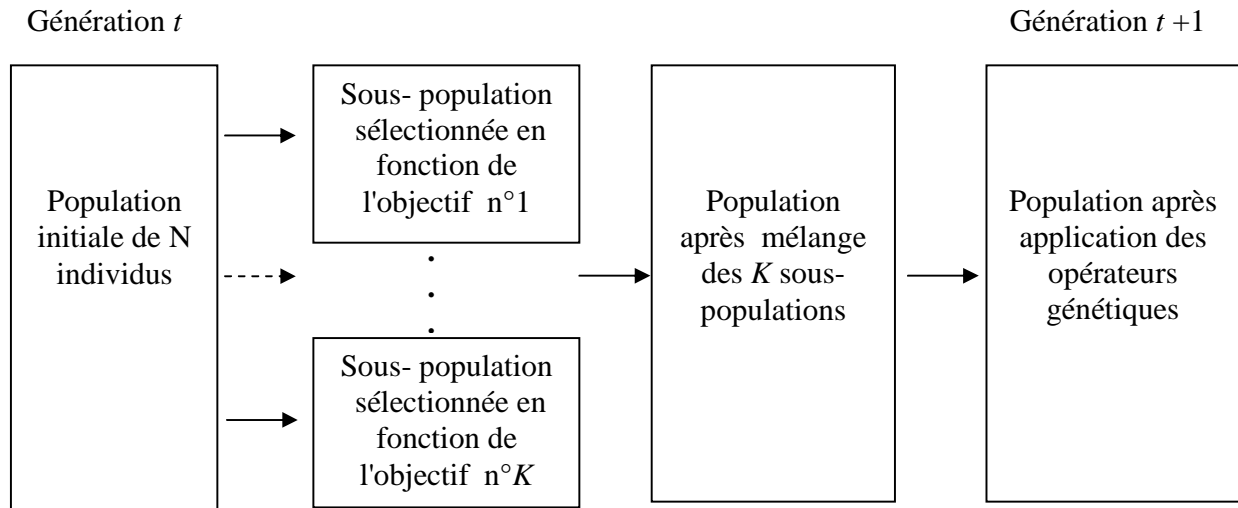


Figure II.5 : Schéma de fonctionnement de VEGA

La méthode VEGA a tendance à créer des sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. L'évolution de la population favorise l'apparition des espèces. En effet, comme la méthode de sélection ne tient compte que d'un seul objectif, elle privilégie les individus qui obtiennent une bonne performance pour cet objectif. Dès lors ces individus ne seront sélectionnés que lorsqu'on effectuera la sélection sur cet objectif. Les individus que Schaffer appelle les individus "milieu", parce qu'ayant une performance générale acceptable mais ne possédant aucun critère fort, vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Cette disparition entraîne la spécialisation des individus pour

chaque objectif. Ce résultat est contraire au but initial de la méthode qui était de trouver un compromis entre les différents critères [17].

4.3. Les approches Pareto :

Les approches Pareto ne transforment pas les objectifs du problème, ceux-ci sont traités sans aucune distinction pendant la résolution.

Dans les paragraphes suivants, nous définissons tout d'abord la notion de dominance au sens de Pareto, la frontière de Pareto, Pareto optimal. Ensuite, nous présentons les techniques évolutionnaires utilisant cette notion.

4.3.1. La dominance au sens de Pareto :

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg pour résoudre les problèmes multiobjectifs. Il suggère d'utiliser le concept d'optimalité de Pareto pour respecter l'intégralité de chaque critère car il refuse de comparer a priori les valeurs de différents critères. L'utilisation d'une sélection basée sur la notion de dominance de Pareto va faire converger la population vers un ensemble de solutions efficaces. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte une aide précieuse au décideur.

Vilfredo Pareto est un mathématicien italien (Pareto, 1896). Il a posé les bases de la solution d'un problème économique multiobjectif : « Dans un problème multiobjectif, il existe un équilibre tel que l'on ne peut améliorer un critère sans détériorer au moins un des autres ». Cet équilibre est appelé *optimum Pareto*. Donc une solution x est dite *Pareto optimale* si elle n'est dominée par aucune autre solution appartenant à l'espace réalisable X . Ces solutions sont appelées solutions *non dominées*, ou *non inférieures*. [17].

Définition (*La dominance au sens de Pareto*) : Considérons un problème de minimisation Soient : $u = [u_1, \dots, u_n]^T$ et $v = [v_1, \dots, v_n]^T$ deux vecteurs de décision. On dit que le vecteur de décision u domine le vecteur v (dénote $u \leq v$), si et seulement si :

$$\forall i \in \{1, 2, \dots, k\}, f_i(u) \leq f_i(v) \wedge \exists i \in \{1, 2, \dots, N\}, f_i(u) < f_i(v) \quad (\text{II-11})$$

Dans le processus d'optimisation multi-objectif, le concept de dominance du Pareto est utilisé afin de comparer et ranger le vecteur de variables des décisions :

u domine v dans le sens du Pareto, signifie que $F(u)$ est mieux que $F(v)$ pour tous les objectifs, et il y a au moins une fonction objectif pour laquelle $F(u)$ est strictement meilleure que $F(v)$ [12]

Définition (Pareto optimal) : Soit $x = [x_1, \dots, x_n]^T$ un vecteur de décision avec : $x \in X$ (L'espace réalisable) x est dit *Pareto optimal*, s'il n'existe pas une solution y domine x . Une solution Pareto optimal appelée aussi : solution efficace, non inférieure ou non dominée solution.

Définition (front de Pareto): Le front (frontière) de Pareto est l'ensemble des solutions Pareto optimales qui sont composées des points, ne sont dominés par aucun autre Le front de Pareto appelé aussi surface de compromise ou l'ensemble des solutions efficaces [13].

La figure II.6, représente le Front du Pareto pour un problème de minimisation et maximisation de deux fonctions objective [12]

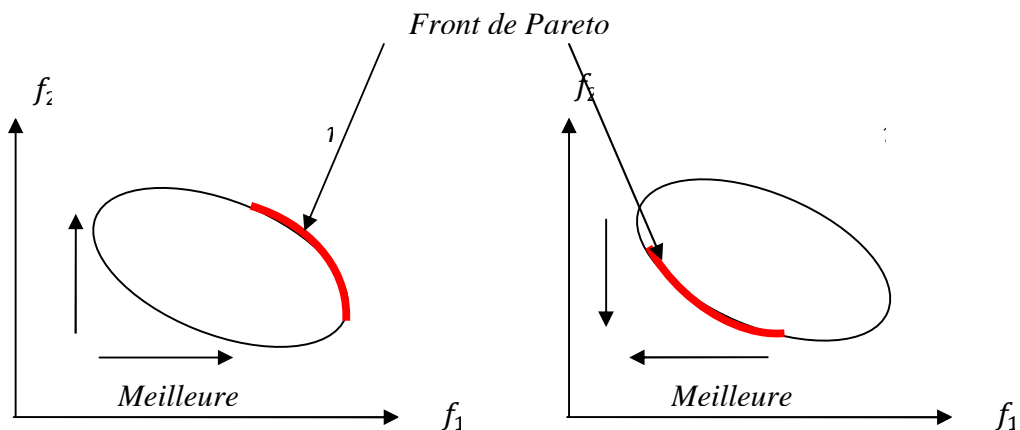


Figure II.6 le front de Pareto.

4.3.2. Exemple : Pour mieux expliciter les notions précédemment définies, considérons l'exemple suivant :

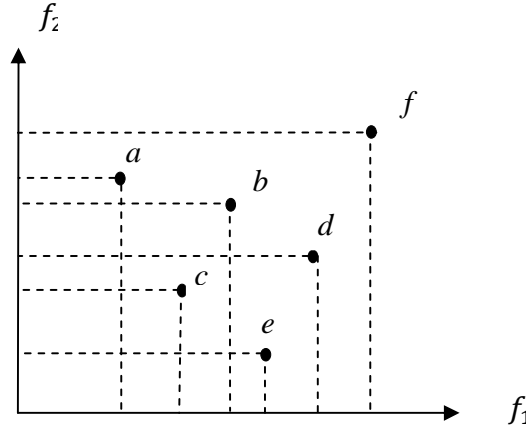


Figure II-7 Exemple de dominance et d'optimalité au sens de Pareto

Soient six vecteurs de décision :

$$a = (a_1, a_2), b = (b_1, b_2), c = (c_1, c_2), d = (d_1, d_2), e = (e_1, e_2), f = (f_1, f_2)$$

Soit la fonction objectif $F(x) = [f_1(x), f_2(x)]$. La représentation graphique de l'image de ces six vecteurs par la fonction F est fournie dans la figure II-3. Dans cet exemple on peut écrire :

- f : est dominé par tous les autres vecteurs.
- b : est dominé uniquement par c .
- d : est dominé par c et e .
- a, c et e : ne sont dominés par aucun vecteur.

Par conséquent les solutions (a, c, e) sont *Pareto optimaux*. Donc le *front de Pareto* est l'ensemble : $\{a, c, e\}$ [4].

4.3.3. Multiple Objectives Genetic Algorithm (MOGA) :

Cet algorithme, proposé par Fonseca et Fleming (1993), utilise la notion de dominance pour ranger les individus de la population. Il diffère de l'algorithme génétique standard uniquement dans la manière dont la fitness est assignée pour chaque solution. Pour démarrer l'algorithme, les relations de domination sont d'abord calculées pour chaque solution. Puis, pour une solution i , un rang égal à un plus le nombre de solutions n_i qui dominent la solution i est attribué. Une fitness

est ensuite attribuée à chaque solution en fonction de son rang, les individus avec les rangs les plus faibles ayant les meilleures fitness. Afin de maintenir la diversité entre les solutions non dominées, les auteurs utilisent une fonction de partage (*Sharing*) [18].

4.3.3.1. Maintenir la diversité :

Maintenir un certain degré de diversité dans la population d'un algorithme évolutionnaire consiste à éviter que la population ne converge prématurément vers une petite zone de l'espace de recherche ou de l'espace des objectifs. En effet, s'il n'existe pas de mécanisme de contrôle de la diversité, les opérations de sélection vont privilégier trop vite certains individus meilleurs à cette étape de la recherche. Cette convergence prématurée a comme effet de limiter la recherche à un sous-ensemble plus restreint de l'espace de recherche, qui peut ne contenir aucune solution optimale [11].

4.3.3.2. Le sharing :

Le sharing consiste à ajuster la fitness des individus pour éviter qu'ils se concentrent dans une niche principale (optimum globale). La technique de partage de la fitness (fitness sharing), introduite par Goldberg et Richardson, réduit la fitness de chaque individu d'un facteur correspondant environ au taux d'agrégation de la population autour de son voisinage [18].

Pour déterminer les bornes du domaine ouvert autour de l'individu choisi, on définit une distance maximale, appelée : σ_{shar} au delà de laquelle les individus ne seront plus considérés comme faisant parti du domaine ouvert.

La distance séparant deux individus i et j est calculée grâce à la fonction $d(i, j)$.

La valeur d'adaptation $F(i)$ d'un individu $i \in P$ (populations) est égale à son cout $F'(i)$ divisé par sa valeur de niche :

$$F(i) = \frac{F'(i)}{\sum_{j \in P} Sh(d(i, j))} \quad (\text{II-12})$$

Où la fonction Sh est définie comme suit [11]:

$$Sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i, j)}{\sigma_{\text{share}}}\right)^2 & \text{si } d(i, j) < \sigma_{\text{share}} \\ 0 & \text{sinon} \end{cases} \quad (\text{II-13})$$

La figure II.8 montre deux exemples de répartition de populations dans le cas d'une fonction multimodale: le premier sans sharing et le deuxième avec sharing.

La méthode permet d'obtenir des solutions de bonne qualité et s'implante facilement. Toutefois, les performances sont très dépendantes de la valeur du paramètre σ_{shar} utilisé dans le sharing [18].

4.3.4. Non dominated Sorting Genetic Algorithm (NSGA):

Dans l'algorithme NSGA proposé par Srinivas et Deb (1993), le calcul de fitness s'effectue en divisant d'abord la population en plusieurs fronts en fonction du degré de dominance au sens de Pareto de chaque individu. Les individus non dominés de la population courante constituent le premier front de Pareto. On attribue alors à tous les individus de ce front la même valeur de fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité de la population, il est nécessaire d'appliquer une fonction de partage sur cette valeur. En suite, ce premier groupe d'individus est temporairement supprimé de la population. On recommence cette procédure jusqu'à l'identification des solutions du deuxième front. La valeur factice de fitness attribuée à ce second groupe est inférieure à la plus petite fitness, après application de la fonction de partage sur le premier front. Ce mécanisme est répété jusqu'à ce que l'on ait traité tous les individus. L'algorithme se déroule ensuite comme un algorithme génétique standard. Grâce à sa procédure d'assignement de fitness basée à la fois sur la notion de dominance et la fonction de partage, le NSGA semble le plus approprié à maintenir la diversité de la population et à répartir plus efficacement les solutions sur le front de Pareto. Néanmoins, cet algorithme présente quelques insuffisances en raison de sa complexité de Calcul et de sa sensibilité au choix de la valeur σ_{shar} [18].

4.3.5 Niche Pareto Genetic Algorithm (NPGA):

Cette méthode proposée par Horn et Nafpliotis (1994) utilise une sélection par tournoi en se basant sur la notion de dominance de Pareto. Le NPGA exécute les mêmes étapes que l'AG standard, la seule chose qui diffère étant la méthode de sélection. A chaque tournoi, deux individus candidats A et B sont pris au hasard dans la population courante. Au lieu de limiter la comparaison aux deux individus (comme c'est le cas pour l'AG standard), une sous population (ou ensemble de comparaison) de taille t_{dom} est également choisie au hasard. Les deux candidats

sélectionnés sont comparés à chaque individu du sous-groupe. Si l'un des candidats est dominé par l'ensemble de comparaison et le second ne l'est pas, ce dernier est alors positionné dans la population suivante. Dans les autres cas, une fonction de partage est appliquée pour choisir le candidat gagnant. Le paramètre t_{dom} permet de contrôler la pression de sélection ou de dominance. L'algorithme NPGA est considéré comme étant l'algorithme le plus rapide parmi les approches précédentes car à chaque génération la comparaison n'est appliquée que sur une portion de la population. Le principal inconvénient de cet algorithme est qu'il nécessite, en plus de spécifier le paramètre de sharing σ_{shar} . Un autre paramètre supplémentaire qui est la taille du tournoi t_{dom} [18].

4.3.6. Non dominated Sorting Genetic Algorithm-II (NSGA-II):

Toutes les méthodes que nous venons de présenter ne conservent pas leurs solutions Pareto-optimales trouvées au cours des générations. Elles sont dites non élitistes. Pour résoudre cette difficulté, de nouvelles techniques ont été appliquées. Nous avons choisi de présenter uniquement le NSGA-II. En proposant le NSGA II, le chercheur Deb (Deb et al. 2002) a tenté de résoudre toutes les critiques faites sur NSGA: non élitiste, complexité de calcul et utilisation de sharing qui implique le réglage d'un ou plusieurs paramètres. Dans cet algorithme, à chaque génération t une population de parents (P_t) de taille N et une population d'enfants (Q_t) de même taille sont assemblées pour former une population (R_t) de taille $2N$, Cet assemblage permet d'assurer l'élitisme. La population (R_t) est ensuite répartie en plusieurs fronts (F_1, F_2, \dots) par une procédure de tri, plus rapide que celle proposée dans la première version de NSGA. Une nouvelle population parent (P_{t+1}) est formée en ajoutant les fronts au complet (premier front F_1 , second front F_2, \dots etc) tant que ce ceux-ci ne dépassent pas N . Si le nombre d'individus présents dans (P_{t+1}) est inférieur à N , une procédure de crowding est appliquée. La figure II.8 illustre le principe de fonctionnement de NSGA-II [18].

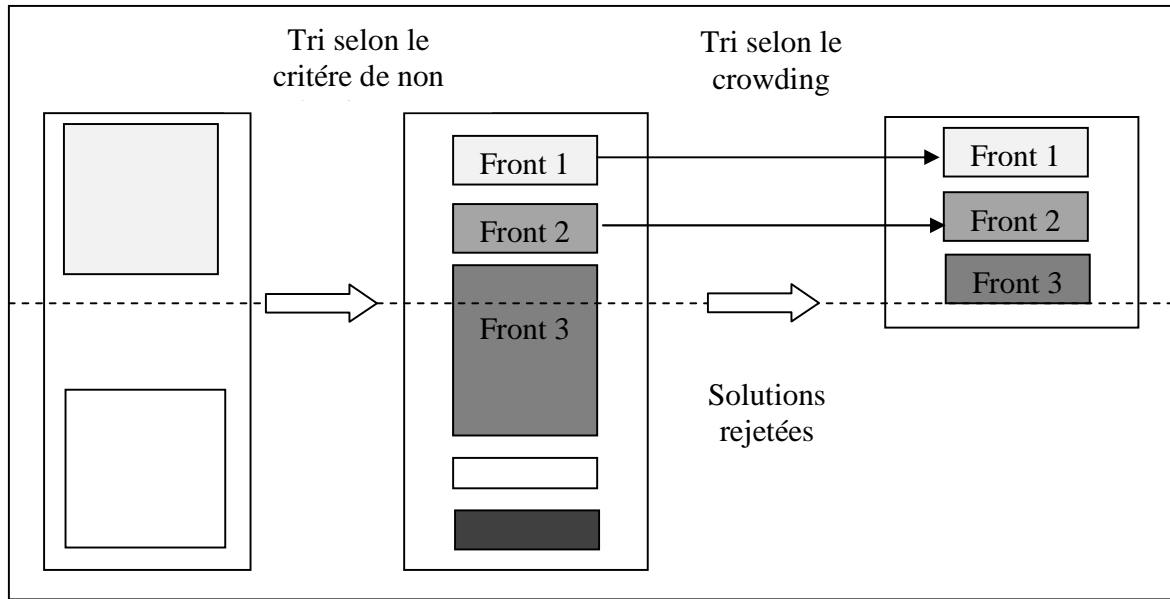


Figure II.8 : Principe de fonctionnement de NSGA-II.

4.3.6.1. Procédure de Crowding :

Une procédure de crowding, basée sur un calcul de distance (distance de crowding) qui ne nécessite aucun paramétrage et qui est également d'une complexité algorithmique moindre que celle de sharing. La distance de crowding d'une solution particulière i se calcule en fonction du périmètre de l'hypercube ayant comme sommets les points les plus proches de i sur chaque objectif. Sur la figure II.9, est représenté l'hypercube en deux dimensions associé au point i . Le calcul de la distance de crowding nécessite, avant tout, le tri des solutions selon chaque objectif, dans un ordre ascendant. Ensuite, pour chaque objectif, les individus possédant des valeurs limites se voient associés une distance infinie. Pour les autres solutions intermédiaires, on calcule une distance de crowding égale à la différence normalisée des valeurs des fonctions objectifs de deux solutions adjacentes. Ce calcul est réalisé pour chaque objectif. La distance de crowding d'une solution est obtenue en sommant les distances correspondantes à chaque objectif [18]

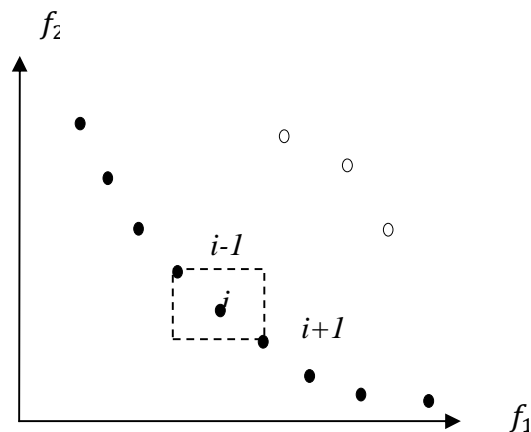


Figure II.9: Distance de Crowding (les points noirs sont des Solutions appartenant au même front)

4.3.7. Strength Pareto Evolutionary Algorithm (SPEA):

En 1998 Zitzler et Thiele proposent une nouvelle méthode d'optimisation multiobjectif qui possède les caractéristiques suivantes:

- ◆ Utilisation du concept de Pareto pour comparer les solutions.
- ◆ Un ensemble de solutions Pareto-optimales est maintenu dans une population externe appelée archive
- ◆ La fitness de chaque individu est calculée par rapport aux solutions stockées dans l'archive
- ◆ Toutes les solutions de l'archive participent à la sélection.
- ◆ Une méthode de clustering est utilisée pour réduire l'ensemble de Pareto sans supprimer ses caractéristiques
- ◆ Une nouvelle méthode de niche, basée sur Pareto, est utilisée afin de préserver la diversité. L'avantage essentiel est qu'elle n'exige pas de réglage de paramètres de sharing [17].

