

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



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

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Sciences et Technologies**

Filière : **Génie électrique**

Spécialité : **Commande des systèmes**

Présenté par

Fariza MERDJANE

Lamia MEFTOUH

Thème

Synthèse d'un Correcteur Robuste basée sur l'Approche des Scénarios

Mémoire soutenu publiquement le 29/09/2016 devant le jury composé de :

M Kamel HAMMOUCHE

PROF, UMMTO, Président

M Ahmed MAIDI

PROF, UMMTO, Encadreur

M Said DJENNOUNE

PROF, UMMTO, Examineur

M Mohand Achour TOUAT

MCB, UMMTO, Examineur

Remerciement

A l'issue de ce travail qu'on a mené pour conclure nos études du master, nous tenons à exprimer des remerciements, bien mérités, pour quelques personnes qui ont contribué à la réalisation de ce projet de fin d'études, mais bien avant il ne sera qu'évident de remercier le Dieu tout-puissant de nous avoir donné le courage pour accomplir ce travail, et aussi de remercier nos parents de nous avoir ouvert les portes de réussite.

*Nos vifs remerciements seront aussi adressés à notre respectueux professeur, **M. Ahmed MAIDI**, d'avoir accepté de nous encadrer pour notre projet de fin d'études, et de nous avoir proposé ce thème qui nous a ouvert les portes à de tels domaines dont on souhaite continuer nos recherches, ainsi que pour son soutien, ses remarques pertinentes et son encouragement, mais surtout pour sa patience avec nous.*

*Nous tenons aussi à remercier tous les membres de jury **M. Said DJENNOUNE**, **M. Mohand Achour TOUAT** et à leur tête **Le Président des Jury M. Kamel HAMMOUCHE** de nous avoir honorés en acceptant de juger notre modeste travail. Veuillez trouver ici le témoignage de nos respects les plus profonds.*

Nos remerciements vont aussi, chaleureusement, à tous nos professeurs, enseignants, tous les membres de la bibliothèque FGEI et toutes les personnes qui nous ont soutenu jusqu'au bout du près et/ou de loin, à ceux qui ont cru en nous, et qui n'ont pas cessé de nous encourager et de nous donner des conseils très importants. Nous vous exprimons nos reconnaissances les plus distinguées.

Dédicaces

Je dédie ce modeste travail :

*A mes chers parents que j'aime beaucoup, et c'est grâce à eux que je suis devenue celle
d'aujourd'hui.*

A mes frères Hamid et Mohammed, ainsi qu'à ma sœur Lynda.

A ma sœur Zina et son époux.

A mes petites nièces : Mélissa, Létitia et Katia, et à mon petit neveu : Ahmed.

A mon cher binôme et sa famille.

*A tous mes amis(es) de la promotion, et plus précisément à Dyhia et Nawel; à qui je leurs
souhaite beaucoup de réussites.*

Fariza.

Dédicace

Je dédier ce modeste travail :

A mes très chers parents, sans vous j'y suis pour rien;

*A mon unique sœur Kahina que j'adore et à mes chers frères Aissam et sa femme Sabrina,
Mehdi, Abdeslam et surtout à mon petit frère Bélaid ;*

A mon neveu, le chouchou de la famille, Enzo qui me manque beaucoup et sa mère ;

Mes oncles, mes tantes, mes voisins,

Ainsi qu'à tous mes cousin(e)s chacun avec son prénom et en particulier Amayas, Lina.

A mon amie et mon binôme Fariza et toute sa chère famille.

A mes amis et amies : Samia, Dyhia, Nawal, Dalila, Younes, Meziane,

*A tous mes profs depuis le primaire en particulier mes profs de Maths et tous ceux qui ont
encore la conscience et enseignent avec un grand cœur.*

*A la mémoire de ma Grand-Mère qui m'a accueilli tout au long de mes études universitaires
« Que Dieu t'accueillera en son vaste paradis »*

*A tous ceux qui m'aiment et tous ceux que j'aime et toutes personne qui m'a marqué et que
j'ai marqué.*

Que Dieu vous protège pour moi et pour tous vos chers. Je vous aime.

Lamia

Achronymes

S.I.S.O : (Single Input – Single Output)

M.I.M.O : (Multi Input – Multi Output)

S.I.M.O : (Single Input – Multi Output)

M.I.S.O : (Multi Input – Single Output)

BO : Boucle ouverte.

BF : Boucle fermée.

LFR : Représentation linéaire fractionnaire.

LPV : Linéaire à paramètre variant.

v. a : Variable aléatoire.

iid : Indépendant identiquement distribué.

AGs : Algorithme génétique.

SP_N : Problème d'optimisation de scénario.

MC : Monte-Carlo.

LCG : générateur congruentiel linéaire.

ARMA : Autorégressive à moyenne mobile (Auto-Regressive Moving-Average).

SIP: Problème semi infinie.

NOTATION

$u(t)$: Signal de commande.

$p(t)$: Signal de Perturbation.