4.3.7.1. Fonctionnement général :

Le passage d'une génération à une autre commence par la mise à jour de l'archive. Tous les individus non dominés sont copiés dans l'archive et les individus dominés déjà présents sont supprimés. Si le nombre d'individus dans l'archive excède un nombre donné, on applique une technique de *clustering* pour réduire l'archive. Ensuite la fitness de chaque individu est mise à jour avant d'effectuer la sélection en utilisant les deux ensembles. Pour terminer, on applique les

opérateurs génétiques de modification. La méthode SPEA implémentée par Zitzler et Thiele est l'illustration même d'un algorithme évolutionnaire élitiste. SPEA maintient un archive externe contenant les meilleurs fronts de compromis rencontré durant la recherche [17].

4.3.7.2. Réduction par clustering :

Lorsque le nombre d'individus de l'archive externe est grand, les performances de l'algorithme peuvent se dégrader significativement. En effet, le nombre d'individus influe sur le calcul de la valeur d'adaptation, celui-ci devient moins fiable, et peut tromper la recherche en la focalisant, par exemple, sur des zones déjà explorées. Pour pallier à ce défaut, une solution consiste à utiliser des techniques de *clustering*. Ces techniques ont été étudiées intensivement dans le contexte des analyses de cluster et ont été appliquées avec succès pour déterminer des partitions d'une collection relativement hétérogène d'éléments. La technique de clustering utilisée par SPEA est assez intuitive. Au début de la procédure, chaque individu constitue son propre groupe, puis on fusionne deux à deux les groupes les plus proches en terme de distance. Cette étape est itérée jusqu'à l'obtention du nombre désiré de groupes. Une fois les groupes identifiés, il ne reste plus qu'à choisir un représentant par groupe. Ce représentant peut être déterminé de plusieurs façons, par exemple en prenant le barycentre du groupe. C'est ce représentant qui sera gardé, les autres éléments étant tout simplement supprimés. Les étapes de la méthode de clustering sont illustrées à la figure II.10 [11].

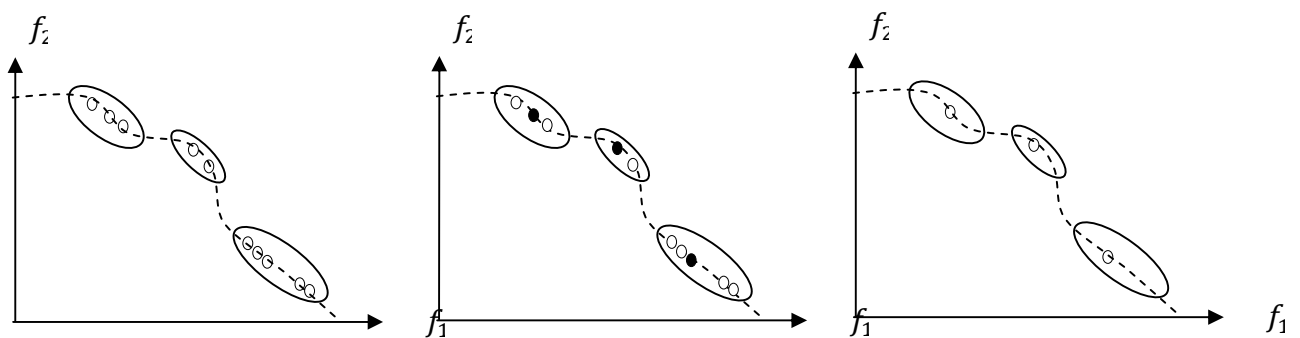


Figure II.10: Illustration du Clustering en dimension 2.

4.3.8. Pareto Archived Evolution Strategy (PAES):

Cette méthode a été développée initialement comme une méthode de recherche locale. Les premiers travaux de Knowles et Corne ont montré que cette méthode simple objectif fournissait des résultats supérieurs aux méthodes de recherche basées sur une population. Par conséquent, les auteurs ont adapté cette méthode aux problèmes multiobjectifs. Les particularités de cette méthode sont les suivantes :

- ◆ Elle n'est pas basée sur une population. Elle n'utilise qu'un seul individu à la fois pour la recherche des solutions.
- ◆ Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales.
- ◆ L'algorithme utilisé est très simple et inspiré d'une stratégie d'évolution ($i+1$).
- ◆ Elle utilise une technique de crowding basé sur un découpage en hypercubes de l'espace des objectifs [18]

4.4. Les méthodes hybrides :

Afin d'améliorer les performances d'un algorithme, on essaye de le combiner avec une autre méthode. Ce principe général appelé hybridation, peut s'appliquer à plusieurs méthodes. Un cas particulier de l'hybridation entre deux méthodes consiste à combiner un algorithme génétique avec une méthode de recherche locale. Dans une telle hybridation on substitue souvent la mutation par une méthode de recherche locale. Dans le cas des problèmes multiobjectifs on peut citer les méthodes hybrides suivantes :

- La méthode MOTS combinant une population et une recherche Tabou,
- La méthode PSA combinant un algorithme génétique et le recuit simulé,
- La méthode M-PAES intégrant un schéma généralisant l'implémentation d'un grand nombre d'algorithmes hybrides pour l'optimisation multiobjectif [16].

5. Conclusion :

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multiobjectif et les travaux de recherche sur les méthodes d'optimisation de problèmes multiobjectifs. Nous avons vu que cette problématique est divisée en trois approches. La première approche appelée L'approche Métaheuristique, qui utilise des méthodes souvent inspirées par des systèmes naturels comme le recuit simulé et les algorithmes évolutionnaires telle que : les algorithmes génétiques pour apporter une réponse à leurs problèmes. La deuxième approche appelée : L'approches non Pareto, cette approche tente de ramener un problème multiobjectif à un problème simple objectif au risque d'enlever toute signification au problème. La troisième approche est : L'approche Pareto, qui elle adopte un point de vue plus global en prenant en compte l'ensemble des critères et en utilisant la notion de dominance au sens de Pareto. Nous avons vu aussi L'approche hybride qui fait l'hybridation entre deux méthodes différentes.

Les Algorithmes Génétiques Multiobjectifs

1. Introduction :

Les méthodes des algorithmes génétiques multiobjectifs connaissent un intérêt croissant depuis le début des années 1990 car elles sont particulièrement bien adaptées au traitement de problèmes multiobjectifs où l'on recherche un ensemble de solutions. Les algorithmes génétiques travaillent sur une population de solution cette caractéristique leur permet de trouver plusieurs solution potentiellement Pareto optimales. Donc l'approche des algorithmes génétiques peut être capable de résoudre le problème de contrôle multiobjectif. Nous présentons dans ce chapitre les concepts principaux d'un algorithme génétique multiobjectif.

2. Les Algorithmes Génétiques (AGs):

Les algorithmes génétiques ont été développés par Holland en 1975. Ils sont basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin en 1859 et la procréation selon les règles de Mendel. Ils ont été efficacement utilisés pour résoudre plusieurs problèmes d'optimisation multicritères. L'algorithme génétique est une technique d'optimisation basée sur les concepts de la sélection naturelle et génétique. L'algorithme commence avec un ensemble de solutions possibles du problème (individus), constituant une population. Les individus sont formés par des variables, qui sont les paramètres à ajuster dans un dispositif. Cette population est conçue aléatoirement à l'intérieur de limites prédéfinies. Certaines solutions de la première population sont utilisées pour former une nouvelle population, à partir d'opérateurs génétiques (croisement, mutation, etc.). Ceci est motivé par l'espoir que la nouvelle population soit meilleure que la précédente. Les solutions qui serviront à former de nouvelles solutions sont sélectionnées aléatoirement d'après leurs mérites (représentés par une fonction objectif spécifique au problème posé, qui devra être minimisée ou maximisée) : meilleur est l'individu, plus grandes seront ses chances de se reproduire c'est-à-dire, plus grande sera sa probabilité d'être sélectionné pour subir les opérateurs génétiques. Ceci est répété jusqu'à ce qu'un critère de convergence soit satisfait (par exemple, le nombre de générations ou le mérite de la meilleure solution) [15].

Comme indiqué auparavant, plusieurs travaux de recherche sont en cours, qui utilisent les AGs, pour les problèmes l'optimisation multicritères. On peut résumer ce qui différencie les AGs des autres approches d'optimisation en quatre points :

- Les AGs utilisent un codage des paramètres et non les paramètres eux-mêmes.
- Les AGs travaillent sur une population de points et non sur un point particulier. Ce qui permet la recherche sur un espace multi dimensionnel.
- Les AGs ne posent aucune condition sur la nature des fonctions à optimiser.
- Les AGs utilisent des règles de transitions probabilistes et non déterministes

2.1. Les opérateurs des algorithmes génétiques :

2.1.1. Population initiale :

L'AG démarre avec une population composée de N individus dans le codage retenu. Le choix des individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace de recherche est totalement inconnue, il est intéressant que la population soit répartie sur tout l'espace de recherche. Si par contre des informations à priori sur le problème sont disponibles, il paraît évident de générer les individus dans un espace particulier afin d'accélérer la convergence. Disposant d'une population initiale souvent non homogène, la diversité de la population doit être entretenue aux cours des générations afin d'explorer le plus largement possible l'espace de recherche. C'est le rôle des opérateurs de croisement et de mutation [18].

2.1.2. Codage des éléments d'une population:

Premièrement, il faut représenter les différents états possibles de la variable dont on cherche la valeur optimale sous forme utilisable pour un AG c'est le codage. Où a chaque paramètre X_i on doit faire correspondre un gène.

Pour les algorithmes génétiques un des facteurs les plus importants est la façon dont sont codées les solutions (ce que l'on a nommé ici les chromosomes), c'est-à-dire les structures de données qui coderont les gènes. Un chromosome regroupe un ensemble d'attributs ordonnés appelés gènes. Un gène prend pour valeur l'une parmi celles possibles : ces valeurs sont les allèles. L'emplacement d'un gène s'appelle locus. Dans le cas d'un codage binaire les allèles possibles sont $\{0, 1\}$. Le codage se base sur deux notions importantes : le génotype et le phénotype [19].

Un génotype : représente l'ensemble des valeurs des gènes d'un chromosome.

Un phénotype : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype. Par exemple, si une solution s'exprime naturellement sous forme d'un vecteur de nombres entiers, le phénotype sera ce vecteur. Le génotype sera ainsi une chaîne de symboles binaires qui code ce vecteur.

Cela permet d'établir une connexion entre la valeur de la variable et les individus de la population, de manière à imiter la transcription génotype-phénotype qui existe dans le monde vivant [18].

Il existe principalement trois types de codage : le codage binaire, le codage réel (appelé aussi codage symbolique) et le codage sous forme d'arbre, alors que le plus utilisé est le codage binaire.

2.1.2.1. Codage binaire :

Ce codage a été le premier à être utilisé dans le domaine des AGs. Il présente plusieurs avantages : alphabet minimum $\{0,1\}$, facilité de mise au point d'opérateurs génétiques et existence de fondements théoriques (théorie sur les schémas).

Néanmoins ce type de codage présente quelques inconvénients :

1. Les performances de l'algorithme sont dégradées devant les problèmes d'optimisation de grande dimension à haute précision numérique. Pour de tels problèmes, les AG basés sur les chaînes binaires ont de faibles performances.
2. La distance de Hamming entre deux nombres voisins (nombre de bits différents) peut être assez grande dans le codage binaire : l'entier 7 correspond à la chaîne 0111 et la chaîne 1000 correspond à l'entier 8. Or la distance de hamming entre ces deux chaînes est de 4, ce qui crée bien souvent une convergence, et non pas l'obtention de la valeur optimale [18].

Cet inconvénient peut être évité en utilisant un codage de Gray par exemple [20].

2.1.3. L'évaluation:

L'évaluation permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population. L'évaluation d'un individu ne dépend pas de celle des autres individus, pour calculer le coût d'un point de l'espace de recherche, on utilise une *fonction d'évaluation* f , cette fonction est souvent une transformation g de la fonction objectif $f(x) = g(\Phi(x))$, appelée aussi la fonction *fitness*, le résultat fourni par cette fonction va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante [21].

L'évaluation de chaque individu consiste à transformer la chaîne binaire 0 et 1 du chromosome en une valeur réelle, appelée *valeur d'adaptation* [11].

2.1.4. La Sélection:

Une opération de sélection est nécessaire pour pouvoir choisir les chromosomes qui garantiront une amélioration de la qualité des solutions. La sélection est un processus qui consiste à choisir parmi tous les individus de la population ceux qui vont participer à la construction d'une nouvelle génération où ce choix est basé essentiellement sur les valeurs d'adaptation de chaque individu. Un membre ayant une grande valeur de la fonction objectif aura plus de chance d'être sélectionné pour participer à la reproduction de la prochaine génération. Nous citons quelques méthodes utilisées pour la sélection des individus, qui vont se reproduire :

- Sélection par roulette (RWS : Roulette Wheel Selection).
- Sélection par (SUS : Stochastic Universal Sampling).
- Sélection par Rang.
- Sélection par Tournoi [15].

2.1.4.1. Sélection par roulette (RWS : Roulette Wheel Selection):

La sélection par la roulette (RWS) consiste à affecter une probabilité de sélection à chaque individu. Cette probabilité est proportionnelle à la valeur d'évaluation, soient :

F_p : l'évaluation de l'individu p .

P_{SP} : Probabilité de sélection de l'individu p .

Pour maximiser la fonction d'évaluation, la probabilité de sélection d'un individu est la suivante :

$$P_{SP} = \frac{F_p}{\sum_P F_p} \quad (\text{III-1})$$

Pour minimiser la fonction d'évaluation, la probabilité de sélection d'un individu est la suivante :

$$P_{SP} = \frac{\frac{1}{F_p}}{\sum_P \frac{1}{F_p}} \quad (\text{III-2})$$

Se voyant affectés ces probabilités, tous les individus de la population peuvent participer à la reproduction. En effet, une roulette divisée en un nombre de secteur égal au nombre des chromosomes est lancée. L'aire de chaque secteur est proportionnelle à l'évaluation de

l'individu correspondant, ainsi un indicateur sur cette roulette indique l'individu sélectionné après l'arrêt de roulette.

L'inconvénient de cette méthode réside dans la présence probable d'un Super individu. Le Super individu est un individu dont la probabilité de sélection est très supérieure à celle des autres individus. Il risque alors d'être toujours choisi ce qui peut limiter le champ d'exploration [15].

On préfère souvent des méthodes qui n'autorisent en aucun cas l'apparition de super-individu. Par exemple, la sélection par tournoi.

2.1.4.2. Sélection par Tournoi :

La sélection par tournoi consiste à comparer une paire d'individus choisis au hasard, où le meilleur de ces deux est gagnant et sera déclaré sur le tournoi, avec une population de N chromosomes, on forme N paires de chromosomes pour la reproduction.

L'avantage de cette méthode est d'éviter d'avoir le super individu qu'on peut rencontrer dans la méthode de sélection précédente. Par contre, le meilleur individu peut ne pas être sélectionné par cette méthode, et ainsi le champ d'exploration est réduit [15].

2.1.4.3. Sélection par Stochastic Universal Sampling (SUS) :

Cette méthode de sélection, a le même principe de sélection par roulette, mais avec des indicateurs multiples et équidistants.

Ces méthodes proportionnées de choix, ont un problème de graduation (scaling problem), ce problème peut être également évité en employant la méthode *de sélection par rang* [7].

2.1.4.4. Sélection par rang :

La sélection par rang d'un chromosome est la même que par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation. La sélection par rang trie d'abord la population par fitness. Ensuite, chaque chromosome se voit associé un rang en fonction de sa position. Ainsi le plus mauvais chromosome aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur chromosome qui aura le rang N (pour une population de N chromosomes) [21].

2.1.5. Le Croisement :

L'opérateur de croisement est un opérateur génétique qui permet à deux chromosomes parents de donner deux chromosomes enfants. Cet opérateur se produit selon une probabilité

P_c fixée par l'utilisateur selon le problème à optimiser. A travers cette reproduction les chromosomes enfants héritent parties de patrimoine génétique de leurs parents.

L'opérateur de croisement permet de créer de nouvelles séquences de gènes pour les chromosomes enfants à partir d'une base de configuration des séquences héritées des chromosomes parents.

Il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que le croisement à un point, le croisement multipoints [15].

A. Croisement en un point :

Pour chaque couple, on choisit au hasard un point de croisement (figure III.1). Le croisement s'effectue directement au niveau binaire, et non au niveau des gènes. Un croisement peut être coupé au milieu d'un gène.

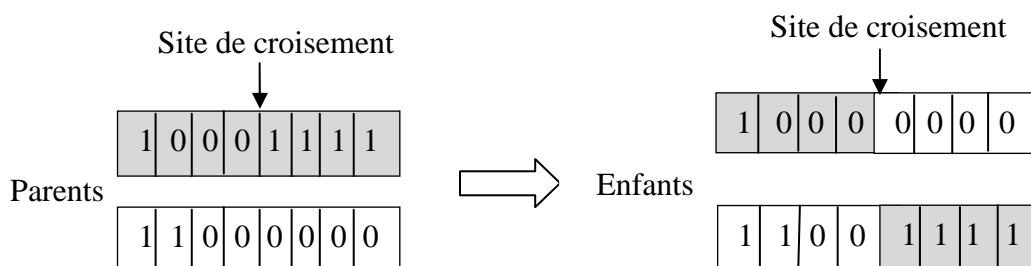


Figure III.1. Représentation schématique du croisement en un point.

B. Croisement en deux points :

On choisi au hasard deux points de croisements successifs. Cet opérateur est généralement considéré comme plus efficace que le précédent [22].

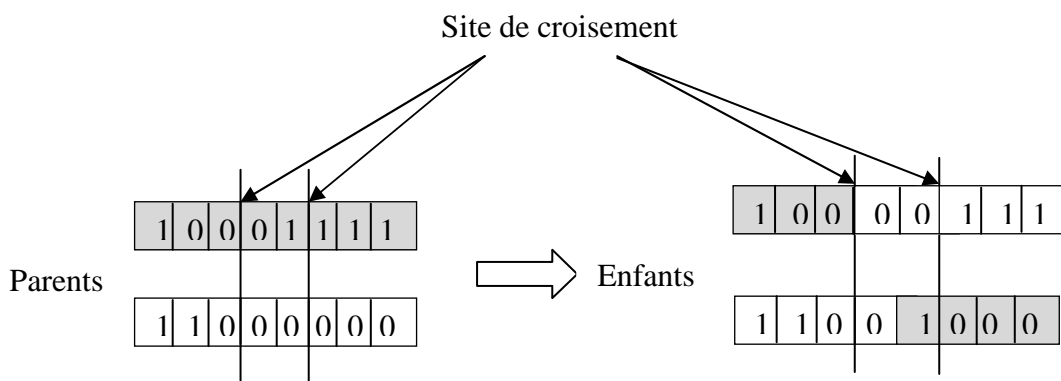


Figure III.2. Représentation schématique du croisement en deux points.

C. Croisement uniforme :

Il opère à l'aide d'un masque qui représente les tirages aléatoires, pour décider de la transmission de la valeur de l'allèle à l'un ou l'autre des descendants. Si, à la même position que l'allèle, la valeur du masque est égale à 1, l'allèle du parent 1 passe à celui de l'enfant 1 et l'allèle du parent 2 passe à l'enfant 2. Sinon, c'est l'inverse qui se produit (Tableau III.1) [23].

Parent 1	1	0	0	0	1	1	1	1	1	0
Parent 2	1	1	1	1	0	0	0	0	0	1
Masque	1	0	0	1	1	1	0	0	1	0
Enfant 1	1	1	1	0	1	1	0	0	1	1
Enfant 2	1	0	0	1	0	0	1	1	0	0

Tableau III.1. Exemple de croisement uniforme.

2.1.6. Mutation :

La mutation est un changement aléatoire selon une certaine règle probabiliste qui doit faire sur les génotypes, avec une faible probabilité P_m (fixée par l'utilisateur) de la valeur d'un ou plusieurs allèles d'un chromosome. En général, la mutation ne permet pas l'obtention de meilleures solutions, mais elle permet de garder une diversité dans l'évolution des individus et d'éviter les optimums locaux, et se protège contre une perte irrécouvrable dans les caractéristiques des individus.

La mutation classique consiste à transformer dans un chromosome binaire un 1 en un 0 ou le contraire. (Figure III.3) [15].