$y(t)$: Signale de sortie du système (de mesure).

t : Variable du temps.

$G(s)$: Fonction de transfert.

$L(s)$: Non linéarité.

s : Variable de Laplace.

$x(t)$: Variable d'état (décision).

$\dot{x}(t)$: Dérivé de la variable d'état.

$y^d(t)$: Consigne qui représente la valeur désirée.

$\varepsilon(t)$: Ecart entre la consigne (valeur désirée) et la valeur mesurée.

$R_{s_{min}}$: Valeur minimal de la résistance (potentiomètre).

$R_{s_{max}}$: Valeur maximal de la résistance (potentiomètre).

ζ_i : Les paramètres incertains.

Δ_i : Matrice d'incertitude.

$K(s)$: Le correcteur.

$G_a(s)$: Le modèle nominal

$M(s)$: Modèle augmenté.

$W_i(s)$: Le compensateur (les filtres).

$S(s)$: Fonction de sensibilité.

$T(s)$: Fonction de sensibilité complémentaire.

b : Bruits.

w : Entrées exogènes: consignes, perturbations

z : Sorties à surveiller

$\|\cdot\|_\infty$: La norme H_∞

$\|\cdot\|_2$: La norme H_2

l_2 : La norme l_2 (euclidienne).

γ : Indice de performance.

ω_b : Pulsation des bases fréquences.

ω_h : Pulsation des hautes fréquences.

n : Nombre de variable d'état.

Ω : Ensemble admissible de variable de décision.

$g_i(x)$: Les contraintes de type égalité.

$h_j(x)$: Les contrainte de type inégalité.

x_i : Variable aléatoire.

N : Nombre de variable aléatoire générés.

ϵ : Paramètre de risque.

β : Paramètre de confiance.

a_m : Multiplicateur

c_i : Incrément.

m_t : Terme.

u_η : Suite de variable aléatoire.

d : Retard.

$E[y^2]$: Variance asymptotique.

Liste des figures

Figure 1.1. Schéma représentatif de la notion de système	3
Figure 1.2. Schéma représentatif de la notion de système selon le type d'entrée.....	4
Figure 1.3 Nature mathématique des systèmes	7
Figure 1.4. Schéma de principe d'un système électrique.....	8
Figure 1.5. Réponse indicielle d'un second ordre avec amortissement.....	8
Figure 1.6. Schéma d'un bloc.....	9
Figure 1.7. Schéma fonctionnel d'un système asservi.	9
Figure 1.8. Schéma bloc d'un élément non linéaire.....	9
Figure 1.9. Schéma représentatif d'un système de commande.....	14
Figure 1.10. Représentation par schéma-bloc d'un procédé en boucle ouverte.....	15
Figure 1.11. Structure générale d'un Système en boucle fermée.....	16
Figure 1.12. Schéma représentant les deux fonctions d'un correcteur.	17
Figure 2.1. Représentation des incertitudes paramétrique.....	22
Figure 2.2. Représentation générale des incertitudes non structurées.	23
Figure 2.3. Schéma d'un système électromécanique	24
Figure 2.4. Schéma d'analyse de la robustesse de la stabilité.	28
Figure 2.5. Système soumis aux pondérations.	29
Figure 2.6. La transformation du problème sous une forme standard.	30
Figure 2.7. Forme standard du problème H_{∞}	31
Figure 2.8. Synthèse par loop-shaping.	34
Figure 3.1. Illustration du type d'optimum.	39
Figure 3.2. Organigramme des algorithmes génétiques.....	44
Figure 3.3. Le croisement en codage binaire.	47
Figure 4.1. Modèle de structure AR.....	60
Figure 4.2. Modèle de structure MA.	60
Figure 4.3. Modèle de structure ARMA.	60
Figure 4.4. Schéma de système de commande	62
Figure 4.5. Robustesse de correcteur	67
Figure 4.6. Représentation de la sortie du système avec le correcteur robuste, et avec le correcteur nominal.....	68

Figure 4.7. Évolution de la sortie du système	69
Figure A.1 schéma fondamental d'un système à rétroaction (annexe)	72
Figure A.2 structure LFR d'un système incertain.....	74

Liste des tableaux

Tableau 2.1 Représentation des incertitudes non paramétrique.....	24
Tableau 3.1 Création de la population initiale.	49
Tableau 3.2. Calcul de la fonction d'adaptation.	49
Tableau 3.3. Sélection des individus	50
Tableau 3.4. Croissement des individus.	50
Tableau 3.5. Reproduction d'une nouvelle génération.	50