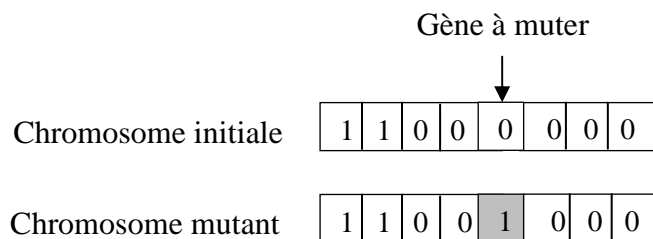


Figure III.3. Représentation schématique de la mutation.

La probabilité P_m de mutation selon Fonseca est calculée comme suivant :

$$P_m = 1 - \sigma^{-\frac{1}{l}} \tag{III-3}$$

l : est le longueur du chromosome.

σ : est la pression sélective, où sa valeur recommandée est: 1,8 [7].

Cependant, il existe plusieurs opérateurs de mutation :

- **Transposition de deux allèles consécutifs** : Cette mutation consiste à choisir deux allèles consécutifs au hasard et d'échanger leurs valeurs respectives.
- **Transposition de deux allèles quelconques** : Cette mutation consiste à choisir deux allèles au hasard et d'échanger leurs valeurs respectives
- **Inversion d'allèles** : Cette mutation consiste à choisir deux allèles au hasard et d'inverser l'ordre des allèles contenus dans la zone sélectionnée [15].

3. Fonctionnement d'un Algorithme Génétique :

Un Algorithme Génétique (AG) est un processus itératif de recherche exploratoire dont les étapes sont les suivantes :

1. Initialisation de la population initiale P_0 (population de génération 0).
2. Évaluation des chromosomes de la population P_i
3. Sélection des meilleurs chromosomes de P_i qui vont constituer les « procréateurs » de la génération P_{i+1} .
4. Croisement deux-à-deux des chromosomes sélectionnés pour générer de nouveaux individus, les « descendants ».
5. Mutation opérée aléatoirement sur un chromosome également choisi au hasard pour apporter de la diversité dans le patrimoine génétique de la population.
6. Remplacement et génération de la nouvelle population P_{i+1} .
7. Répéter le processus à partir de 2 tant que le critère d'arrêt n'est pas satisfait (i.e. génération N terminale non atteinte, solution optimale non trouvée...) [24].

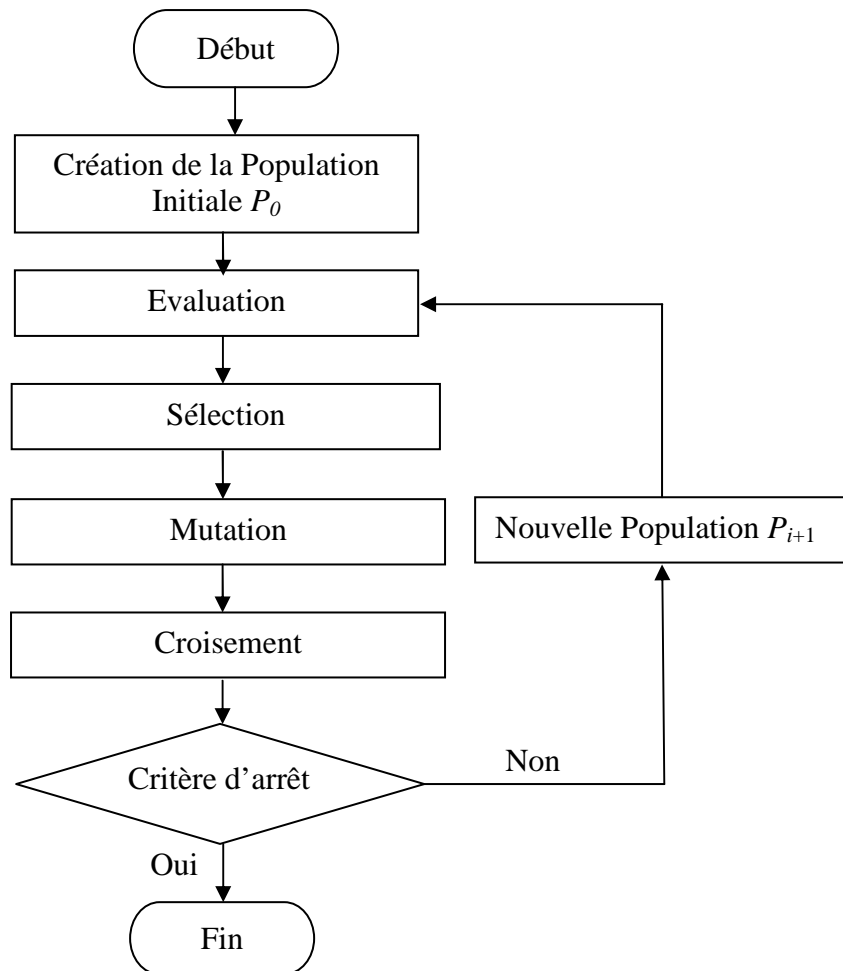


Figure III.4. Schéma de principe d'un Algorithme Génétique.

4. Les paramètres de l'AG :

Les opérateurs de l'AG sont guidés par un certain nombre de paramètres généralement fixés à l'avance et dont dépend très fortement la « bonne » convergence de l'algorithme.

Présentons brièvement les principaux paramètres de l'AG.

- La taille de la population et la longueur du codage de chaque individu. Il est conseillé de prendre comme taille de la population la valeur correspondant à la longueur du codage des individus. En effet, si cette taille est très grande, l'évaluation de tous les individus de la population peut s'avérer trop longue. Par contre, si elle est très petite, l'algorithme peut converger trop rapidement [25].
- La probabilité de croisement P_c dépend de la forme de la fonction d'évaluation. Son choix est généralement expérimental et sa valeur est très souvent prise entre 0.5 et 0.9. Plus elle

est élevée, plus la population subit des changements importants. Ainsi, la convergence est très rapide si le taux de croisement est proche de 1.

- La probabilité de mutation P_m : le taux de mutation est généralement faible, le risque d'un taux élevé étant de modifier les meilleurs individus et ainsi, de s'éloigner de l'optimalité [26].

5. Algorithmes Génétiques Multi-Objectifs (MOGA) :

En 1993 Fonseca et Fleming ont proposé une méthode dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent. Ensuite, ils ont utilisé une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang.

Soit un individu x_i à la génération t , dominé par $p_i(t)$ individus. Le rang de cet individu est :

$$\text{Rang}(x_i, t) = 1 + p_i(t) \quad (\text{III-4})$$

Tous les individus non dominés sont de rang 1.

Dans l'exemple ci-dessous, les points 1,3 et 5 ne sont dominés par aucun autre. Alors que le point 2 est dominé par le point 1, et que le point 4 est dominé par les points 3 et 5.

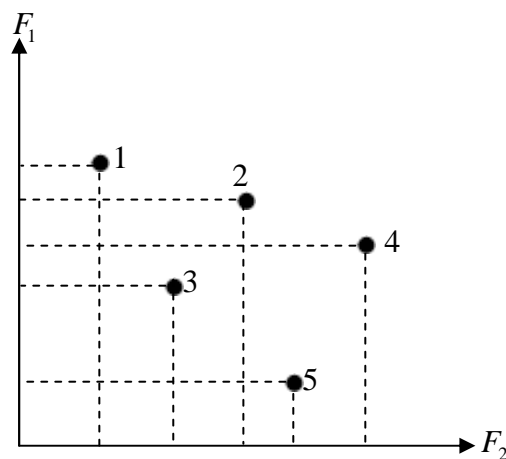


Figure III.5. Exemple de dominance.

Dans cet exemple, le tableau de dominance est le suivant :

Point	Dominé par	Rang
1	aucun	1
2	1	2
3	aucun	1
4	3 et 5	3
5	aucun	1

Tableau III.2 : Tableau de dominance

Fonseca et Fleming (1993) calculent la fitness de chaque individu de la façon suivante :

- a) Calcul du rang de chaque individu.
- b) Affectation de la fitness de chaque individu par application d'une fonction de changement d'échelle (On parle aussi de fonction de scaling.) sur la valeur de son rang. Cette fonction est en général linéaire. Suivant le problème, d'autres types de fonction pourront être envisagés afin d'augmenter ou de diminuer l'importance des meilleurs rangs ou d'atténuer la largeur de l'espace entre les individus de plus fort rang et de plus bas rang.

Discussion :

L'utilisation de la sélection par rang a tendance à répartir la population autour d'un même optimum. Or cela n'est pas satisfaisant pour un décideur car cette méthode ne lui proposera qu'une seule solution. Pour éviter cette dérive, les auteurs utilisent une fonction de sharing. Ils espèrent ainsi répartir la population sur l'ensemble de la frontière de Pareto.

Le sharing utilisé dans cette méthode agit sur l'espace des objectifs. Cela suppose que deux actions qui ont le même résultat dans l'espace des objectifs ne pourront pas être présentes dans la population. Cette méthode obtient des solutions de bonne qualité et son implémentation est facile. Mais les performances sont dépendantes de la valeur du paramètre δ_{share} utilisé dans le sharing [17].

L'adaptation des algorithmes génétiques à l'optimisation multi-objectifs se fait principalement au niveau de l'étape d'évaluation de l'efficacité d'une solution qui est alors divisée en deux étapes :

- évaluation de l'efficacité de la solution en termes de convergence.
- évaluation de l'efficacité de la solution en termes de diversification.

Le schéma de l'algorithme génétique avec ajout de ses phases est représenté dans la (figure III.6). L'étape 2a correspond au calcul de l'efficacité par rapport à la convergence et l'étape 2b correspond au calcul de la diversité de la solution. Les principaux mécanismes employés pour ces tâches sont présentés dans les sous sections qui suivent. Il est à noter que selon les mécanismes employés il est possible d'obtenir deux efficacités différentes pour une même solution un pour la convergence et l'autre pour la déversification. il est alors nécessaire de prendre en compte ce fait dans la phase de sélection lorsqu'il s'agit de comparer deux solutions .un autre mécanismes, l'élitisme est présenté dans cette partie.

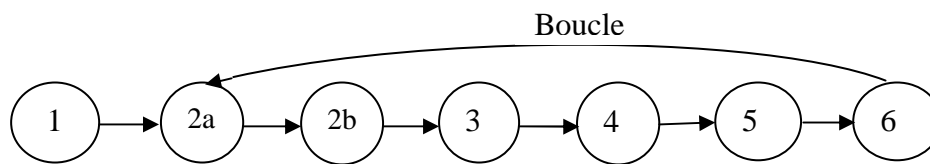


Figure III.6 Etapes d'un algorithme génétique multiobjectif

Les étapes de l'algorithme génétique multi-objectif sont les mêmes que celles de la (figure III.6). L'étape scindée en deux sous étapes la première (2a) correspond à l'évaluation de la qualité de la solution en termes de la convergence et la seconde (2b) correspond à l'évaluation de la qualité de la solution en termes de la diversité

1. initialisation de la population.
3. sélection.
4. croisement.
5. mutation.
6. remplacement [27].

5.1. Sélection multicritère:

Le problème posé consiste à sélectionner les individus mais en tenant compte de plusieurs critères au lieu d'un seul. On considère que la technique de base consistant à réunir tous les critères sous la forme d'une somme pondérée ne répond pas correctement à ce problème. Il faut donc adapter les opérateurs de sélection. Une première méthode consiste à sélectionner tour à tour les individus sur chacun des critères. Mais il est encore plus efficace d'utiliser la notion de Pareto-optimalité. Cette notion de pareto optimalité permet d'établir une relation de dominance entre les individus portant sur plusieurs critères. où seuls deux individus se confrontent, l'individu finalement sélectionné sera celui qui domine l'autre au sens de pareto. L'ordre induit par l'optimalité selon Pareto étant partiel, il arrive qu'il soit impossible de

décider si un individu domine l'autre. Dans ce cas, il faut utiliser des techniques plus avancées. On peut choisir par exemple de ne sélectionner que les individus non dominés. Cependant, s'il est nécessaire de fixer un ordre sur les individus, alors plusieurs approches sont possibles. Par exemple, on peut utiliser une dominance de rang pour laquelle on prend en compte le nombre d'individus dominant un individu. D'autres approches utilisent la profondeur de dominance, dans ce cas la population est divisée en plusieurs fronts, chacun correspondant à l'ensemble des individus n'étant dominé par aucun autre. Enfin une autre technique consiste à utiliser le nombre d'individus dominés par un certain individu. Les individus non dominés peuvent alors être ordonnés entre eux [28].

5.2. Assignment de l'efficacité :

L'idée de calculer l'efficacité d'un individu en utilisant la dominance Pareto a été introduite par Goldberg. En général on parle alors du rang de l'individu. Plusieurs approches ont été proposées pour assigner un rang aux différentes solutions. La première technique consiste à attribuer à chaque solution un rang de dominance qui correspond au nombre de solutions dans la population qui domine l'individu. Le rang d'un individu est alors égal au nombre d'individus de la population qui le domine plus un. Cette technique est notamment employée par l'algorithme de Fonseca et Fleming.

Une autre possibilité est d'attribuer une profondeur de dominance. Il s'agit de diviser la population en plusieurs fronts et le rang d'une solution correspond à la profondeur du front auquel elle appartient. Le rang d'un individu est calculé par l'algorithme. Les individus qui ne sont dominés que par des solutions de E_1 reçoivent le rang 2 et forme par l'ensemble E_2 . De manière générale un individu ne reçoit le rang k que s'il est dominé par des individus de la population appartenant à l'ensemble $E_1 \cup E_2 \cup \dots \cup E_{k-1}$. Cette est notamment utilisée dans l'algorithme NSGA II.

Une dernière possibilité, le compte de dominance consiste à compter le nombre de solutions qu'un individu domine cette mesure ne donne pas tout de suite une information sur l'efficacité de la solution et doit être utilisée, à l'instar de ce qui est fait dans SPEA et SPAE 2, en conjonction avec une autre technique.

Les différentes techniques sont illustrées dans la figure III.7, leur intérêt par rapport à des méthodes scalaires ou autres est qu'elles permettent d'évaluer l'efficacité d'une solution par rapport à tout l'ensemble de la population et qu'elles ne sont pas affectées par la forme ou la continuité de la frontière Pareto [27].

5.3. Préservation de la diversité :

Les méthodes présentées précédemment tendent à favoriser la convergence vers le front Pareto optimale en avantageant les individus qui sont peu ou pas dominés. Cependant ces méthodes ne sont pas capables de garantir que l'approximation obtenue sera de bonne qualité

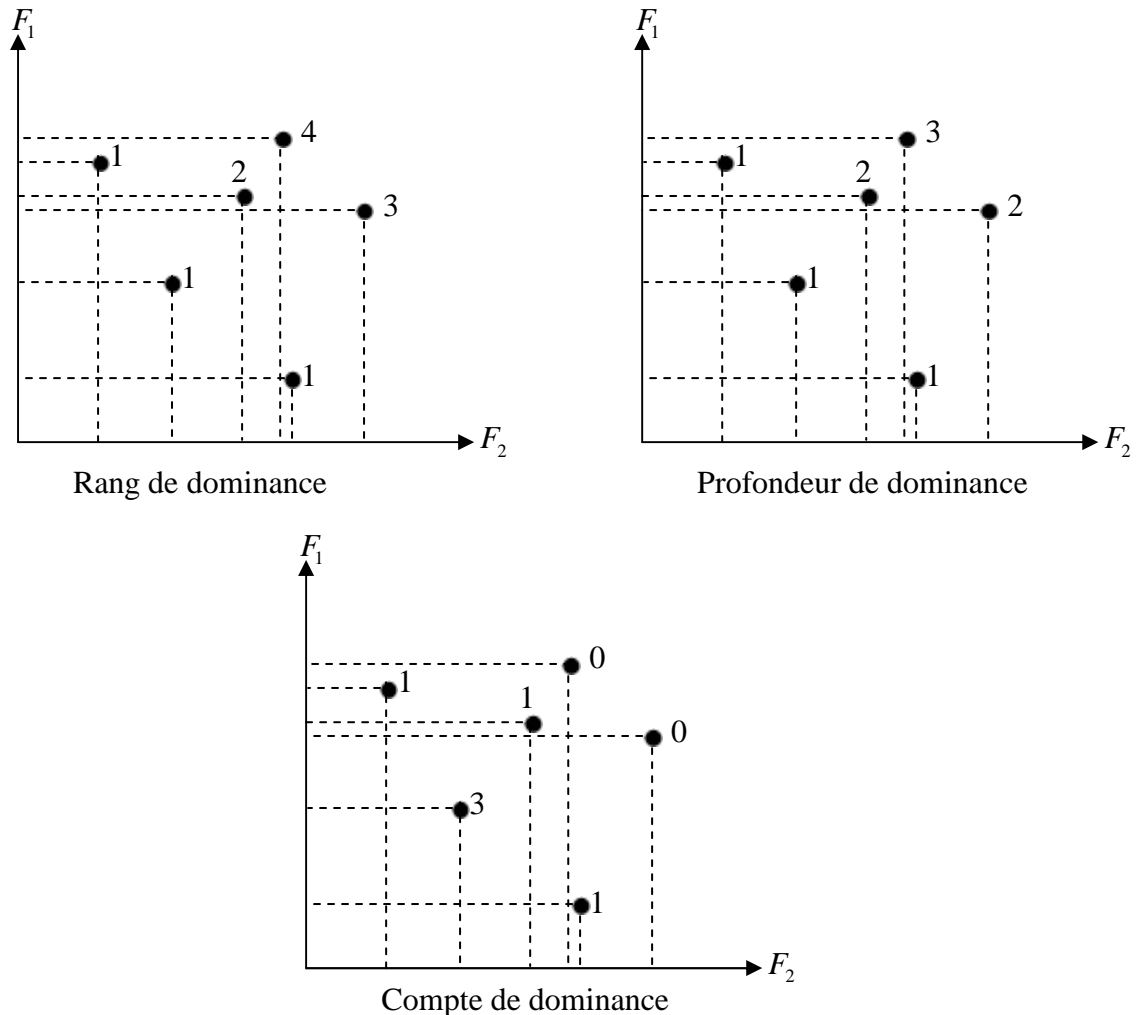


Figure III.7. Assignment de l'efficacité d'un point de vue de dominance

En termes de diversification. Ainsi les algorithmes génétique incorporent des méthodes qui évaluent la diversité d'une solution par rapport aux autres solutions de la population les différentes approches prennent en compte une notion de voisinage qu'il est possible de définir de trois manières différentes comme suggéré par Zitzler et al, les trois catégories sont basées sur celles utilisées en estimation de densité statistique.

Les méthodes à noyau définissent le voisinage d'une solution i par rapport à une fonction K qui prend en paramètre la distance d_{ij} entre i et les autres points j sont calculées et la somme des $K(d_{ij})$ effectuée cette somme représente la densité autour de l'individu i le fitness sharing

utilisé notamment dans MOGA, NSGA et NPGA est certainement la technique la plus populaire dans les algorithmes évolutionnaire.

Les techniques de voisin le plus proche calculent la distance entre un point donné et son $k^{\text{ème}}$ voisin le plus proche pour estimer la densité dans son voisinage. Cette technique a notamment été utilisée dans SPEA 2

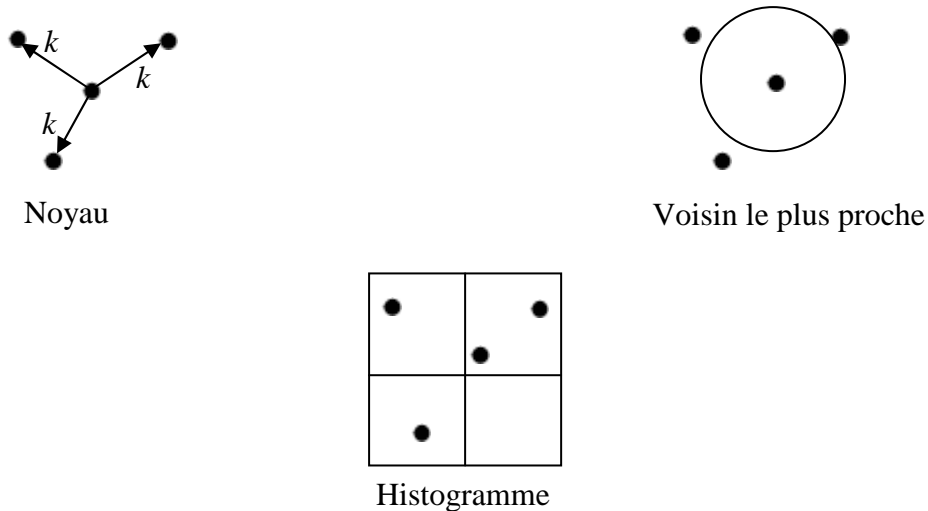


Figure III.8 Illustration des techniques de la préservation de la diversité.

Une troisième catégorie utilise les histogrammes, dans cette technique, l'espace est divisé en voisinage par une hypergrille. La densité autour d'une solution est estimée par le nombre de solutions se trouvant dans la même case de la grille. Cette technique a été employée dans PAES.

Les différentes approches sont illustrées dans la figure III.8, pour toutes ces méthodes il est nécessaire de définir une métrique qui peut être calculée sur le génotype et/ou le phénotype dans l'espace décisionnel et/ou dans l'espace des objectifs [27].

5.4. Elitisme :

La stratégie élitiste consiste à conserver le meilleur individu à chaque génération. Ainsi l'élitisme empêche l'individu le plus performant de disparaître au cours de la sélection ou que ses bonnes combinaisons soient affectées par les opérateurs de croisement et de mutation. Après chaque évaluation de la performance des individus à une génération t donnée, le meilleur individu de la génération précédente ($t - 1$) est réintroduit dans la population si aucun des individus de la génération t n'est meilleur que lui. Par cette approche, la performance du meilleur individu de la population courante est monotone de génération en génération. Il apparaît que l'élitisme améliore considérablement les performances de l'algorithme génétique

pour certaines classes de problèmes, mais peut les dégrader pour d'autres classes, en augmentant le taux de convergences prématurées.

L'algorithme génétique multiobjectif (MOGA) a été appliqué avec succès aux problèmes de contrôle des systèmes, par exemple : contrôleur d'un système MIMO et l'optimisation des paramètres, contrôle des systèmes par la commande H_∞ .

MOGA est un algorithme évolutionnaire emploie les opérateurs suivantes :

- le rang de Pareto optimal.
- Fitness Sharing.
- restriction d'accouplement (Mating Restriction).
- sélection, croisement et mutation.

La philosophie de MOGA est de développer une population des solutions Pareto optimales ou proches de Pareto optimales dans tout le processus d'optimisation [7].

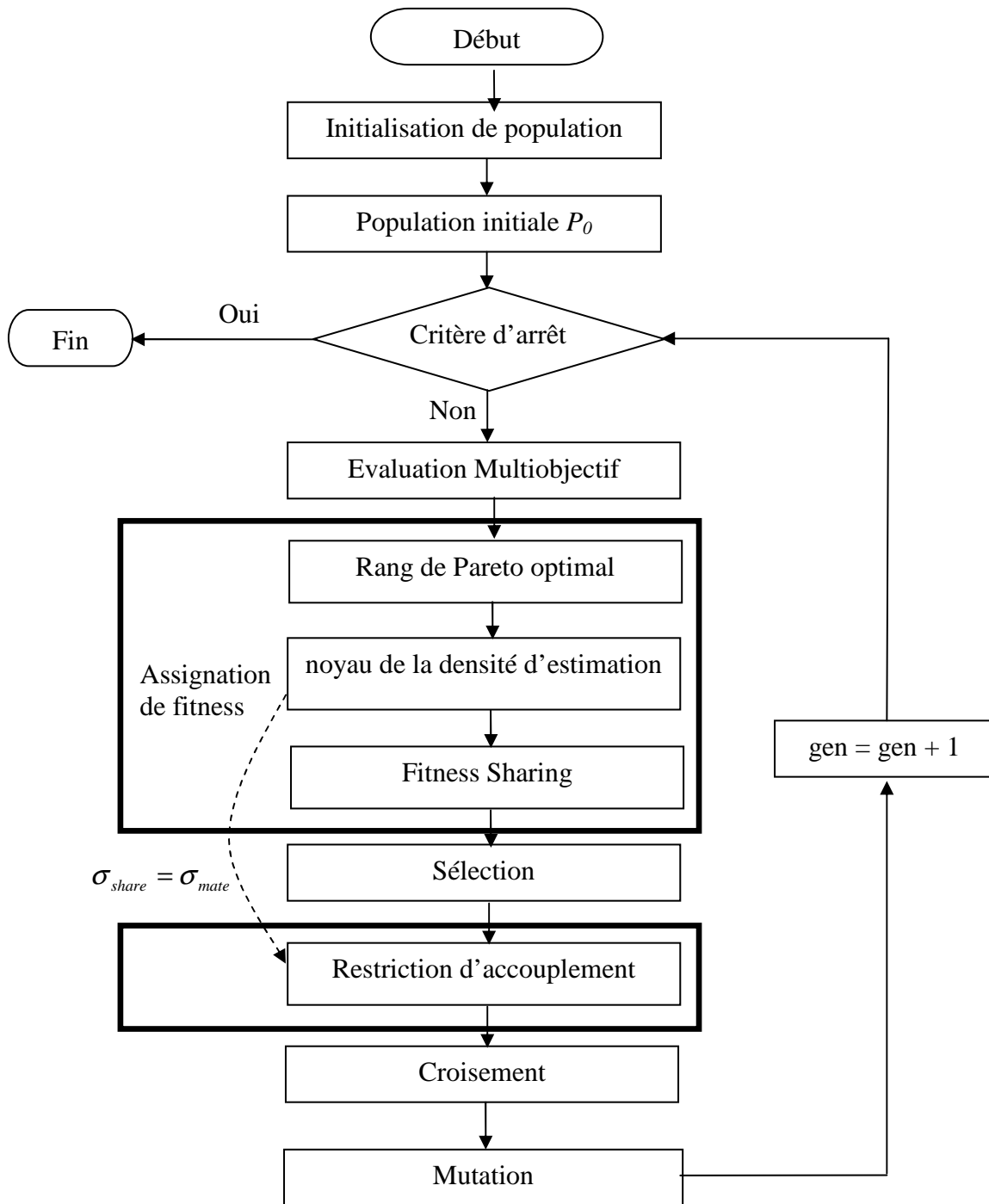


Figure III.9 Algorithme Génétique Multiobjectifs

Dans MOGA, la population initiale est choisie aléatoirement dans une marge définie et alors décodée (en cas de chromosome non réel). Et puis évalué chacun individu de population initiale par un ensemble de valeurs de fonction objectif. Alors la séquence des opérateurs génétiques est appliquée, ayant un résultat pour la génération suivante, Noté que la tâche

d'assigner fitness est un processus plus raffiné et importante (voir la région ombreuse sur le Figure III.9).

5.5. Rang de Pareto optimal

En absence d'information concernant l'importance des objectifs, la dominance de Pareto est la seule méthode de déterminer l'évaluation de performance. Donc tous les individus Non dominés sont considéré les meilleurs individus et ainsi sont assignés la même fitness, par exemple zéro. Cependant, la détermination d'une valeur de fitness pour les individus dominés est une question plus subjective.

Les individus sont rangés, en un vecteur but et on fait la préférence (but, priorité). La considération du but et de la priorité sélective exclut des objectifs selon leurs priorités et s'ils atteignent aux leurs buts, l'information de priorité et de but peuvent être souvent extraite directement à partir de la description de problème. Les priorités sont des valeurs de nombre entier qui déterminent auxquels ordre d'objectifs doivent être optimisés, selon leur importance. Fréquemment dans le contrôle des systèmes, la stabilité en boucle fermé a la priorité la plus élevée et il est d'abord minimiser les valeurs de buts indiquent le niveau désiré de performance dans chaque dimension d'objectif. Le vecteur de but trace la région de l'échange où MOGA concentre son effort de calcul. Une fois le rang est trié, cet opérateur génétique assignera la fitness aux individus par l'interpolation du meilleur au plus mauvais, selon une règle exponentielle. Alors une valeur simple de fitness est dérivée pour chaque groupe d'individus avec le même coût, en utilisant la moyenne.

5.6. Fitness Sharing et noyau de la densité d'estimation

Bien que la population puisse rechercher potentiellement à beaucoup d'optimums locaux, une population finie tendra à évoluer vers une petite région de l'espace de recherche même s'ils existent d'autres optimums équivalents. Ce phénomène est connu par la migration génétique. Un remède pour ce problème a proposé par Fonseca et Fleming (1995) avec le *Fitness Sharing* C'est une technique comportant l'estimation de la densité de population, aux points définis par chacun individu. Une taille de niche σ_{share} appropriée a été développée par Fonseca et Fleming en employant une méthode semblable par l'estimation de densité connue sous le nom de noyau d'Epanechnikov, est employée pour pénaliser des individus selon la proximité d'autres individus [7].

5.7. Technique de niche

Cette technique permet l'exploration de régions distinctes qui constituent des optimums locaux. En pratique, la détection de solutions différentes donne à l'ingénieur la possibilité d'un choix final pas seulement à partir des objectifs prédéfinis, mais aussi par exemple à partir de la facilité de construction de l'une ou de l'autre solution. Nous proposons de calculer des indices de niche, ou de ressemblance, dans les domaines des objectifs (N_{obj}) et des paramètres (N_{par}), de façon à prendre en compte les deux domaines. Ces indices sont les distances entre individus pris dans l'ordre des valeurs de chaque objectif. L'opérateur génétique de sélection ira travailler avec ces indices, et pas avec les évaluations du problème. Le but est de détecter des niches distinctes (dans l'espace de paramètres), c'est-à-dire de détecter des optimums locaux et/ou globaux dans un problème multimodal.

Le processus est constitué de deux étapes. Premièrement, pour chaque objectif k , on met la population en ordre croissant (ou décroissant) selon l'objectif analysé (il faut sauvegarder un vecteur qui indique l'ordre initial) et on calcule les distances entre les individus dans les deux espaces, N_{objk} et N_{park} , par rapport à l'ordre établi. La deuxième étape consiste à joindre des deux indices de ressemblance par une fonction de transfert [4].

5.8. La restriction d'accouplement

Cette issue de restriction est de garder la diversité sur le long de la surface du compromis, Ceci évite une combinaison arbitraire des paires en dehors d'une distance donnée de σ_{mate} cela qui pourrait conduire à la formation d'un grand nombre de génération incapable. La distance est généralement $\sigma_{mate} = \sigma_{share}$. En outre, la diversité de population est encouragée en appliquant un opérateur de mutation à un nombre restreint d'individus existants.

L'approximation de Pareto optimal est particulièrement difficile parce qu'il est inconnu. La manière traditionnelle est d'expérimenter et décider en employant la connaissance a priori ou en rivalisant avec les résultats d'autres méthodes si le front de Pareto disponible s'améliore vers Pareto optimal. Généralement doit être prise l'attention en employant des algorithmes évolutionnaires. Même si un bon choix des paramètres est trouvé pour une application particulière ; cet ensemble sera sous-optimal pour beaucoup d'autres problèmes. La réalisation du choix faux des paramètres peut produire excessivement des mauvaises résultats [7]

6. Conclusion

Les algorithmes génétiques (AG's) semblent être une solution intéressante pour résoudre le problème d'optimisation multiobjectifs. Ce chapitre montre les principes et l'intérêt des AG's à la recherche du Pareto optimale pour un problème d'optimisation multiobjectifs.

Contrôle multi-objectifs H_2/H_∞ en utilisant l'approche LMI

1. Introduction :

Depuis la fin des années 80, les théories de la synthèse H_2 et de la synthèse H_∞ sont bien établies. Ce sont principalement les applications de ce type de synthèse dans l'espace d'état à des cas pratiques de plus en plus complexes qui se sont développées ces dernières années. La confrontation de ces théories très élégantes mais en partie fermées a conduit à faire deux constats complémentaires. D'une part, il est bien connu depuis les travaux présentés, que le régulateur LQG peut présenter des marges de robustesse vis-à-vis d'incertitudes paramétriques, dramatiquement et arbitrairement faibles. D'autre part, les nombreuses applications récentes des compensateurs H_∞ à des problèmes pratiques ont également montré qu'il est relativement difficile d'obtenir avec ce type de correcteurs un niveau satisfaisant de performances mesurées par exemple par la norme H_2 . Cela provient en partie du fait que la synthèse H_∞ est essentiellement fondée sur l'analyse de la performance dans le pire des cas alors que la norme H_2 reflète une performance moyenne. Face à ces deux constats, l'idée d'associer ces deux types de performance est donc apparue relativement naturelle d'autant que les approches H_2 et H_∞ partagent le même formalisme fondé sur le problème standard.

Pour cette raison, nous avons choisi de focaliser notre attention sur le problème mixte H_2/H_∞ qui est également un problème très largement ouvert.

2. Théorie de la synthèse H_∞ :

La synthèse H_∞ a été initiée par Zames au début des années 80 et développée, en particulier par Doyle, Glover, Khargonekar et Francis. Elle est devenue ces dernières années une des méthodes phares de la « commande robuste ». Le problème H_∞ consiste à concevoir une commande assurant la stabilité asymptotique du système bouclé avec un certain taux d'atténuation de l'influence des entrées exogènes sur les sorties du système. Ces entrées sont constituées des signaux de consignes et éventuellement des perturbations liées à

l'environnement du système. La synthèse H_∞ fournit une commande robuste vis-à-vis de ces perturbations et les incertitudes.

Définition (Norme H_∞) : on appelle norme H_∞ du transfert $T(s)$ entre w_1 et z_1 le réel positif défini par :

$$\|T(j\omega)\|_\infty = \sup_{w_1 \neq 0} \left(\frac{\|z_1\|_2}{\|w_1\|_2} \right) \quad (\text{IV-1})$$

On peut écrire aussi:

$$\|T(j\omega)\|_\infty = \sup \bar{\sigma}(T(j\omega)) \quad (\text{IV-2})$$

Où $\bar{\sigma}$ représente la valeur singulière maximale [29].

La norme H_∞ représente la valeur maximale du rapport entre l'énergie du signal de sortie et l'énergie du signal d'entrée. Il est clair que la minimisation de cette norme est nécessaire pour optimiser le rejet de perturbations et le suivi de consignes.

3. Problème H_∞ standard :

Sous sa forme la plus simple, le problème H_∞ est un problème de réjection de perturbation. Il consiste à minimiser l'effet d'une perturbation w sur le comportement du système. Le signal w est supposé d'énergie finie et sa taille est mesurée en norme l_2 . Son effet sur le système est mesuré par la norme l_2 d'un vecteur (coût). Enfin, on peut agir sur le système par une commande u et on dispose d'une observation y . Il s'agit donc de synthétiser une loi de commande $u = K(s).y$ qui minimise l'impact de w sur z . On mesurera cet impact par le rapport

$\frac{\|z\|_2}{\|w\|_2}$. La stabilité interne du système bouclé devra bien sûr être assurée.

Ce problème standard est représenté schématiquement par la figure IV-1.

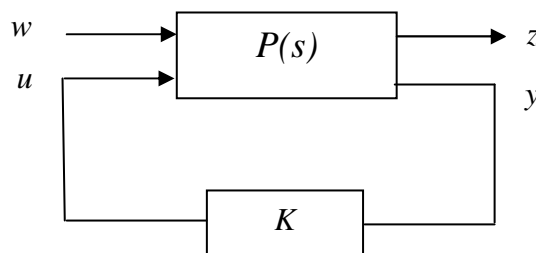


Figure. IV-1. Problème H_∞ standard.

Dans cette figure $P(s)$ représente le modèle augmenté, et $K(s)$ le correcteur. Le système $P(s)$ est formé des quatre transferts multivariables existant entre les entrées u (commande du système) et w (entrées exogènes: consignes, perturbations,...) et les sorties y (mesures) et z (sorties régulées, ou grandeurs à surveiller) :

$$\begin{bmatrix} Z(s) \\ Y(s) \end{bmatrix} = P(s) \cdot \begin{bmatrix} W(s) \\ U(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} W(s) \\ U(s) \end{bmatrix} \quad (\text{IV-3})$$

Cette représentation permet de calculer, avec les outils appropriés, le correcteur K de manière à optimiser, selon un critère donné, la fonction de transfert $T_{zw}(s)$ entre les entrées exogènes $w(t)$ et les sorties régulées $z(t)$. Par ailleurs, pour une loi de commande $u = K.y$ connue, $T_{zw}(s)$ peut se calculer de la façon suivante :

$$T_{zw}(s) = F_l(P, K) = P_{11}(s) + P_{12}(s) \cdot K(s)(sI - P_{12}(s)K(s))^{-1}P_{21}(s) \quad (\text{IV-4})$$

Où : $K = D_K + C_K(sI - A_K)^{-1}B_K$

Une réalisation de $F_l(P, K)$ est donnée par:

$$F_l(P, K) = D_{BF} + C_{BF}(sI - A_{BF})^{-1}B_{BF} \quad (\text{IV-5})$$

La stabilité interne de la boucle fermée est équivalente à la stabilité de A_{BF} , c'est-à-dire à $\Re(\lambda_i(A_{BF})) < 0$. Cette expression, généralement notée $F_l(P, K)$, est appelée Transformation Fractionnaire Linéaire inférieure (LFT). Rappelons en outre, que $P_{22}(s) = G(s)$, où $G(s)$ est la fonction de transfert du système.

En observant que le rapport $\frac{\|z\|_2}{\|w\|_2}$ est dans le pire des cas donné par:

$$\sup_{w \neq 0} \frac{\|z\|_2}{\|w\|_2} = \|F_l(P, K)\|_\infty \quad (\text{IV-6})$$

Le problème décrit ci-dessus peut se formuler mathématiquement comme suit:

3.1. Problème H_∞ Optimal: minimiser $\|F_l(P, K)\|_\infty$ sur l'ensemble des compensateurs $K(s)$ qui stabilisent le système de manière interne.

Le minimum est noté γ_{opt} appelé gain (ou atténuation) "H ∞ optimal". Le problème sous-optimal associé joue également un rôle important:

3.2. Problème H_∞ Sous-Optimal: étant donné $\gamma > 0$, trouver un compensateur $K(s)$ qui stabilise le système de manière interne et assure que : $\|F_l(P, K)\|_\infty < \gamma$ [19]

Cette performance garantit que le gain l_2 du transfert $w_1 \rightarrow z_1$ est inférieur à γ d'une part d'autre part que la stabilité asymptotique est préservée quelque soit une perturbation bornée, et en fin que la valeur singulière maximale de la matrice de transfert pour le couple de sorties/entrées (z_1/w_1) est inférieure à γ pour toutes les fréquences [30].

4. Résolution du Problème H_∞ :

Une réalisation minimale du système $P(s)$, est associée à la description interne suivante pour la représentation d'état (x étant le vecteur d'état et $D_{22} = 0$) :

$$P(s) : \begin{cases} \dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) = C_2 x(t) + D_{21} w(t) \end{cases} \quad (\text{IV-7})$$

Soit $K(s)$ un contrôleur dynamique LTI défini par :

$$K(s) : \begin{cases} \dot{x}_K(t) = A_K x_K(t) + B_K y(t) \\ u(t) = C_K x_K(t) + D_K y(t) \end{cases} \quad (\text{IV-8})$$

Avec $P(s)$ et $K(s)$ définis ci dessus, la boucle fermée $T_{zw}(s)$ admet la réalisation :

$$T_{zw}(s) : \begin{cases} \dot{x}_{cl}(t) = A_{cl} x_{cl}(t) + B_{cl} w(t) \\ z(t) = C_{cl} x_{cl}(t) + D_{cl} w(t) \end{cases} \quad (\text{IV-9})$$

$$x_{cl}^T = [x^T \quad x_K^T]$$

On supposera que $D_{12} \in \mathbb{R}^{p_1 \times m_2}$; $D_{21} \in \mathbb{R}^{p_2 \times m_1}$.

Avec : $m_1 > p_2$ et $p_1 > m_2$. Enfin n désignera la taille de A , (l'ordre du système $P(s)$).

La solution par variable d'état n'est applicable que sous les hypothèses suivantes :

Hypothèses 1:

(A1) (A, B_2, C_2) est stabilisable et détectable. Cette condition est nécessaire et suffisante pour l'existence d'un compensateur qui stabilise le système de manière interne.

(A2) Les matrices D_{12} et D_{21} sont de plein rang.

$$\text{(A3) } \text{rang} \begin{pmatrix} j\omega I - A & -B_2 \\ C_1 & D_{12} \end{pmatrix} = n + m_2 \quad \text{et} \quad \text{rang} \begin{pmatrix} j\omega I - A & -B_1 \\ -C_2 & D_{21} \end{pmatrix} = n + p_2$$

Pour tout : $\omega \in \mathbb{R}$ Autrement dit, $P_{12}(s)$ et $P_{21}(s)$ n'ont pas de zéro sur l'axe imaginaire. Ces deux dernières hypothèses sont appelées hypothèses de régularité. Dans un premier temps, nous ferons en plus les hypothèses simplificatrices suivantes dites de "normalisation":

(A4) normalisation: $D_{12}^T(D_{12}, C_1) = (I, 0)$ et $D_{21}(D_{21}^T, B_1^T) = (I, 0)$

(A5) $D_{22} = 0$ et $D_{11} = 0$ [23].