Table des matières

Introduction Générale.....	1
-----------------------------------	----------

Chapitre01 : Généralités sur les systèmes

I. Introduction	3
II. Généralité sur les systèmes	3
II.1. Notion de système	3
II.2. Classification des systèmes	4
II.2.1. Selon la gestion du temps.....	4
II.2.1.1. Système à temps continu	4
II.2.1.2. . Système à temps discret.....	4
II.2.2. Selon la nature des paramètres	5
II.2.2.1. Système stationnaire ou invariant	5
II.2.2.2. Système à paramètres variant.....	5
II.2.2.2.1. Système à paramètre localisé.....	5
II.2.2.2.1. Système à paramètre distribués	5
II.2.3. Selon le comportement du système	5
II.2.3.1. Système déterministe	5
II.2.3.2. Système stochastique (aléatoire/non déterministe)	5
II.2.4. Système linéaire/Non linéaire.....	6
II.2.4.1. Système linéaire.....	6
II.2.4.2. Système non linéaire	6
II.2.5. Selon la nature des signaux	6
II.2.5.1. Système statique – système dynamique	6
II.2.5.2. Système causal	6
II.2.6. Selon le nombre d'entrées et de sorties	6
II.3. Représentation des systèmes	7
II.3.1. Représentation schématisée	7
II.3.1.1. Schéma du système physique ou schéma de principe	8
II.3.1.2. Schéma graphique	8
II.3.1.3. Schéma fonctionnel(en anglais block diagram)	9
II.3.2. Représentation mathématique	9
II.3.2.1. Obtention du modèle	10
II.3.2.2. Type du modèle	10
II.3.2.3. Quelques formes générales des modèles obtenus	10

II.3.2.3.1. Représentation externe	10
II.3.2.3. 2. Représentation interne	12
III. Généralité sur les systèmes de commande	14
III.1. Notion de système commande	14
III.2. Propriétés d'un système de contrôle	15
III. 3. Type de commande	15
III.3. 1. Commande en boucle ouverte	15
III.3. 2. Commande en boucle fermée	16
III.4. Asservissement et régulation	18
III.5. Commande classique	18
III.6. Commande moderne	19
IV. CONCLUSION.....	19

Chapitre 02 : Généralité sur la commande Robuste.

I. Introduction	20
II. Notion de système incertain	20
III. notion de l'incertitude.....	21
III. 1. Sources des incertitudes	21
III.1. 1. Perturbations externes négligé	21
III.1. 2. Non-linéarités négligées	21
III.1. 3. Dynamique non-modélisée	22
III.1. 4. Variations des paramètres réels	22
III. 2. Classification et représentations des incertitudes	22
III.2. 1. Incertitudes structurées ou incertitudes paramétriques	22
III.2. 2. Incertitudes non structurées ou incertitudes dynamique	23
IV. Notion de robustesse	25
V. Commande robuste	26
V. 1. Classification des méthodes (commandes) robustes	26
V.1. 1. Méthodes déterministes	27
V.1.1. 1. Quelques méthodes déterministes	27
V.1.2. Méthodes probabilistes	35
V. 2. Inconvénient des commandes robustes	35
V. 3. Notion de complexité algorithmique	35
V.3.1. Théorie de la complexité algorithmique	35
V.3.2. Classe de complexité	35
V.3.3. Classification des problèmes	36
V.3.3.1. Classe P (Polynomial time)	36

V.3.3.2. Classe NP (Non deterministic Polynomial time)	36
V.3.3.3. Classe NP-complet	36
V.3.3.4. Classe NP-difficile ou NP-dur	36
VI. Conclusion	37

Chapitre 03 : Optimisation Robuste par l'Approche des Scénarios.

I. Introduction	38
II. Généralité sur l'optimisation	38
II.1. Optimisation	38
II.2. A quoi sert l'optimisation ?	38
II.3. Mise en forme mathématique d'un problème d'optimisation	38
II.4. Quelques définitions	39
II.4.1. Notion d'optimum	39
II.4.2. Qu'est-ce qu'une contrainte ?	40
II.5. Formulation mathématique d'un problème d'optimisation	40
II.6. Comment classifier un problème d'optimisation ?	41
III. Résolution d'un problème d'optimisation	42
III.1. Méthodes globales et méthodes locales	42
III.2. Méthodes déterministes	42
III.3. Méthodes stochastiques	43
III.3.1. Méthodes Métaheuristiques	43
IV. Algorithmes Génétiques	44
IV.1. Qu'est-ce qu'un algorithme génétique ?	44
IV.2. Principe des AGs	44
IV.3. Fonctionnement des AGs	45
IV.3.1. Population initial	45
IV.3.2. Codage et population initiale1	45
IV.3.3. Fonction d'évaluation	46
IV.3.4. Operateurs des algorithmes génétiques	46
IV.3.4.1. Opérateur de sélection	46
IV.3.4.2. Opérateur de Croisement	46
IV.3.4.3. Operateur de mutation	47
IV.3.5. Condition d'arrêt des AGs	47
IV.3.6. Création d'une nouvelle population	47
IV.4. Avantage et inconvénient	48
V. Optimisation en présence d'incertitude	51
V.1. Optimisation Robuste	51