5. Techniques LMI pour la synthèse H_∞ :

En dehors de l'approche fondée sur les équations algébriques de Riccati et des techniques utilisant des représentations en transfert, il existe une approche plus récente pour la résolution du problème H_∞ . Il s'agit des techniques LMI (Linear Matrix Inequalities). Ces dernières utilisent une formulation du problème en termes d'inégalités matricielles linéaires en les variables X et Y. Du fait de la linéarité, l'ensemble ainsi décrit est convexe et il résulte de cette propriété que l'on peut aisément et avec une grande efficacité calculatoire extraire une solution particulière [30].

Dans le cadre H_∞ , les techniques LMI utilisent le lemme fondamental suivant.

Lemme : La solution du problème H_∞ est alors fondée sur l'utilisation du Lemme Réel Borné. $T_{zw}(s)$ est stable de manière interne et $\|T_{zw}(s)\|_\infty < \gamma$ si et seulement si il existe une matrice symétrique $P > 0$ telle que:

$$\begin{pmatrix} A_{cl}^T P + P A_{cl} & P B_{cl} & C_{cl}^T \\ B_{cl}^T P & -\mathcal{A} & D_{cl}^T \\ C_{cl} & D_{cl} & -\mathcal{A} \end{pmatrix} < 0 \quad (\text{IV-10})$$

P : représente la matrice de Lyapunov définie positive [30].

Les matrices inconnues de l'inégalité (IV-9) sont P et A_K, B_K, C_K et D_K . Par conséquent

(IV-10) n'est pas une LMI, c'est une inégalité matricielle bilinéaire BMI en P et A_K, B_K, C_K et D_K . Ce sont des inégalités difficiles à résoudre. Pour se ramener à un ensemble de LMI, nous effectuons un changement de variable linéarisant [31].

La matrice P peut être obtenue en effectuant une décomposition, où on déduit deux matrices non singulières $M, N \in \mathbb{R}^{n \times n}$ de rang plein vérifiant:

$MN^T = I - XY$. En partitionnant P et P^{-1} comme suit :

$$P = \begin{bmatrix} Y & N \\ N' & Y' \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} X & M \\ M & X' \end{bmatrix}, \quad \text{où : } X \in S_n, Y \in S_n.$$

S_n : sous-espace vectoriel des matrices symétriques de dimension n [29].

Théorème 1 : Il existe un compensateur $K(s)$ solution du problème H_∞ sub-optimal de performance γ , si et seulement si il existe des matrices symétriques X et Y solution du problème LMI :

$$\begin{bmatrix} \mathcal{N}_X & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} XA + A^T X & XB_1 & C_1^T \\ B_1^T X & -\gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix} \begin{bmatrix} \mathcal{N}_X & 0 \\ 0 & I \end{bmatrix} < 0 \quad (\text{IV-11})$$

$$\begin{bmatrix} \mathcal{N}_Y & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} YA^T + AY & YC_1^T & B_1 \\ C_1 Y & -\gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{bmatrix} \begin{bmatrix} \mathcal{N}_Y & 0 \\ 0 & I \end{bmatrix} < 0 \quad (\text{IV-12})$$

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0 \quad (\text{IV-13})$$

Où \mathcal{N}_X et \mathcal{N}_Y sont des bases des noyaux des matrices $[C_2 \ D_{21}]$ et $[B_2^T \ D_{12}^T]$.

Le système de LMI ci-dessus peut être aisément résolu par des algorithmes de programmation (convexe) semi-définie. Lorsqu'on dispose d'une solution des contraintes LMI (IV-11), (IV-12) le compensateur associé est directement obtenu par des relations algébriques de Théorème (2).

Théorème 2 : Supposons (A1)-(A5) et soit $\gamma > \gamma_{\text{opt}}$. Alors le compensateur

$$K_c = C_c(sI - A_c)^{-1}B_c \quad (\text{IV-14})$$

Avec :

$$A_K = A + (\gamma^{-2}B_1B_1^T - B_2B_2^T)X - (I - \gamma^{-2}YX)^{-1}YC_2^T C_2 \quad (\text{IV-15})$$

$$B_K = (I - \gamma^{-2}YX)^{-1}YC_2^T, \quad C_K = -B_2^T X$$

Stabilise le système de manière interne et satisfait :

$$\|F_l(P, K)\|_\infty < \gamma \quad (\text{IV-16})$$

Cette solution particulière du problème H_∞ sous-optimal est appelée compensateur central.

On notera que le compensateur central est strictement propre et d'ordre égal à celui du système $P(s)$, et l'optimum est caractérisé par $\rho(X Y) = \gamma_{opt}^2$

ρ : désigne le module de la plus grande valeur propre.

Ces formules deviennent singulières au voisinage de γ_{opt} ($-\gamma^{-2}YX$ n'est plus inversible à l'optimum). Cependant, on montre que K_c tend alors vers un compensateur d'ordre réduit, la chute d'ordre étant égale à la chute de rang de $(-\gamma_{opt}^{-2}YX)$. Cette réduction d'ordre provient de la simplification de pôle(s) à l'infini.

Pour conclure cette section, notons que lorsque $\gamma \rightarrow +\infty$, le compensateur central tend vers le compensateur de paramètres:

$$A_K = A + B_2 B_2^T X - Y C_2^T C_2 \quad (IV-17)$$

$$B_K = Y C_2^T, \quad C_K = -B_2^T X$$

Où X et Y sont les solutions stabilisantes des équations de Riccati [30]

$$A^T X + X A + X B_2 B_2^T X + C_1^T C_1 = 0 \quad (IV-18)$$

$$A Y + Y A^T + Y C_2^T C_2 Y + B_1 B_1^T = 0 \quad (IV-19)$$

Le but est celui de trouver des matrices A_K , B_K , C_K et D_K telles que la norme $\|T_{zw}(s)\|_\infty$ de la boucle fermée soit la plus petite possible :

$$\gamma_{opt} = \text{Min } \gamma \quad (IV-20)$$

$$\text{tel que : } \|T_{zw}(s)\|_\infty < \gamma$$

Donc (IV-10) peut être remplacée par les LMIs (IV-11), (IV-12) les nouvelles inconnues matricielles sont $\hat{A}, \hat{B}, \hat{C}, \hat{D}, X$ et Y

$$\begin{bmatrix} M_{11} & * & * & * \\ M_{21} & M_{22} & * & * \\ M_{31} & M_{32} & -\gamma I & * \\ M_{41} & M_{42} & M_{43} & -\gamma I \end{bmatrix} < 0 \quad (IV-21)$$

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0$$

On trouve la matrice (*) par le symétrique transposé.

$$\begin{cases} M_{11} = AX + XA^T + B_2\hat{C} + (B_2\hat{C})^T \\ M_{21} = \hat{A} + (A + B_2\hat{D}C_2)^T \\ M_{22} = A^TY + YA + \hat{B}C_2 + (\hat{B}C_2)^T \\ M_{31} = (B_1 + B_2\hat{D}D_{21})^T \\ M_{32} = (YB_1 + \hat{B}D_{21})^T \\ M_{41} = C_1X + D_{12}\hat{C} \\ M_{42} = C_1 + D_{12}\hat{D}C_2 \\ M_{43} = D_{11} + D_{12}\hat{D}D_{21} \end{cases} \quad (\text{IV-22})$$

Comme γ entre linéairement dans (IV-21) et (IV-22) il peut être directement minimisé par une optimisation LMI pour chercher la plus petite norme H_∞ atteignable.

Une fois les matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}, X$ et Y obtenues, la procédure de construction du contrôleur consiste à trouver des matrices non singulières M et N satisfaisant $MN^T = I - XY$ (une décomposition orthogonale triangulaire QR peut être utilisée), et ensuite à calculer le contrôleur par [32].

$$K(s): \begin{cases} D_K = \hat{D} \\ C_K = (\hat{C} + D_K C_2 X)(M^{-1})^T \\ B_K = N^{-1}(\hat{B} - Y B_2 D_K) \\ A_K = N^{-1}(\hat{A} - N B_K C_2 X - Y B_2 C_K M^T - Y(A + B_2 D_K C_2)X)(M^{-1})^T \end{cases} \quad (\text{IV-23})$$

Remarque. La méthodologie de synthèse d'un régulateur H_∞ proposée ci-dessus n'est pas la seule, ce problème peut être résolu avec un autre ensemble de LMI en passant par des compléments orthogonaux de certaines matrices.

6. Problème de performance H_2 :

Historiquement, l'utilisation de la norme H_2 , est apparue dans les années 1960 avec la théorie de la commande Linéaire Quadratique Gaussienne (LQG). La norme H_2 d'un transfert $w_2 \rightarrow z_2$ est une mesure précise de la puissance du signal de sortie z_2 pour un bruit blanc unitaire en entrée. Il s'agit d'un critère qui mesure la performance d'atténuation de la puissance des perturbations.

Définition (Norme H_2) : on appelle norme H_2 du transfert G entre w et z le réel positif défini par :

$$\|G(s)\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{trace}[G(-j\omega)^T G(j\omega)] d\omega = \sup_{W \neq 0} \frac{\|Z(s)\|_2}{\|W(s)\|_\infty} \quad (\text{IV-24})$$

La norme $\|G(s)\|_2$ est finie si et seulement si $G(s)$ est strictement propre ($D = 0$).

6.1. Calcul de la norme H_2 :

Pour calculer la norme H_2 d'une fonction de transfert : $G(s) = C(sI - A)^{-1}B$, avec $D = 0$. on supposera que : A est stable. En remarquant que $G(j\omega)$ est la transformée de Fourier de : $Ce^{At}B$, l'identité de Parseval donne[21]:

$$\int_0^{+\infty} B^T e^{tA^T} C^T C e^{tA} B dt = \frac{1}{2\pi} \int_0^{+\infty} G^T(-j\omega)G(j\omega) d\omega \quad (\text{IV-25})$$

On peut obtenir une expression algébrique précise de la norme H_2 . Pour son calcul on emploie les grammians d'observabilité X et de commandabilité Y , qui sont solutions des inégalités de Lyapunov suivantes:

$$\begin{cases} A^T X + XA + C^T C < 0 \\ AY + YA^T + BB^T < 0 \end{cases} \quad (\text{IV-26})$$

Si le système $G(s)$ est strictement propre ($D = 0$) on écrit:

$$\|G(s)\|_2^2 = \text{trace}\left[B^T \int_0^{+\infty} e^{A^T t} C^T C e^{At} dt B\right] = \text{trace}(B^T X B) \quad (\text{IV-27})$$

$$\text{Avec : } X = \int_0^{+\infty} e^{tA^T} C^T C e^{tA} dt$$

et :

$$\|G(s)\|_2^2 = \text{trace}\left[C \int_0^{+\infty} e^{At} B B^T e^{A^T t} dt C^T\right] = \text{trace}(C Y C^T) \quad (\text{IV-28})$$

$$\text{Avec : } Y = \int_0^{+\infty} e^{At} B B^T e^{A^T t} dt$$

Donc [7] :

$$\|G(s)\|_2^2 = \text{trace}(B^T X B) = \text{trace}(C Y C^T) \quad (\text{IV-29})$$

Le critère de performance H_2 assure que pour un bruit blanc Gaussien w , la variance de réponse z est bornée par γ , que pour une entrée scalaire w du type impulsion, l'énergie de la sortie z est inférieure à γ , et que pour une sortie scalaire la déviation de z est bornée en amplitude par γ quelque soit une entrée w d'énergie unitaire [33].

6.2. Problème de Synthèse H_2 :

Le but est celui de trouver des matrices A_K , B_K , C_K et D_K du contrôleur K telles que la norme H_2 de la boucle fermée $T_{zw}(s)$ est inférieure à γ_2 :

$$\|T_{zw}(s)\|_2 < \gamma_2 \quad (\text{IV-30})$$

Proposition: Supposons que le système LTI de fonction de transfert $G(s)$

$$\begin{cases} \dot{x}(t) = Ax(t) + Bw(t) \\ z(t) = Cx(t) + Dw(t) \end{cases} \quad (\text{IV-31})$$

Ce système est asymptotiquement stable et strictement propre ($D = 0$), les propositions suivantes sont équivalentes :

1. $\|G(j\omega)\|_2 < \gamma_2$

2. il existe $X = X^T > 0$ avec :

$$\text{Trace}(B^T X B) < \gamma_2^2$$

$$A^T X + X A + C^T C < 0$$

3. il existe $Y = Y^T > 0$ avec :

$$\text{Trace}(C Y C^T)$$

$$A Y + Y A^T + B B^T < 0$$

4. il existe $P_2 = P_2^T > 0$ et Z où les LMIs suivantes sont faisables :

$$\text{Trace}(Z) < \gamma_2^2$$

$$\begin{bmatrix} A^T P_2 + P_2 A & P_2 B \\ B^T P_2 & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} P_2 & C^T \\ C & Z \end{bmatrix} > 0$$

5. il existe $P_2 = P_2^T > 0$ et Z avec les LMIs suivantes sont faisables :

$$\text{Trace}(Z) < \gamma_2^2 \quad (\text{IV-32})$$

$$\begin{bmatrix} A^T P_2 + P_2 A & P_2 C^T \\ C P_2 & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} P_2 & B \\ B^T & Z \end{bmatrix} > 0$$

On peut écrire le problème de contrôle H₂ en boucle fermé comme suit [32] :

$$\text{Trace}(Z) < \gamma_2^2 \quad (\text{IV-33})$$

$$\begin{bmatrix} A_{cl}^T P_2 + P_2 A_{cl} & P_2 B_{cl} \\ B_{cl}^T P_2 & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} P_2 & C_{cl}^T \\ C_{cl} & Z \end{bmatrix} > 0$$

Théorème 3 : La norme H₂ d'un système G , est obtenue par le problème d'optimisation suivant [34] :

$\text{Min Trace}(Z)$

$$A^T P_2 + P_2 A + C C^T < 0$$

$$B^T P_2 B < Z$$

$$P_2 > 0$$

Soit la boucle de synthèse optimale H₂ suivante :

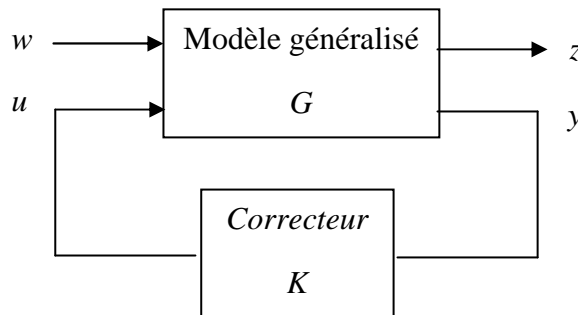


Figure IV.2 : La synthèse optimale H₂

$G(s)$ est donné par la représentation suivante :

$$G(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \quad (\text{IV-34})$$

La synthèse optimale H₂ consiste à déterminer le correcteur K minimisant une norme H₂ du transfert entre les sorties exogènes z et les entrées exogènes w :

$$\min_{K \in \mathcal{K}} \|T_{zw}\|_2 \quad (\text{IV-35})$$

- $\gamma > 0$ et $G(s)$ étant donnés, déterminer un correcteur K tel que : $\|T_{zw}\|_2 \leq \gamma$

\mathcal{K} : est l'ensemble des compensateurs K stabilisant l'interconnexion de manière interne ou ensemble des compensateurs admissibles.

Hypothèses 2:

1- (A, B_1) et (A, B_2) stabilisables, (C_1, A) et (C_2, A) détectables.

2- $D_{12}^T [C_1 \ D_{12}] = [0 \ I]$ et $[B_1^T \ D_{21}^T] D_{21}^T = [0 \ I]^T$.

3- soit G et K rationnelles, réelles et propres.

4- $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ et $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ sont de rang plein $\forall \omega$

Définissant les deux équations de Riccati algébriques duales :

$$A'X_2 + X_2A - X_2B_2B_2'X_2 + C_1'C_1 = 0$$

$$AY_2 + Y_2A' - Y_2C_2'C_2Y_2 + B_1B_1' = 0$$

Avec :

$$F_2 = -B_2'X_2, \quad L_2 = -Y_2C_2' \quad \text{et} \quad \hat{A}_2 = A + B_2F_2 + L_2C_2$$

Théorème 3: L'unique compensateur H₂ optimal est donné par:

$$K_{H_2}^* \sim \begin{bmatrix} \hat{A}_2 & -L_2 \\ F_2 & 0 \end{bmatrix} \quad (\text{IV-36})$$

Où :

$$\gamma^* = \min_{K \in \mathcal{K}} \|T_{zw}\|_2^2$$

Théorème 4: Si les hypothèses 2 sont vérifiées, $\gamma > 0$ et G étant données, l'ensemble de tous les correcteurs H₂ sous-optimaux i.e. $\|T_{zw}\|_2 \leq \gamma$ sont données par [35] :

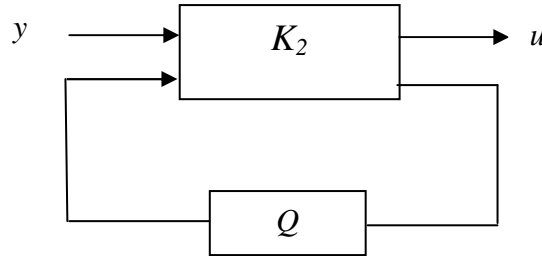


Figure IV.3 correcteur H₂ sous optimal.

$$K_{H_2} = \mathcal{L}_l(Q, K_2), \quad \text{avec} : Q \in \mathcal{RH}^{m_2 \times p_2}$$

Le compensateur H₂ sous optimal est donné par

$$K_2 = \begin{bmatrix} \hat{A}_2 & -L_2 & B_2 \\ F_2 & 0 & I_{m_2} \\ -C_2 & I_{p_2} & 0 \end{bmatrix} \quad (\text{IV-37})$$

6.2.1. Solution du problème de synthèse H₂ par les formulations LMIs:

On trouve le correcteur $K(s)$ qui résous le problème de synthèse H₂ par retour de sortie dynamique, si et seulement si il existe les matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}, X$ et Y qui vérifient les LMIs suivantes pour minimiser γ_2 :

$$\text{Trace}(Z) < \gamma_2 \quad (\text{IV-38})$$

$$\begin{bmatrix} M_{11} & M_{21}^T & M_{31}^T \\ M_{21} & M_{22} & M_{32}^T \\ M_{31} & M_{32} & M_{33} \end{bmatrix} < 0$$

$$\begin{bmatrix} N_{11} & N_{21}^T & N_{31}^T \\ N_{21} & N_{22} & N_{32}^T \\ N_{31} & N_{32} & N_{33} \end{bmatrix} < 0$$

Avec :

$$\left\{ \begin{array}{l} M_{11} = AX + XA^T + B_2\hat{C} + (B_2\hat{C})^T \\ M_{21} = \hat{A} + (A + B_2\hat{D}C_2)^T \\ M_{22} = A^TY + YA + \hat{B}C_2 + (\hat{B}C_2)^T \\ M_{31} = (B_1 + B_2\hat{D}D_{21})^T \\ M_{32} = (YB_1 + \hat{B}D_{21})^T \\ M_{33} = -I_{nu} \\ N_{11} = X \\ N_{21} = I_n \\ N_{22} = Y \\ N_{31} = C_1X + D_{12}\hat{C} \\ N_{32} = C_1 + D_{12}\hat{D}C_2 \\ N_{33} = Z \end{array} \right. \quad (\text{IV-39})$$

On trouve $K(s)$ par l'utilisation de formulation (IV-23) [32].

7. Problème multi-objectifs mixte H₂/H_∞ :

Cette idée a été étudiée à la fin des années 80, en prenant la forme de la définition du problème mixte H₂/H_∞ dont le schéma standard est représenté par la figure IV.4

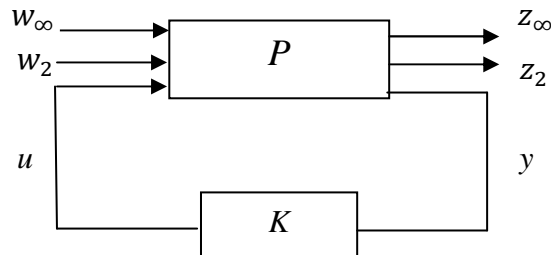


Figure IV.4. Modèle standard du problème H₂/H_∞

La matrice de transfert P est usuellement partitionnée suivant les signaux d'entrée et de sortie.

$$P(s) = \begin{bmatrix} P_{\infty} & P_{\infty 2} & P_{\infty u} \\ P_{2\infty} & P_2 & P_{2u} \\ P_{y\infty} & P_{y2} & P_{yu} \end{bmatrix} \quad (\text{IV-40})$$

La réalisation du modèle d'état généralisé est donnée par :

$$P(s) = \begin{bmatrix} A & B_{\infty} & B_2 & B \\ C_{\infty} & D_{\infty} & 0 & D_{\infty u} \\ C_2 & 0 & D_2 & D_{2u} \\ C & D_{y\infty} & D_{y2} & 0 \end{bmatrix} \quad (\text{IV-41})$$

En définissant les matrices de transfert obtenues comme des transformations fractionnaires linéaires de K sur la partition adéquate de P :

$$\begin{cases} T_{z_{\infty}w_{\infty}}(K) = P_{\infty} + P_{\infty u}K(I - P_{yu}K)^{-1}P_{y\infty} \\ T_{z_2w_2}(K) = P_2 + P_{2u}K(I - P_{yu}K)^{-1}P_{y2} \end{cases} \quad (\text{IV-42})$$

7.1. Problème H₂/H_∞ avec retour de sortie dynamique d'ordre plein :

Etant donné un niveau de performance H_{∞} admissible γ , déterminer un retour de sortie admissible tel que:

$$\inf_{(K)} \|T_{z_2w_2}(K)\|_2 \quad (\text{IV-43})$$

$$\text{Sous: } \|T_{z_{\infty}w_{\infty}}(K)\|_{\infty} \leq \gamma$$

L'objectif est de résoudre un problème d'optimisation des performances nominales de la boucle fermée, mesurées par la norme H_2 , sous une contrainte de stabilité robuste exprimée par la norme H_{∞} . Le problème mixte H_2/H_{∞} permet d'intégrer ainsi l'exigence de performances optimisées et de robustesse en un seul correcteur.

Le système bouclé par un tel correcteur s'écrit alors :

$$\begin{cases} \dot{\mathcal{X}} = \mathcal{A}\mathcal{X} + \mathcal{B}_{\infty}w_{\infty} + \mathcal{B}_2w_2 \\ z_{\infty} = \mathcal{C}_{\infty}\mathcal{X} + \mathcal{D}_{\infty}w_{\infty} + \mathcal{D}_{\infty 2}w_2 \\ z_2 = \mathcal{C}_2\mathcal{X} + \mathcal{D}_2w_2 + \mathcal{D}_{1\infty}w_2 \end{cases} \quad (\text{IV-44})$$

Avec : $\mathcal{X} = [x \quad x_K]^T \in \mathbb{R}^{2n}$

x_K : Le vecteur d'état du correcteur K , $x_K \in \mathbb{R}^n$

Où ;

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} A + BD_K C & BC_K \\ B_K C & A_K \end{bmatrix}, \quad \mathcal{B}_\infty = \begin{bmatrix} B_\infty + BD_K D_{y\infty} \\ B_K D_{y\infty} \end{bmatrix}, \quad \mathcal{B}_2 = \begin{bmatrix} B_2 + BD_K D_{y2} \\ B_K D_{y2} \end{bmatrix} \\ \mathcal{C}_\infty &= [C_\infty + D_{\infty u} D_K C \quad D_{\infty u} C_K], \quad \mathcal{C}_2 = [C_2 + D_{2u} D_K C \quad D_{2u} C_K] \\ \mathcal{D}_\infty &= [D_\infty + D_{\infty u} D_K D_{y\infty}], \quad \mathcal{D}_{\infty 2} = [D_{\infty 2} + D_{\infty u} D_K D_{y2}] \\ \mathcal{D}_{2\infty} &= [D_{2\infty} + D_{2u} D_K D_{y\infty}], \quad \mathcal{D}_2 = [D_2 + D_{2u} D_K D_{y2}] \end{aligned}$$

7.2. Formulation par la théorie de Lyapunov :

La théorie de Lyapunov permet de reformuler et de paramétrer le problème 7.1 à l'aide de variables matricielles additionnelles solutions d'équations de Riccati et de Lyapunov ou d'inégalités matricielles.

7.2.1. Formulation par inégalités matricielles:

Etant donné un niveau de performance H ∞ admissible γ , le problème mixte est de déterminer un correcteur K et deux matrices de Lyapunov $P_2 \in S_n^+$, $P_\infty \in S_n^+$ solutions de :

$$\begin{aligned} \text{Min}_{K, P_2, P_\infty} \text{Trace}(P_2 \mathcal{C}'_2 \mathcal{C}_2) & \tag{IV-45} \\ \begin{bmatrix} \mathcal{A}P_2 + P_2 \mathcal{A}' & \mathcal{B}_2 \\ \mathcal{B}'_2 & -I \end{bmatrix} & < 0 \\ \begin{bmatrix} \mathcal{A}P_\infty + P_\infty \mathcal{A}' & \mathcal{B}_\infty & P_\infty \mathcal{C}_\infty \\ \mathcal{B}'_\infty & -I & 0 \\ \mathcal{C}'_\infty P_\infty & 0 & -\gamma^2 I \end{bmatrix} & < 0 \end{aligned}$$

$$\text{Où : } \mathcal{D}_2 = D_2 + D_{2u} D_K D_{y2} = 0$$

La formulation (IV-44) fait clairement apparaître le caractère bilinéaire du problème d'optimisation associé au problème mixte à travers les multiplications entre les variables de décision définissant le correcteur et les variables de décision liées à la théorie de Lyapunov.

7.3. Approches par relaxation convexe

Comme nous venons de le voir, les approches numériques directes ne sont pas réellement bien adaptées à la solution du problème mixte. Un certain nombre d'approches fondées sur des relaxations convexes obtenues par la définition de problèmes associés permettent de donner des solutions sous-optimales au problème initial.

La modélisation comprend un signal d'entrée exogène commun et deux sorties exogènes distinctes. Le compensateur recherché est choisi strictement propre ($D_K = 0$) d'ordre fixé

$$n_k = n$$

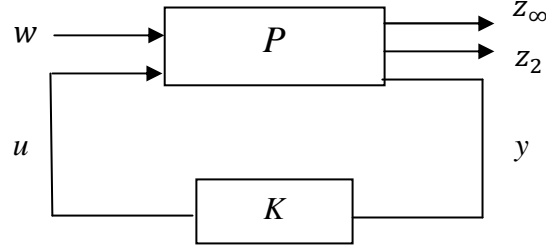


Figure IV.5. Problèmes H₂/H_∞ à une entrée - deux sorties.

Le modèle du système est le même que précédemment avec $w_2 = w_\infty$ et par conséquent :

$$B_\infty = B_2 = B_{\infty 2}, \quad D_{y_\infty} = D_{y_2} = D_{y_\infty 2}, \quad D_2 = D_\infty = 0$$

La réalisation minimale d'état du modèle en boucle fermée est :

$$\begin{cases} \dot{\mathcal{X}} = \mathcal{A}\mathcal{X} + \mathcal{B}_{\infty 2}w \\ z_\infty = \mathcal{C}_\infty\mathcal{X} + \mathcal{D}_\infty w \\ z_2 = \mathcal{C}_2\mathcal{X} + \mathcal{D}_2 w \end{cases} \quad (\text{IV-46})$$

Il est possible dans le cadre le plus général du modèle (IV-42) de proposer une relaxation convexe du problème mixte initial par l'utilisation conjointe d'un changement de variables linéarisant et la relaxation connue sous le nom de « Lyapunov Shaping Paradigm ». Pour cela, nous reprenons la formulation du problème mixte par inégalités matricielles.

Les inégalités présentées dans (IV-45) ne sont pas conjointement linéaires en K , P_2 et P_∞ . L'idée principale de la relaxation consiste à ne rechercher qu'une seule et unique matrice de Lyapunov attestant de la performance H_2 et de la contrainte H_∞ .

$$P_\infty = P_2 = P \quad (\text{IV-47})$$

Ce choix introduit un certain degré de pessimisme mais permet par ailleurs de montrer à l'aide d'un changement de variables linéarisant que le nouveau problème ainsi posé est un problème d'optimisation LMI. En effet, en partitionnant P et P^{-1} comme :

$$P = \begin{bmatrix} Y & N \\ N' & Y \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} X & M \\ M & X \end{bmatrix}$$

Où : $X \in S_n$, $Y \in S_n$, $M \in \mathbb{R}^{n \times n}$, $N \in \mathbb{R}^{n \times n}$

S_n : sous-espace vectoriel des matrices symétriques de dimension n

Le changement de variables inversible est défini par :

$$\begin{cases} \hat{D} = D_K \\ \hat{C} = C_K M + D_K C X \\ \hat{B} = N B_K + Y B D_K \\ \hat{A} = N A_K M^T + N B_K C X + Y B C_K M + Y(A + B D_K C) X \end{cases} \quad (\text{IV-48})$$

Après application du changement de variables linéarisant et de la relaxation « Lyapunov shaping paradigm », la relaxation convexe du problème mixte est définie par :

$$\text{Min}_{X,Y,Z,\hat{A},\hat{B},\hat{D},\hat{C}} \text{Trace}(Z) \quad (\text{IV-49})$$

$$\begin{bmatrix} AX + XA^T + B\hat{C} + \hat{C}^T B^T & * & * & * \\ \hat{A} + (A + B\hat{D}C)^T & A^T Y + YA + \hat{B}C + (\hat{B}C)^T & * & * \\ (B_\infty + B_2 \hat{D} D_{y_\infty})^T & (YB_1 + \hat{B} D_{21})^T & -\gamma I & * \\ C_\infty X + D_{\infty u} \hat{C} & C_\infty + D_{\infty u} \hat{D} C & D_\infty + D_{\infty u} \hat{D} D_{y_\infty} & -\gamma I \end{bmatrix} < 0$$

$$\begin{bmatrix} AX + XA^T + B\hat{C} + \hat{C}^T B^T & * & * \\ \hat{A} + (A + B\hat{D}C)^T & A^T Y + YA + \hat{B}C + (\hat{B}C)^T & * \\ (B_2 + B\hat{D} D_{y_2})^T & (YB_2 + \hat{B} D_{y_2})^T & -I \end{bmatrix} < 0$$

$$\begin{bmatrix} X & * & * \\ I & Y & * \\ C_2 X + D_{2u} \hat{C} & C_2 + D_{2u} \hat{D} C & Z \end{bmatrix} < 0$$

$$\text{Où : } D_2 + D_{2u} \hat{D} D_{y_2} = 0$$

$$X \in S_n^{++}, Y \in S_n^{++}, Z \in S_r^{++}, \hat{A} \in \mathbb{R}^{n \times n}, \hat{B} \in \mathbb{R}^{n \times r}, \hat{C} \in \mathbb{R}^{m \times n}, \hat{D} \in \mathbb{R}^{m \times r}$$

S_n^{++} : Cône convexe des matrices symétriques semi définies positives de dimension n .

Donc on écrit le contrôleur $K(s)$ comme suit [31]:

$$K(s): \begin{cases} D_K = \hat{D} \\ C_K = (\hat{C} + D_K C X)(M^{-1})^T \\ B_K = N^{-1}(\hat{B} - Y B D_K) \\ A_K = N^{-1}(\hat{A} - N B_K C X - Y B C_K M^T - Y(A + B D_K C) X)(M^{-1})^T \end{cases} \quad (\text{IV-50})$$

8. Exemple (Problème de Benchmark)

Le système se compose de deux masses et un ressort avec des paramètres incertains comme le montre la figure IV.6

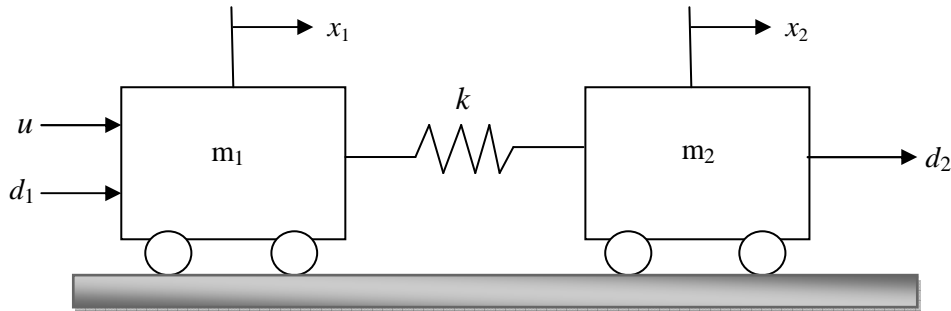


Figure IV.6. Deux masses - ressort avec paramètres incertains

Supposant que : $m_1 = m_2 = 1$, $k = 1$

Le système défini par la représentation suivante :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & -k/m_1 & 0 & 0 \\ -k/m_2 & -k/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} (u + d_1) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_1 \end{bmatrix} d_2 \quad (\text{IV-51})$$

$$y = [0 \ 1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (\text{IV-52})$$

x_1, x_2 positions des deux masses 1 et 2 respectivement

x_3, x_4 correspondent les vitesses de masse 1 et 2

u la force de contrôle appliqué sur la masse 1

d_1, d_2 entrées de perturbation

y la position mesurée de la masse 2

soit la représentation d'état du système précédant :

$$\begin{cases} \dot{x}(t) = Ax + B_u u + B_d d \\ y = C_y x + D_u u + D_d d \end{cases} \quad (\text{IV-53})$$

avec : $C_y = I$, $D_u = 0$ et $D_d = [0 \ 0]$, $B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_1 & 0 \\ 0 & 1/m_1 \end{bmatrix}$

En utilise la correction par retour de sortie avec: $u = Ky$

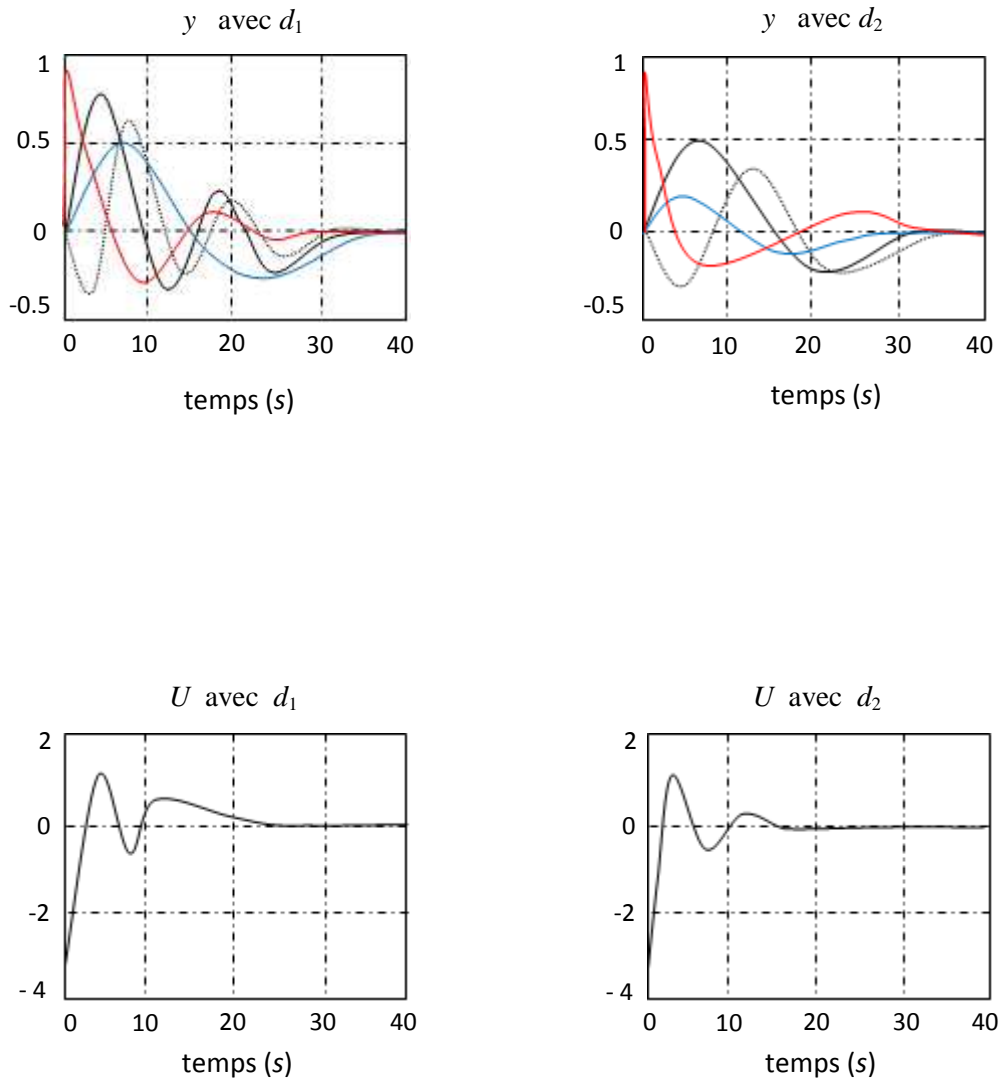
Où : $y = x_2$ et $0.5 \leq k \leq 2$

Le paramètre k et varie entre $0.5 \leq k \leq 2$ donc k est une incertitude paramétrique

k : paramètre incertain.

Avec d_1, d_2 entrées de perturbation.

La régulation de notre système, est un cas particulier du suivi où le signal de référence est nul, c'est-à-dire on l'on désire maintenir la sortie y et les états du système en $BF(x_1, x_2, x_3, x_4)$ à zéro.

8.1. Solution de problème H₂ avec LMI :Pour : $\gamma_2 = 0.7$ **Figure IV.7 :** les réponses Impulsionnelle et la commande H₂ avec LMI

8. 2. Solution de problème H_∞ avec LMI :

Pour : $\gamma_\infty = 0.7$

Pour un niveau d'atténuation de perturbations spécifié : $\gamma_\infty > 0.7$, on veut synthétiser un correcteur pour le système (IV-51), tel que le système global résultant en boucle fermée $T(K)$ soit stable et vérifie : $\|T(K)\|_\infty \leq \gamma_\infty$