V.2. Qu'est-ce- qu'une optimisation semi-infinie ?	52
VI. Approche des scénarios	53
VI.1. Fondement de la méthode	53
VI.2. Procédure d'optimisation des scénarios	54
VII. Génération des nombre aléatoire	55
VII.1. Nombre aléatoire.....	55
VII.2. Propriétés des nombres aléatoires	56
VII.3. Méthode de Monte Carlo	56
VII.3.1. Principe de la méthode de MC	56
VII.3.2. Concept de la méthode de MC	56
VII.4. Générateur nombre pseudo- aléatoire	57
VII.4.1. Générateur congruentiel linéaire (LCG)	57
VII.4.1.1. Critère de Knuth	58
VII.4.2. Avantage de l'approche pseudo- aléatoire	59
VIII. Conclusion.....	59

Chapitre 04 : Synthèse d'un correcteur robuste par l'Approche des Scénarios

I. Introduction	60
II. Structure <i>ARMA</i> (Auto-Regressive Moving-Average)	60
III. Application de l'Approche des Scénarios pour la synthèse d'un correcteur robuste pour un système <i>ARMA</i>	62
III.1. synthèse du correcteur nominal	63
III.2. synthèse d'un correcteur robuste	64
IV. Conclusion	69

Conclusion générale 70

Annexe 71

Références bibliographique..... 76

Introduction Générale

Introduction Générale

L'automatique est une branche interdisciplinaire de l'ingénierie et des mathématiques qui s'intéresse au comportement des systèmes dynamiques. L'un des principaux buts de l'automaticien est de concevoir un système, appelé correcteur, capable de commander un système physique (également appelé processus physique) donné, autrement dit, de l'influencer de façon à ce qu'il se comporte d'une certaine façon. L'intérêt est d'obtenir des processus autorégulés requérant un minimum d'intervention humaine pour fonctionner : par exemple, le régulateur de vitesse d'une voiture a pour rôle de maintenir le véhicule à une vitesse donnée malgré des perturbations comme l'état de la route ou de la circulation.

En pratique, l'automaticien est confronté à deux problèmes majeurs:

- le problème d'évaluer si un correcteur assure au système le comportement voulu : c'est le problème d'analyse ;
- le problème de concevoir un correcteur assurant au système le comportement voulu : c'est le problème de synthèse.

La synthèse d'une loi de commande se fait généralement sur un modèle nominal simplifié, qui ne prend pas en compte toute la complexité du système (les dynamiques négligées et les variations des paramètres dues par exemple aux changements des conditions de fonctionnement comme la température) ce qui provoque un écart, connu par le terme d'*incertitude*, entre le comportement du système réel et celui du modèle.

Au cours des dernières décennies, une grande partie des travaux s'est focalisée sur la modélisation d'« incertitudes », les analyser et établir des synthèses. La commande robuste, apparue depuis les années 1980, offre une large gamme de méthodes fondées sur une base mathématique solide destinées, essentiellement, à garantir au système la robustesse souhaitée (assurer la stabilité et un certain niveau de performance) même lorsque ce dernier ; en lui-même n'est pas connue avec précision (incertain). De plus, la majorité des problèmes de commande robustes sont formulés sous forme d'un problème d'optimisation, alors les deux communautés (automaticiens et mathématiciens) se sont rassemblées et l'optimisation robuste fut apparaitre [11].

Néanmoins, l'application des méthodes développées pour la synthèse d'une commande robuste aux problèmes pratiques de grandes dimensions a révélé que la complexité de calcul de ces méthodes croît en fonction de la taille du problème, et il a été établi que plusieurs problèmes d'intérêt pratique sont NP-difficile. Ce constat a poussé les chercheurs à développer des méthodes alternatives. Ainsi, pour prendre en considération les incertitudes, des méthodes dites probabilistes ont été proposées. Le principe de ces dernières consiste à supposer que la distribution de ces incertitudes obéit à une certaine loi probabiliste, puis de concevoir des correcteurs assurant les performances désirées avec une certaine probabilité de défaillance.

L'objectif de notre travail est de présenter et d'appliquer une des méthodes de l'approche probabiliste, la plus utilisée, appelée approche des scénarios.

L'essentiel du travail réalisé est structuré en quatre chapitres. Ainsi,

Le **chapitre 1** porte sur des généralités sur les systèmes et les systèmes de commande.

Dans le **chapitre 2** on s'intéresse aux questions d'incertitudes et de robustesse. Le chapitre présente aussi les différentes approches de synthèse d'un système de commande robuste.

Le **chapitre 3** est consacré aux méthodes d'optimisation. Dans ce chapitre on montre qu'un problème de commande robuste se ramène toujours à un problème d'optimisation robuste qu'on peut relaxer avec l'approche de scénarios.

Dans le **chapitre 4** la synthèse d'un correcteur robuste par l'approche des scénarios est illustrée en considérant un exemple.

Le mémoire se termine par une conclusion générale.

Chapitre 01

Généralités sur les systèmes

I. Introduction :

Ce chapitre présente des généralités sur les systèmes où on abordera la définition d'un système, ses classifications et les différentes représentations qui pourront le décrire. Enfin, on se focalisera sur des notions de base des systèmes de commande.