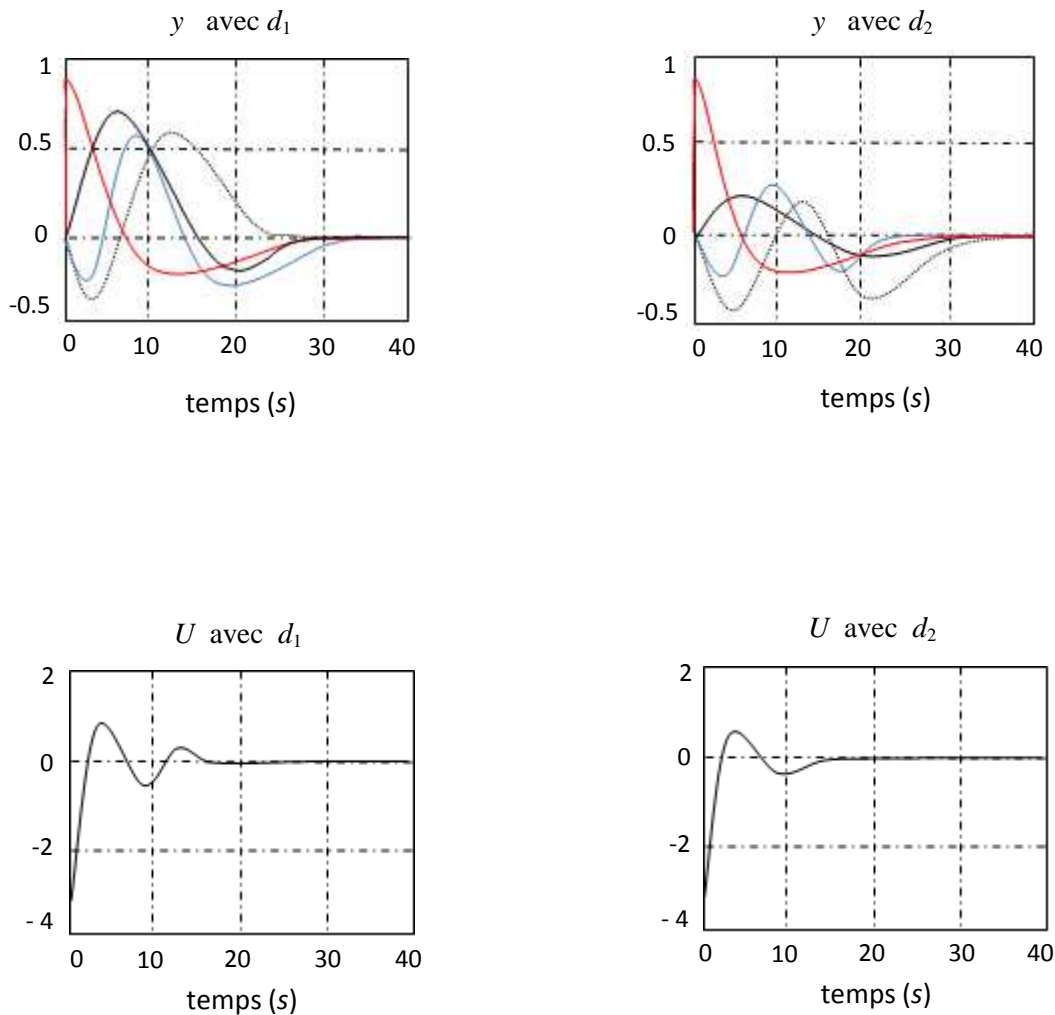
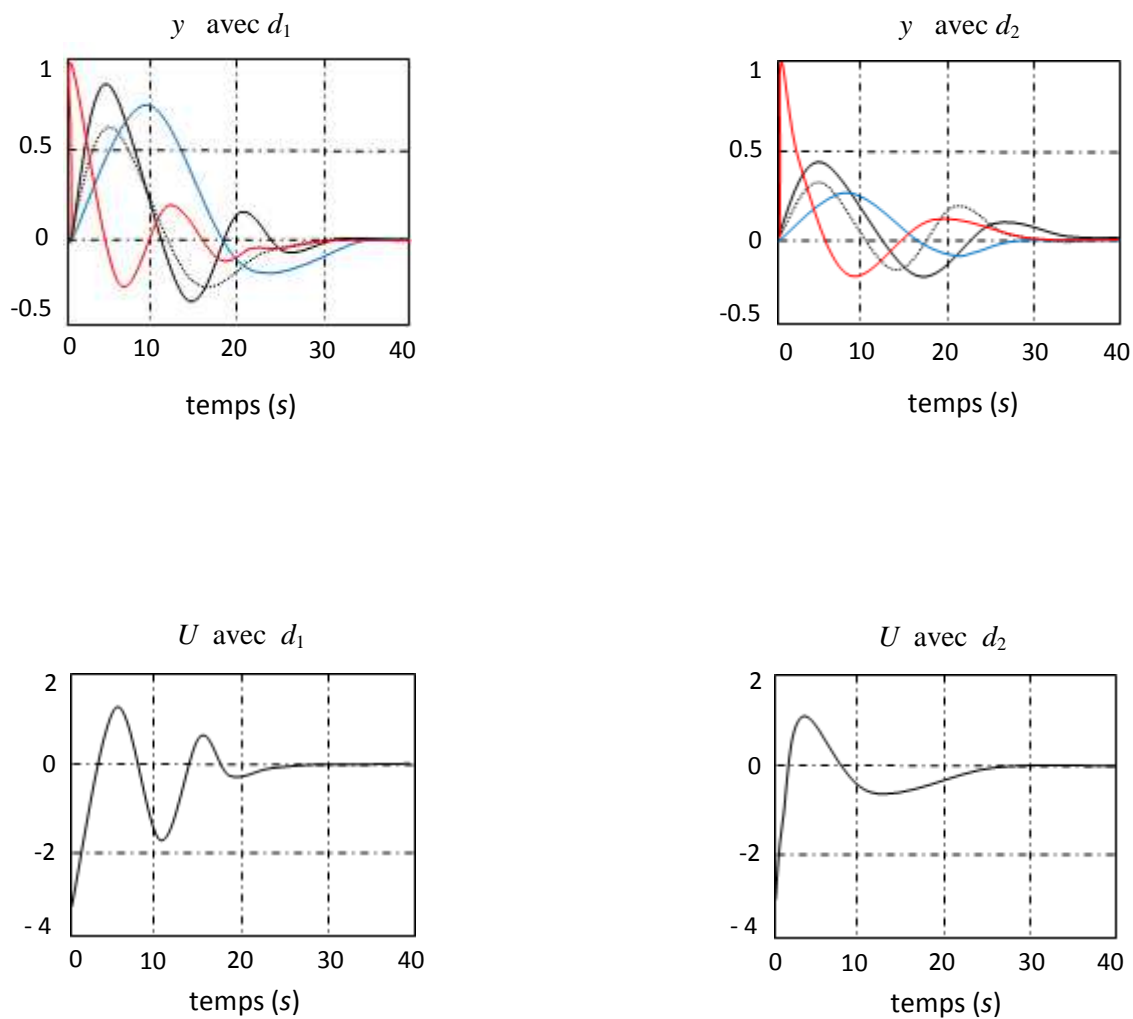


Figure IV.8 : les réponses Impulsionnelle et la commande H_∞ avec LMI

On constate que la synthèse H_∞ fournit une commande robuste vis-à-vis des perturbations d_1, d_2 . De plus, le correcteur K est pris en compte l'incertitude paramétrique (k) du modèle du système, donc on peut dire que la commande H_∞ assure la stabilité en boucle fermée de système étudié et également, il est robuste vis-à-vis les incertitudes et les perturbations.

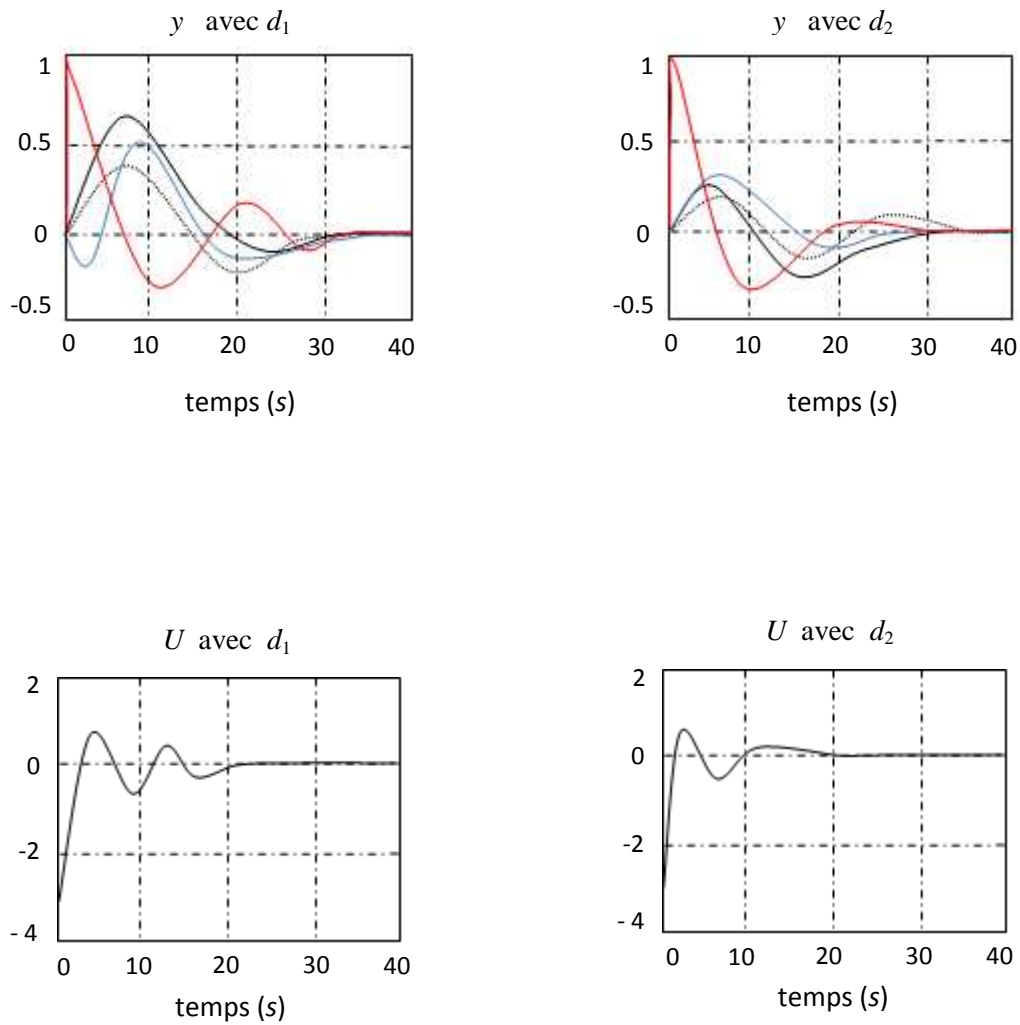
8.3. Solution de problème H_2 avec MOGA :

Population de 200 -500

**Figure IV.9 :** les réponses Impulsionnelle et la commande H_2 avec MOGA

8.4. Solution de problème H ∞ avec MOGA :

Population de 200 -500

**Figure IV.10 :** les réponses Impulsionnelle et la commande H ∞ avec MOGA

8.5. Problème multi-objectifs H_2/H_∞ :**8.5.1. Solution de problème H_2/H_∞ avec LMI :**

Pour : $\gamma = \gamma_2 = \gamma_\infty = 0.7$

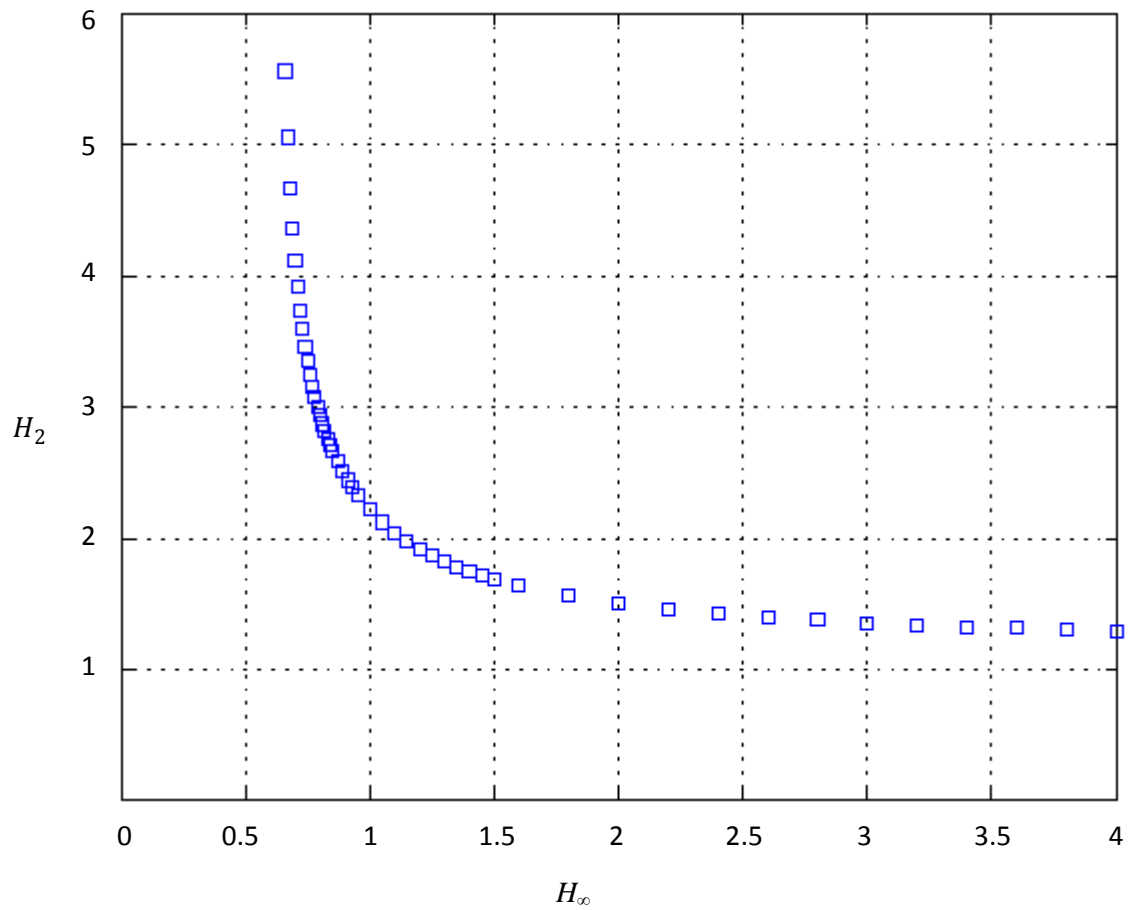
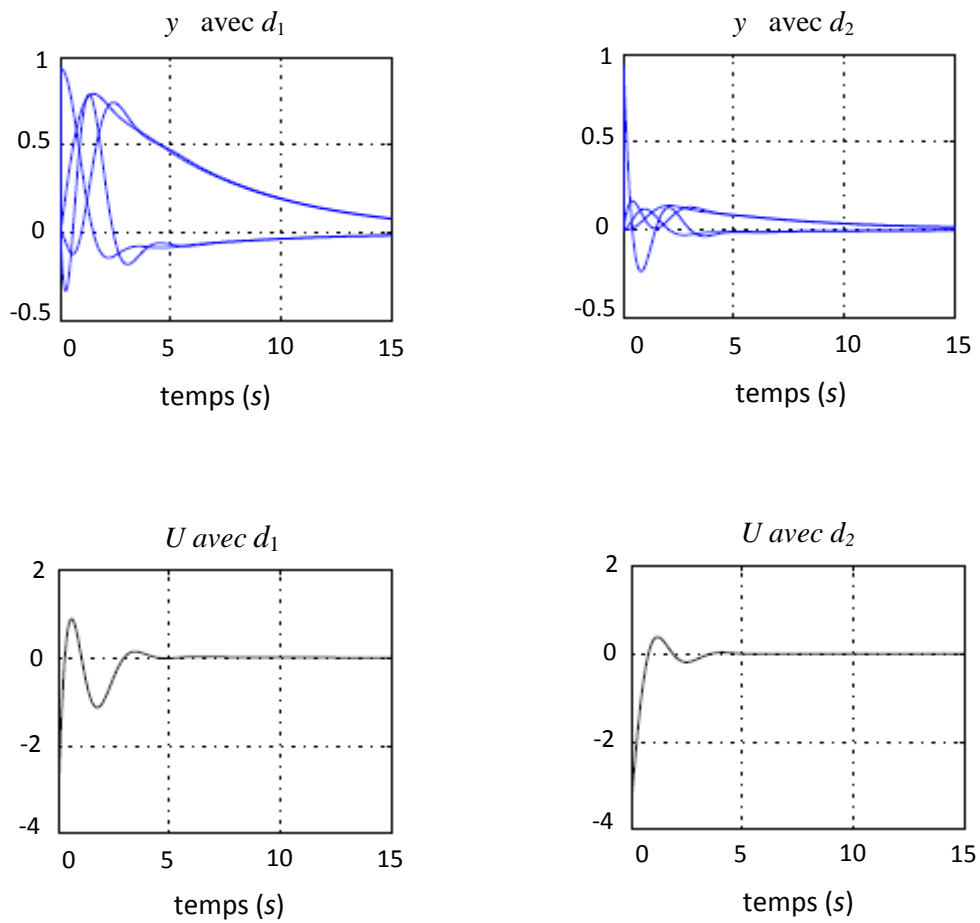
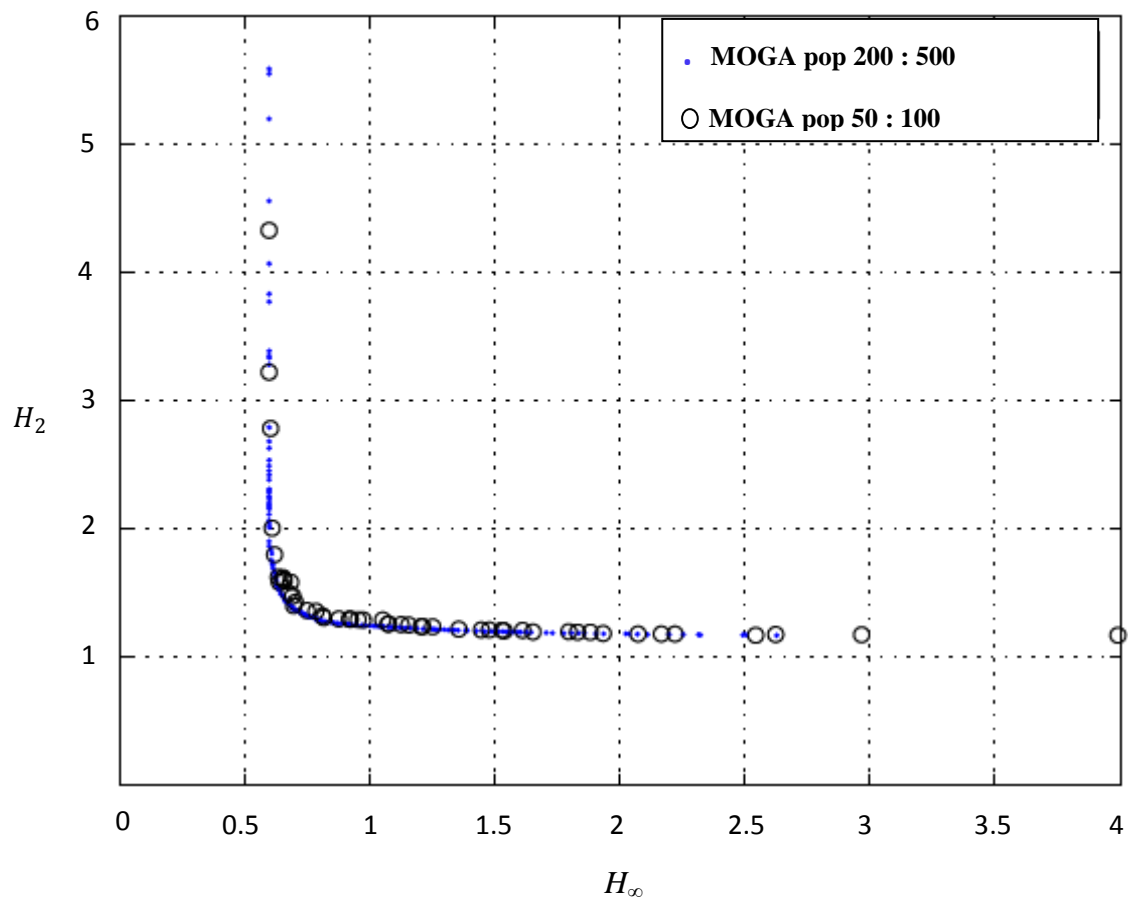


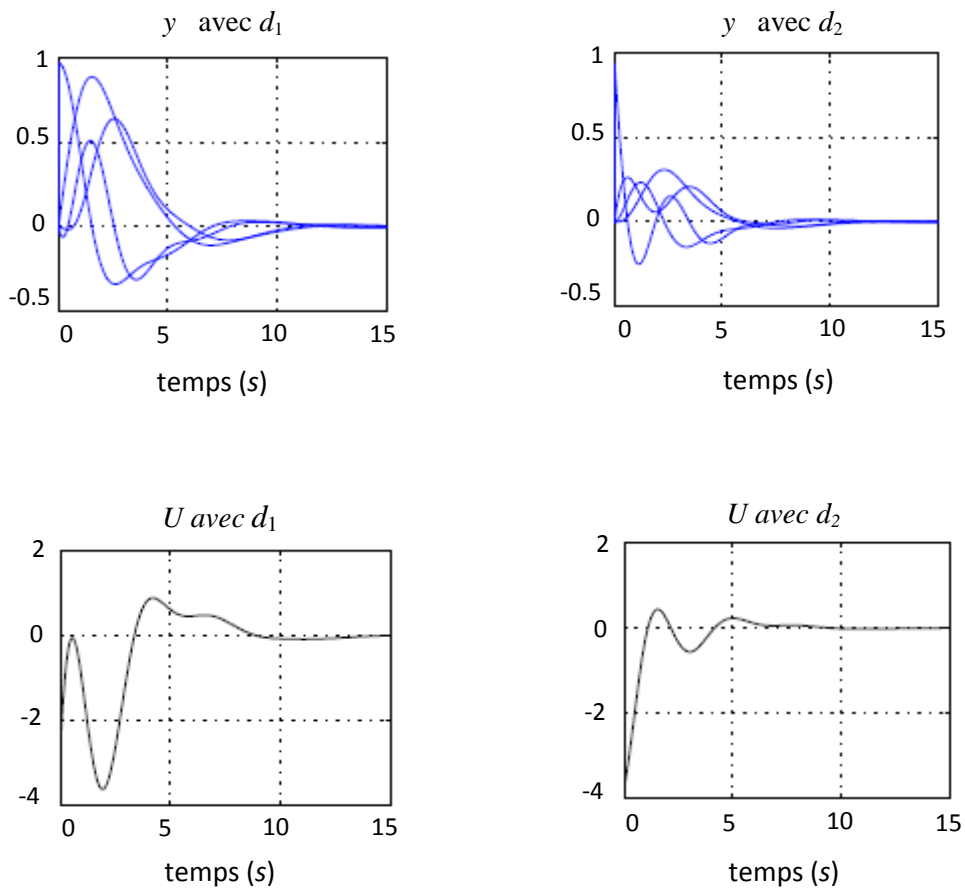
Figure IV.11 résolution du problème H_2/H_∞ avec LMI