II. Généralité sur les systèmes :

II.1. Notion de système :

En général, un **système** désigne un ensemble d'éléments interagissant entre eux naturellement ou artificiellement et d'une façon intelligente selon certains principes ou règles. Par exemple : le système nerveux, économique, politique, ... etc.

En automatique, le mot système est une notion clé. Il peut être défini comme suit :

- D'un point de vue réel concerne le système physique, concret. C'est un ensemble d'objets ou d'entités connectées et interagissant les uns avec les autres pour accomplir une tâche prédéfinie. Chaque composant du système peut être lui-même un sous-système ce qui conduit à des systèmes plus ou moins hiérarchisés et donc plus ou moins complexe.
- D'un point de vu abstrait, le système est assimilé à une maquette virtuelle, matérialisé par des symboles et des équations, qui a pour qualité de représenter et de décrire aux mieux le comportement et l'évolution du système dit réel. C'est le modèle.

En général un système est représenté en multi variables par le schéma de la figure 1.1

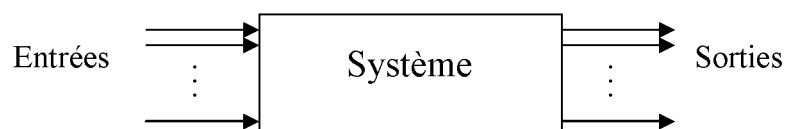


Figure 1.1. Schéma représentatif de la notion de système

L'influence du milieu extérieure au système dû à l'échange entre système et son environnement impose certaine décomposition des entrées exogène comme suit :

- **La commande** : Notée $u(t)$. Entrée exerçant une action entraînant le fonctionnement souhaité du système. C'est une entrée commandable.
- **La perturbation** : Notée $p(t)$. Entrée non désirée, exerçant une action qui trouble le fonctionnement désiré du système. C'est une entrée non commandable.

Le système, comme effet à ces entrées, émet une contre action connu comme sortie ou réponse du système et caractérisé par un signal de sortie.

- **Sortie** : Notée $y(t)$. On appelle sortie la réponse effective obtenue à partir du système. C'est cette variable $y(t)$ que l'on veut maîtriser.

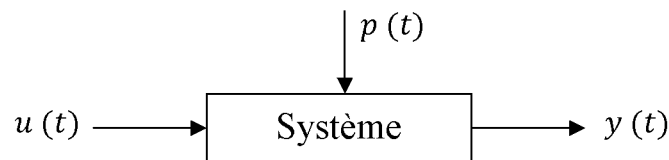


Figure 1.2. Schéma représentatif de la notion de système selon le type d'entrée.

II.2. Classification des systèmes :

La classification d'un système dépend de sa nature physique mais aussi des hypothèses simplificatrices qu'il est possible de faire sur son modèle. Pour cela les systèmes peuvent être classés en plusieurs catégories comme suit [10, 19]:

II.2.1. Selon la gestion du temps.

On trouve les Systèmes à temps continu et à temps discret.

II.2.1.1. Systèmes à temps continu :

Ce sont les systèmes qui existent naturellement. Pour ces systèmes, les signaux d'entrées et de sorties sont définis à chaque instant de t .

II.2.1.2. Systèmes à temps discret :

On parle de système discret ou échantillonné, lorsque tous les signaux d'entrées et de sorties sont définis périodiquement dans le temps t , ou le signal n'est transmis qu'à des instants discrets privilégiés.

On parle aussi de système continu-discret quand on veut piloter (commander) un processus par ordinateur. On considère alors un bloqueur et un échantillonneur.

II.2.2. Selon la nature des paramètres.

On a les systèmes stationnaires et les systèmes variant.

II.2.2.1. Système stationnaire ou invariant :

Ce sont des systèmes dont les paramètres du système ne varient pas au cours du temps.

II.2.2.2. Système à paramètres variant :

Il se décompose en système à paramètres localisé et système à paramètres distribués.

II.2.2.2.1. Système à paramètre localisé :

C'est un système dynamique où les paramètres sont en fonction du temps t , ils sont généralement décrits par des équations différentielles ordinaires.

II.2.2.2.2. Système à paramètre distribués :

Le système est dit distribué ou a paramètre répartis lorsque ses paramètres évoluent dans le temps et l'espace. Ils sont gouvernés par des équations aux dérivées partielles.

II.2.3. Selon le comportement du système.

On distingue le système déterministe et le système stochastique (aléatoire/probabiliste).

II.2.3.1. Système déterministe :

Un système déterministe est un système dont le comportement est parfaitement prévisible. C'est-à-dire en connaissant le signal d'entrée, la sortie peut être calculée (prédite) exactement à partir du modèle du système et pour chaque entrée $u(t)$, il n'existe qu'une seule sortie possible $y(t)$.

II.2.3.2. Système stochastique (aléatoire/non déterministe) :

Un système stochastique est un système à comportement aléatoire. Certains paramètres (dynamique, donnée initiales, perturbations) dépendent de variables aléatoires et

font que le signal de sortie ne peut pas être déterminé exactement. Il y a plusieurs sorties possibles, chacune d'elles étant affectée d'une certaine probabilité.

II.2.4. Système linéaire/non linéaire

II.2.4.1. Système linéaire :

Un système est dit **linéaire** au sens mathématique si le principe de superposition (voir annexe) est vérifié. Son comportement dynamique est régi par un système d'équations différentielles linéaires.

II.2.4.2. Système non linéaire :

Un système est non linéaire si son signal de sortie est une fonction non linéaire par rapport au signal d'entrée, donc le système ne vérifie pas le théorème de superposition.

II.2.5. Selon la nature des signaux.

Les signaux causaux, numériques, logiques, analogiques, déterministes ou stochastiques.

II.2.5.1. Système statique – système dynamique :

Un système statique est un système dont la réponse à un signal d'entrée est instantanée. Sinon il est dit dynamique c'est-à-dire la réponse dépend du signal d'entrée et des valeurs précédentes passées (il est à mémoire).

II.2.5.2. Système causal :

Un système est dit causal s'il n'émet pas de réponse tant qu'il n'y a pas d'entrée ; notons que tous les systèmes physiques sont causaux.

II.2.6. Selon le nombre d'entrées et de sorties.

Dans ce cas, on distingue :

- **SISO** (Single Input – Single Output) : il s'agit de systèmes à une entrée et à une sortie, appelés systèmes mono-variables.
- **MIMO** (Multi Input – Multi Output) : il s'agit de systèmes à plusieurs entrées et à plusieurs sorties appelés systèmes multi-variables.

- **SIMO** (Single Input – Multi Output) : il s'agit de systèmes à une entrée et à plusieurs sorties.
- **MISO** (Multi Input – Single Output): il s'agit de systèmes à plusieurs entrées et à une seule sortie.

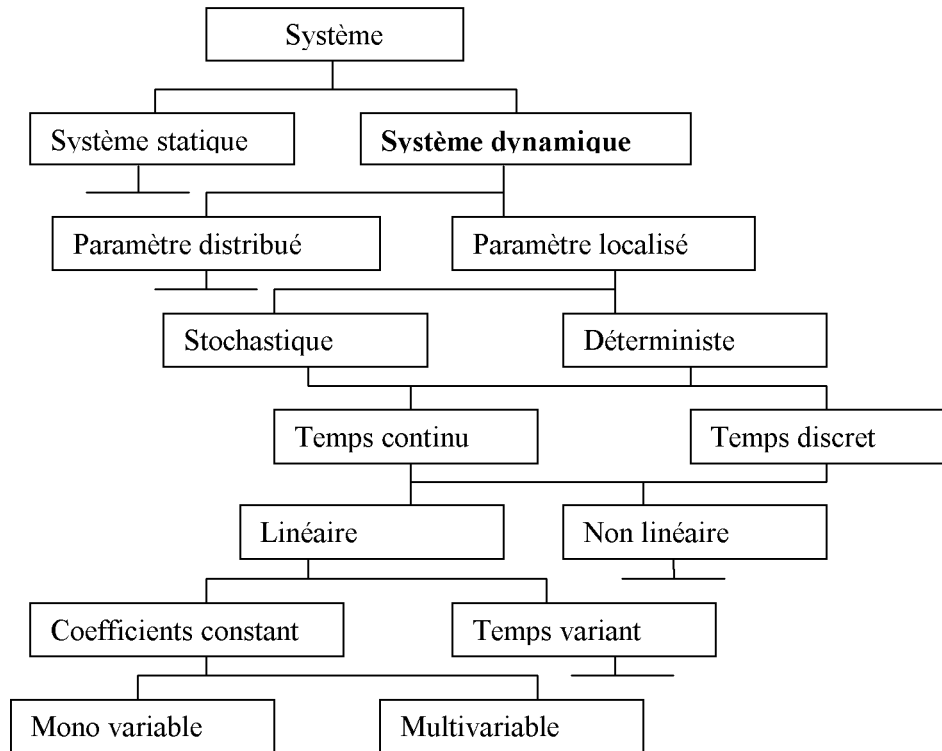


Figure 1.3 Classification des systèmes.

II.3. Représentation des systèmes :

Pour étudier un système et le commander, généralement un système est représenté schématiquement (symbole, graphe, schéma, tableau) ou algébriquement (modèle mathématique : fonctions, équations,...), et cela, dépend du besoin, de niveau d'étude, et de la quantité d'information et de données requises sur ce système.

II.3.1. Représentation schématisée :

Généralement on distingue les représentations suivantes :

II.3.1.1. Schéma du système physique ou schéma de principe:

Ensemble de symboles reliés entre eux, représentant les entités physiques constituant le système (composants matériels et/ou soft) et donnant une description globale sur le système (son domaine, son architecture et son principe de fonctionnement). A titre d'exemples, le schéma électrique suivant :

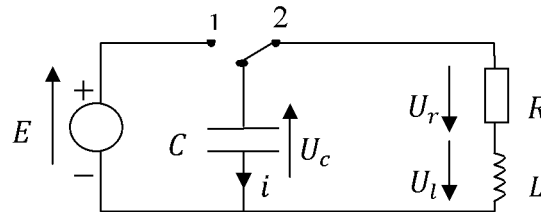


Figure 1.4. Schéma de principe d'un système électrique.