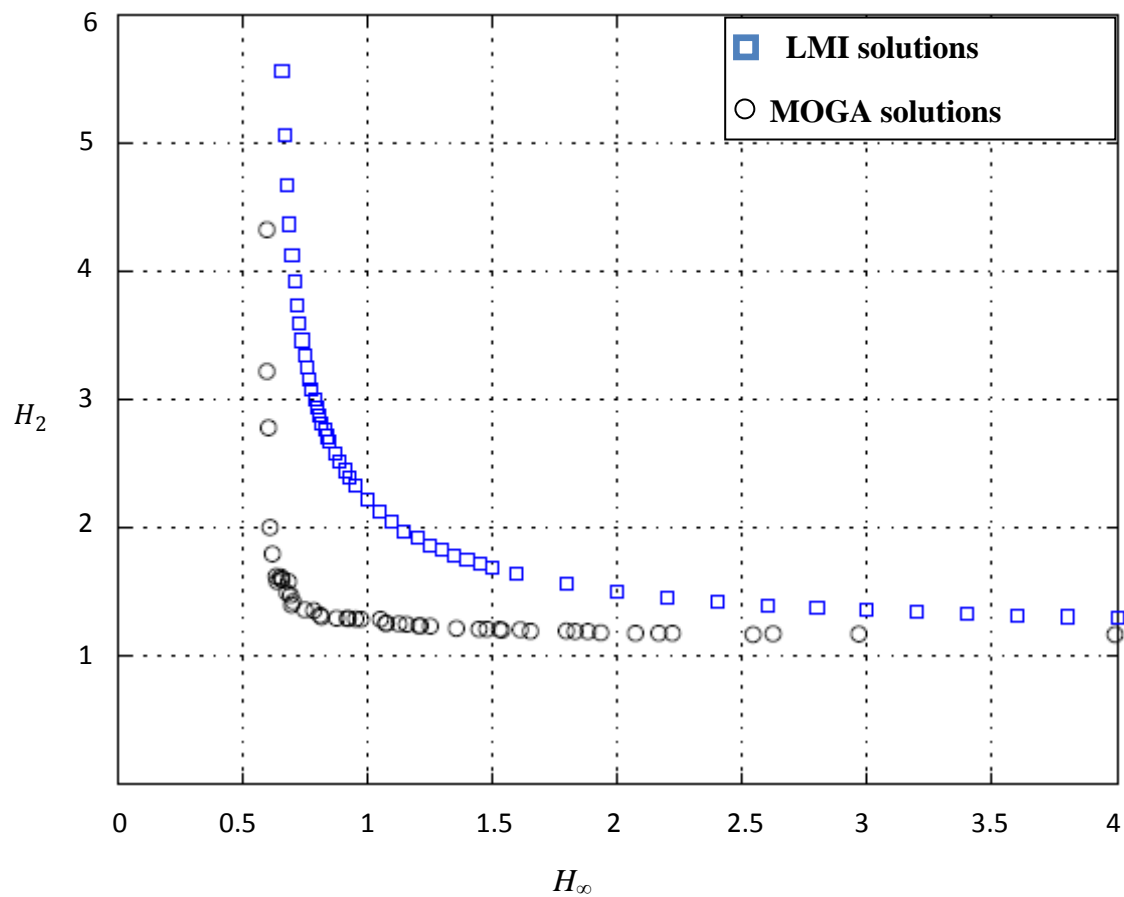
8.5.1.1. Les réponses du système avec l'approche LMI :**Figure IV.12 :** les réponses Impulsionnelle et la commande avec LMI

8.5.2. Solution de problème H_2/H_∞ avec MOGA :

Population de 50-100 et de 200 -500

**Figure IV.13 :** résolution du problème H_2/H_∞ avec MOGA

8.5.2.1. Les réponses du système avec l'approche MOGA :**Figure IV.14** : les réponses Impulsionnelle et la commande avec MOGA

8.6. Comparaison entre les résultats de la commande LMI et MOGA :**Figure IV.15** Comparaison entre les résultats de LMI et MOGA

L'approche MOGA donne des meilleures solutions que l'approche LMI, la figure IV.15 présentée un front de Pareto de l'approche MOGA est mieux que le front de l'approche LMI puisqu'il conforme avec les exigences de la commande multi-objectifs qui exige d'atteindre le minimum de la norme H₂ du système en BF : $Min (\|T(K)\|_2)$ avec le contrainte H_∞ ($\|T(K)\|_∞ \leq 0.7$). Donc en trouve sur le front de l'allure MOGA le Pareto optimale qui contient les solutions efficaces, (non dominées), solutions non dominées par des autres points de front de Pareto.

On constate que les réponses du système sont stabilisées sur le consigne zéro après 15 secondes dans l'approche LMI, mais ils sont stabilisées après 10 secondes avec l'approche MOGA, malgré la présence des entrées de perturbation d_1, d_2 et l'incertitude paramétrique, donc en peut dire que les deux commande LMI et MOGA sont des commandes robustes vis-à-vis les perturbations et l'incertitude.

Dans l'approche LMI et avec l'entrée de perturbation d_1 en trouve un taux de dépassement sur les états supérieur que le dépassement de l'approche MOGA.

Une condition nécessaire pour les deux approches LMI et MOGA pour trouver l'optimum des solutions est la convexité de l'ensemble des objectifs

Les résultats de simulation montrent l'efficacité et les performances de deux types de commande LMI et MOGA. Ces commandes assurées, la stabilisation des systèmes et la robustesse vis-à-vis les perturbations et les incertitudes, donc d'après cette étude en peut dire :

L'approche LMI est basée sur le problème de faisabilité LMI, c'est-à-dire la résolution de problème multi-objectifs avec LMI est conditionnée par l'existence des matrice $X, Y, Z, \hat{A}, \hat{B}, \hat{D}, \hat{C}$ qui sont formulés le correcteur et aussi la condition de convexité de l'ensemble des objectifs à optimisés mais cette technique de résolution est compliqué sur tous le choix de γ optimal.

L'approche MOGA est basée sur l'idée de développer une population initiale, qui est choisie aléatoirement et utiliser la notion de Pareto-optimalité (dominance de Pareto) pour faire la sélection multicritères. Cette notion de Pareto optimalité permet d'établir une relation de dominance entre les individus portant sur plusieurs critères, pour atteindre des solutions Pareto optimales dans tout le processus d'optimisation mais la réalisation du choix faux des paramètres peut produire excessivement des mauvais résultats.

D'après cette étude et ces résultats de simulation, on peut dire que le contrôleur de l'approche MOGA satisfait les exigences du contrôle multi-objectifs H_2/H_∞ d'une manière efficace et avec une simplicité de calcul des paramètres de contrôleur par contre l'approche LMI est plus compliquée.

9. Conclusion :

Dans cette partie, nous avons abordé le problème de la commande multi-objectifs H_2/H_∞ par retour de sortie. Les différents critères de performances considérés incluent la stabilisation du système en boucle fermée, des performances H_2 ainsi que des performances H_∞ . Nous avons proposé une caractérisation LMI des correcteurs stabilisant le système en boucle fermée et assurant des niveaux de réjection de perturbations.

L'application de deux types de commande LMI et MOGA sur un exemple de simulation (le système se compose de deux masses et un ressort avec des paramètres incertains) a donné de bons résultats pour la stabilisation et la robustesse de ces commandes.

Conclusions et Perspectives

Dans ce mémoire, on a étudié le problème de commande multi-objectifs en utilisant la technique des inégalités matricielles linéaires (LMIs) et les algorithmes génétiques.

Le problème H_∞ consiste à concevoir une commande assurant la stabilité asymptotique du système bouclé avec un certain taux d'atténuation de l'influence des entrées exogènes sur les sorties du système. Ces entrées sont constituées des signaux de consignes et éventuellement des perturbations liées à l'environnement du système. La synthèse H_∞ fournit une commande robuste vis-à-vis de ces perturbations et les incertitudes. La performance H_2 , est apparue dans les années 1960 avec la théorie de la commande Linéaire Quadratique Gaussienne (LQG). Il s'agit d'un critère qui mesure la performance d'atténuation de la puissance des perturbations donc le problème multi-objectifs mixte H_2/H_∞ permet d'intégrer l'exigence de performances optimisées et de robustesse en un seul correcteur, lorsque en résoudre un problème d'optimisation des performances nominales de la boucle fermée, mesurées par la norme H_2 , sous une contrainte de stabilité robuste exprimée par la norme H_∞ .

L'approche LMI est utilisée pour la résolution du problème multi-objectifs H_2/H_∞ . Il s'agit des techniques LMI (Linear Matrix Inequalities). Ce problème utilise des formulations en termes d'inégalités matricielles linéaires en les variables X et Y , et ainsi la programmation semi définie (SDP), qui considère comme un problème d'optimisation convexe utilise les formulations des inégalités matricielles linéaires (LMIs) pour contrôler plusieurs spécifications et contraintes convexes. De cette manière et à grâce les concepts des LMIs l'unification des contraintes d'optimisation multi-objectifs est devient possible.

Les algorithmes génétiques (AG's) semblent être une solution intéressante pour résoudre le problème d'optimisation multiobjectifs. L'algorithme génétique multiobjectif (MOGA) a été appliqué avec succès aux problèmes de contrôle des systèmes, par exemple : contrôleur d'un système MIMO et l'optimisation des paramètres, contrôle des systèmes par la commande H_∞ .

MOGA est un algorithme évolutionnaire emploie les opérateurs suivantes :(le rang de Pareto optimal, Fitness Sharing, restriction d'accouplement, sélection, croisement et mutation).

Le principe de MOGA est de développer une population des solutions Pareto optimales ou proches de Pareto optimales dans tout le processus d'optimisation.

Enfin et pour illustrer l'efficacité de deux approches étudiées, un exemple de simulation est présenté (Problème de Benchmark ce système est se compose de deux masses et un ressort avec des paramètres incertains). Les résultats de simulation montrent le rôle important que la commande multiobjectifs mixte H_2/H_∞ avec l'approche LMI et MOGA peut jouer pour assurent la stabilité robuste donc il est fournit une commande robuste vis-à-vis les perturbations et les incertitudes.

Liste des figures

Figure I.1 Représentation d'un ensemble convexe et non convexe	7
Figure I.2 Graphe d'une fonction convexe	8
Figure I.3 Problème H_∞ standard	19
Figure II.1. Escape de recherché et escape réalisable.	26
Figure II.-2 Interprétation graphique de la méthode d'agrégation par pondération	32
Figure II-3 Interprétation graphique de l'approche Min-Max	34
Figure II-4 Interprétation graphique de l'approche but à atteindre	35
Figure II.5 : Schéma de fonctionnement de VEGA	37
Figure II.6 le front de Pareto.	39
Figure II-7 Exemple de dominance et d'optimalité au sens de Pareto	40
Figure II.8 : Principe de fonctionnement de NSGAI.	44
Figure II.9: Distance de Crowding.	45
Figure II.10: Illustration du Clustering en dimension 2.	46
Figure III.1. Représentation schématique du croisement en un point.	54
Figure III.2. Représentation schématique du croisement en deux points.	54
Figure III.3. Représentation schématique de la mutation.	55
Figure III.4. Schéma de principe d'un Algorithme Génétique.	57
Figure III.5. Exemple de dominance.	58
Figure III.6 Etapes d'un algorithme génétique multi-objectifs.	60
Figure III.7. Assignation de l'efficacité d'un point de vue de dominance	62
Figure III.8 Illustration des techniques de la préservation de la diversité.	63
Figure III.9 Algorithme Génétique Multi-objectifs	65
Figure. IV-1. Problème H_∞ standard.	70
Figure IV.2 La synthèse optimale H_2	79
Figure IV.3 correcteur H_2 sous optimal.	80

Figure IV.4. Modèle standard du problème H_2/H_∞	81
Figure IV.5. Problèmes H_2/H_∞ à une entrée - deux sorties.	84
Figure IV.6. Deux masses - ressort avec paramètres incertains	86
Figure IV.7 : les réponses Impulsionnelle et la commande H_2 avec LMI	88
Figure IV.8 : les réponses Impulsionnelle et la commande H_∞ avec LMI	89
Figure IV.9 : les réponses Impulsionnelle et la commande H_2 avec MOGA	90
Figure IV.10 : les réponses Impulsionnelle et la commande H_∞ avec MOGA	91
Figure IV.11 résolution du problème H_2/H_∞ avec LMI	92
Figure IV.12 : les réponses Impulsionnelle et la commande avec LMI	93
Figure IV.13 : résolution du problème H_2/H_∞ avec MOGA	94
Figure IV.14 : les réponses Impulsionnelle et la commande avec MOGA	95
Figure IV.15 Comparaison entre les résultats de LMI et MOGA	96

Liste des tableaux

Tableau III.1. Exemple de croisement uniforme	55
Tableau III.2 : Tableau de dominance	59

Notations

$F(x)$: Inégalité matricielle linéaire (LMI) affine des éléments de x .

$x \in \mathbb{R}^n$: est un vecteur de valeurs inconnues (variables).

F_i Les matrices symétriques.

C : ensemble des points: $x \in C$ qui vérifient LMI: $F(x) > 0$.

$V(x)$: Fonction de Lyapunov.

$\dot{V}(x)$: Dérivé de la fonction de Lyapunov.

P : est une matrice symétrique et définie positive.

K : Le contrôleur.

$\bar{\sigma}(A(x))$ La valeur singulière maximale de la matrice A .

P_m : La probabilité de mutation.

P_i : population.

$\|T_{zw}(s)\|_{\infty}$: La norme H_{∞} de la fonction de transfert $T(s)$

$\|T_{zw}(s)\|_2$: La norme H_2 de la fonction de transfert $T(s)$

X et Y : matrices symétriques.

ρ : désigne le module de la plus grande valeur propre.

S_n : sous-espace vectoriel des matrices symétriques de dimension n .

S_n^{++} : Cône convexe des matrices symétriques semi définies positives de dimension n .

\mathcal{N}_X et \mathcal{N}_Y : sont des bases des noyaux des matrices $[C_2 \ D_{21}]$ et $[B_2^T \ D_{12}^T]$.

Références

- [1] E.G. Migliore « Commande Prédictive a Base de Programmation Semi Définie », Thèse de doctorat, l'Institut National des Sciences Appliquées de Toulouse, France 2004.
- [2] G. Scorletti, « Approche Unifiée de la commande des systèmes par formulation LMI » Thèse de doctorat, université de Paris Orsay 1997.
- [3] E. Laroche, « Cours Commande Optimale », Ecole Nationale Supérieure de Physique Université Louis Pasteur de Strasbourg, 5 décembre 2005.
- [4] S. L. Avilla, « Optimisation Multiobjectif et Analyse de Sensibilité Appliquées à la Conception de Dispositifs », Thèse de doctorat, l'Ecole doctorale électronique, électrotechnique, automatique de Lyon, France 2006.
- [5] K. Ben Mahammed, « Cour Commande avancée ». Université de Sétif 2008.
- [6] G. Scorletti, « Outils Pour la Robustesse : Inégalités Matricielles Affines », notes de cours Université d'Orsay SUPELEC, France 2006.
- [7] A. M. Cristóba, « Multiobjective Control : Linear Matrix Inequality Techniques and Genetic Algorithms Approach ». Thèse de Doctorat Université de Sheffield, Avril 2005.
- [8] D. Henrion, « Stabilité des Systèmes Linéaires Incertains a Commande Contrainte ». Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse 1999.
- [9] A. Kruszewski, « Lois de commande pour une classe de modèles non linéaires sous la forme Takagi-Sugeno : Mise sous forme LMI » Thèse de doctorat l'université de Valenciennes et du hainaut-cambresis 2006.
- [10] M. Chadli « Stabilité et Commande de Systèmes Décrit par des Multimodèles », Thèse de doctorat institut national polytechnique de Lorraine 2002.
- [11] V. Barichard, « Approches Hybrides Pour les Problèmes Multiobjectifs », Thèse de doctorat. Ecole Doctorale d'Angers, 24 Novembre 2003.
- [12] R. Benabid, « Optimisation Multiobjectif de la Synthèse des FACTS par les Particules en Essaim pour le Contrôle de la Stabilité de Tension des Réseaux Electriques » Mémoire de Magister en Electrotechnique. Université de Jijel 2007.
- [13] M. M. Hizem, « Recherche de Chemins dans un Graphe à Pondération Dynamique Application à l'Optimisation d'Itinéraires dans les Réseaux Routiers » Thèse de Doctorat, l'Ecole Centrale de Lille, 29 novembre 2008.

-
-
- [14] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, et E. Teller «Equations of State Calculations by Fast Computing Machines». Journal of Chemical Physics, 21: 1087-1091, 1953.
- [15] R. Kammarti « Approches Evolutionnistes Pour la Résolution du 1-PDPTW Statique et Dynamique », Thèse de doctorat, l'Ecole centrale de Lille et l'université des sciences et technologies de Lille, 2006
- [16] K. Mahdi, « L'Optimisation Multiobjectif et l'Informatique Quantique », Mémoire de Magister en informatique. Université de Mentouri Constantine.
- [17] A. Berro « Optimisation Multiobjectif et Stratégies d'Evolution en Environnement Dynamique ». Thèse de doctorat, l'université des Sciences Sociales Toulouse I, décembre 2001.
- [18] O. Guenounou, « Méthodologie de Conception de Contrôleurs Intelligents par l'Approche Génétique Application à un Bioprocédé ». Thèse de doctorat, l'Université de Toulouse avril 2009.
- [19] R. Chevrier, «Optimisation de Transport à la Demande dans des Territoires Polarisesés». Thèse de doctorat, l'Université d'Avignon et des Pays de Vaucluse. Novembre 2008.
- [20] S. Bouktif. « Amélioration de la prédiction de la qualité du logiciel par combinaison et adaptation de modèles». Thèse de Doctorat. Université de Montréal. Mai 2005.
- [21] S.Amédée et R .Francois « Algorithmes Génétiques » notes de cours.
- [22] E. Golderberg « Algorithme Génétique ». Notes de cours 1994.
- [23] K. Zidi « Système Interactif d'Aide au Déplacement Multimodal (SIADM) ». Thèse Doctorat. Université des Sciences et Technologies de Lille et l'Ecole Centrale de Lille, décembre 2006.
- [24] R. Chevrier, « Optimisation de Transport à la Demande dans des Territoires Polarisesés ». Thèse de doctorat, l'Université d'Avignon et des Pays de Vaucluse. novembre 2008.
- [25] K. Deb, A. Pratap, S. Agarwal, et T. Meyarivan, « A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II », IEEE Trans, on Evolutionary Computation, April 2002.
- [26] M. K. GHALI « Méthodologie de Conception Système a Base de Plateformes Reconfigurables et Programmables». Thèse de doctorat, l'université Paris XI Orsay, 2005.
- [27] N. Jozefowies, « Modilisation et Resolution Approchée de Probleme de Tournées Multiobjectif ». Thèse de doctorat, l'université des Sciences et Technologies de Lille, 2004.
- [28] F. Picarougne « Recherche d'information sur Internet par algorithmes évolutionnaires ». Université François Rabelais Tours École Doctorale : Santé, Sciences et Technologies 2004.

-
-
- [29] A.A.Benlatreche, « Contribution à la Commande Robuste H_∞ de Systèmes à Grande Echelle d'Enroulement de Bandes Flexibles ». Thèse de Doctorat. Université Louis Pasteur Strasbourg I, 2007.
- [30] P. Apkarian « Eléments de la Théorie de la Commande Robuste », livre. France.
- [31] D. Arzelier, « Theorie de Lyapunov Commande Robuste et Optimisation » Habilitation à diriger des recherches.
- [32] C. POUSSOT-VASSAL, « Commande Robuste LPV Multivariable de Châssis Automobile ». Thèse de Doctorat. INP Grenoble 2008.
- [33] D. Peaucelle & D. Arzelier « Robust Multi-Objective Control ». Version 1. Boîte à outils Fondements théoriques de RoMulOC et son utilisation illustrée sur des exemples. LAAS-CNRS. Toulouse, France.
- [34] D. Peaucelle « Formulation Generique de Problemes En Analyse et Commande Robuste Par Les Fonctions De Lyapunov Dependant Des Parametres » l'Ecole Centrale de Lille.
- [35] D. Arzelier, « Analyse et Synthèse Robustes Des Systèmes Linéaires Synthèse LQG/LTR et H_2 ». Cours LAAS-CNRS. Toulouse, France.