C : condensateur, R : résistance, L : inductance, E : Ggénérateur de tension, i : intensité du courant électrique.

U_c, U_l et U_r . Respectivement les tensions aux bornes du condensateur, de l'inductance et de la résistance.

II.3.1.2. Schéma graphique :

En général, cette représentation est dédiée à la représentation des caractéristiques des signaux d'entrées et de sorties (réponse du système à ces entrées) issus de l'expérimentation.

Prenons la réponse indicielle comme exemple explicatif :

Entrée échelon.

Sortie (réponse) indicielle

$$\begin{cases} u(t) = 0 \text{ pour } t > 0 \\ u(t) = 1 \text{ pour } t < 0 \end{cases} \longrightarrow \begin{cases} H(p) = \frac{k}{\frac{1}{\omega_0^2}p^2 + \frac{2\varepsilon}{\omega_0}p + 1} \end{cases} \quad (1.2)$$

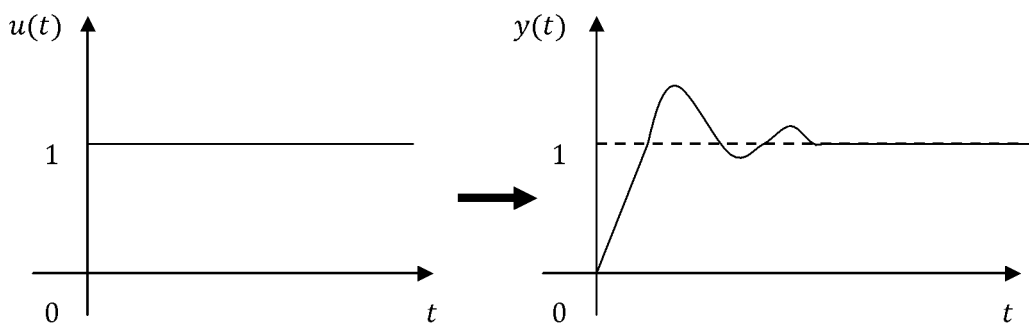


Figure 1.5. Réponse indicielle d'un second ordre avec amortissement.

II.3.1.3. Schéma fonctionnel(en anglais block diagram) :

Un schéma fonctionnel ou encore Diagramme/Bloc fonctionnel est une représentation graphique simplifiée d'un processus relativement complexe, mettant en évidence les différentes fonctions mise en œuvre entre les organes d'un système de commande. Il est composé de *blocs* connectés par des *lignes d'action*.

Le bloc ou élément ou encore sous-système est représenté par un rectangle avec l'action de l'élément (par exemple : $d/dt, G(s), H(s), \dots$). Il est parfois accompagné d'une description (par ex. dérivateur, intégrateur...) et du symbole du *signal d'entrée* (ou *variable de commande*) et du *signal de sortie* (ou *variable commandée*).

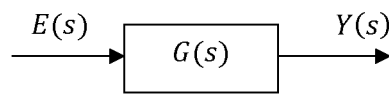


Figure 1.6. Schéma d'un bloc.

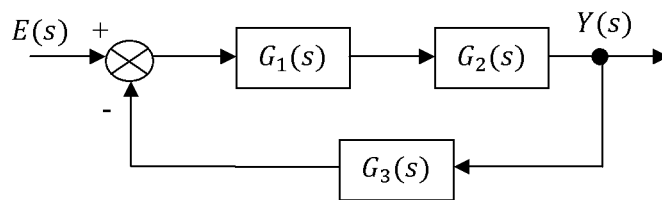


Figure 1.7. Schéma fonctionnel d'un système asservi.

Remarque 1.1

Un élément non linéaire est représenté par un rectangle double encadré.

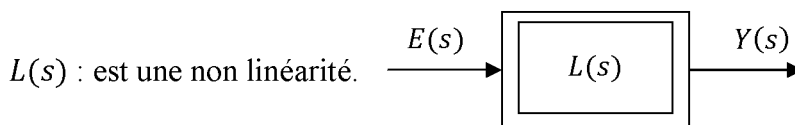


Figure 1.8. Schéma bloc d'un élément non linéaire.

II.3.2. Représentation mathématique :

Cette représentation est donnée par des équations mathématiques, mettant en œuvre les différentes relations du système. Il s'agit du **modèle**. La qualité d'un modèle est jugée par la simplicité qu'il offre tout en gardant sa fidélité au système réel (approchant au mieux le système réel), d'où vient la difficulté de sa détermination.

II.3.2.1. Obtention du modèle :

Il existe deux approches permettant d'obtenir un modèle mathématique.

- **La modélisation :** écriture des lois de la physique régissant le système réel, Ce qui conduit à des équations mathématiques.
- **L'identification :** procédure expérimentale qui s'appuie sur l'application d'entrées bien spécifiées et observation des sorties correspondantes, autrement dit, utilisation des mesures d'entrées-sorties.

II.3.2.2. Type du modèle :

On distingue trois principaux types de modèle obtenus par la modélisation et l'identification [10] :

- **Modèle de connaissance :**

Ce type de modèle est aussi dit *boite blanche* et donne une *représentation interne* au système. Il est élaboré en utilisant la modélisation et ses paramètres ont un sens concret.

- **Modèle de comportement ou de représentation :**

Ce type de modèle est dit *boite noire* ou *représentation externe* ou encore *modèle cause à effet*. Il est obtenu par identification vis-à-vis d'un type d'entrée donné. Les paramètres n'ont pas un sens physique concret.

- **Modèle intermédiaire :**

Ce type de modèle est aussi dit *boite grise*. Il est situé entre les deux modèles précédents ; la forme de modèle est obtenue par modélisation, tandis que les paramètres sont obtenus par identification. Les paramètres peuvent avoir un sens concret.

II.3.2.3. Quelques formes générales des modèles obtenus :

II.3.2.3.1. Représentation externe :

Le mot externe vient du fait que le système est décrit d'un point de vue extérieur : relation entrées-sorties. Et on peut différencier les cas suivants :

1. Dans le domaine temporel :

Le système est représenté soit par une **équation différentielle** pour les systèmes continus, soit par une **équation aux différences** pour les systèmes en temps discret.

Citons à titre d'exemple la forme générale d'une équation différentielle ordinaire décrivant le système en temps continu :

$$\begin{aligned} a_n \frac{dy^n(t)}{dt^n} + a_{n-1} \frac{dy^{n-1}(t)}{dt^{n-1}} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) \\ = b_m \frac{du^m(t)}{dt^m} + b_{m-1} \frac{du^{m-1}(t)}{dt^{m-1}} + \dots + b_1 \frac{du(t)}{dt} + b_0 u(t) \end{aligned} \quad (1.3)$$

Remarque 1.2 : Pour le système réel causal $n \geq m$.

2. Dans le domaine fréquentiel :

Le système est représenté soit par la **Fonction de transfert** (pour les systèmes SISO) ou par la **matrice de fonctions de transfert** (pour les systèmes MIMO) selon la dimension du système. En continu les fonctions sont exprimées dans le domaine de Laplace et en discret dans le domaine de la transformée en Z.

La **Fonction de transfert** noté $G(s)$ donne la relation entre l'entrée et la sortie du système dans le domaine de Laplace (fréquentiel).

Elle est donnée par :

$$G(s) = \frac{Y(s)}{U(s)} \quad (1.4)$$

Et peut être obtenue par une transformation de l'équation différentielle par la transformée de Laplace (\mathcal{L}) dans le cas continu ou de la transformation d'équation aux différences par la transformée en Z pour le cas discret.

2.1. Forme générale d'une fonction de transfert :

Rappelons que pour des conditions initiales nulles ('s' la variante de Laplace) on a

$$\mathcal{L}\{u(t)\} = U(s) \quad (1.5)$$

$$\mathcal{L}\{y(t)\} = Y(s) \quad (1.6)$$

$$\mathcal{L}\left\{\frac{dp^n(t)}{dt^n}\right\} = s^n P(s) \quad (1.7)$$

Donc, en appliquant la transformée de Laplace pour l'équation (1.3), il en résulte :

$$\begin{aligned} a_n s^n Y(s) + a_{n-1} s^{n-1} Y(s) + \dots + a_0 Y(s) \\ = b_m s^m U(s) + b_{m-1} s^{m-1} U(s) + \dots + b_0 U(s) \end{aligned} \quad (1.8)$$

$$\begin{aligned} (a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s^1 + a_0) Y(s) \\ = (b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s^1 + b_0) U(s) \end{aligned} \quad (1.9)$$

Finalement :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0} \quad (1.10)$$

II.3.2.3. 2. Représentation interne :

Ici, le système est décrit par un modèle, dit modèle d'état, exprimant les interactions entre les états du système, ce qui donne au système une description interne.

La forme générale de la représentation d'état est donnée comme suit :

$$\dot{x} = f(x(t), u(t), t) \quad (1.17)$$

$$y = h(x(t), u(t), t) \quad (1.18)$$

La structure générale d'un modèle de représentation d'état pour les systèmes linéaires est :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (1.19)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (1.20)$$

Où : (1.19) nommé équation de commande

(1.20) nommé équation d'observation

$A \in R^{n \times n}$ Matrice de transition ou matrice d'état.

$B \in R^{n \times m}$ Matrice de commande.

$C \in R^{r \times n}$ Matrice d'observation.

$D \in R^{r \times m}$ Matrice de transmission directe.

$u(t)$ vecteur d'entrée de commande.

$x(t)$ vecteur d'état.

$y(t)$ vecteur de sortie.

Notons que généralement $D(t) = 0$. (1.21)

Pour le cas d'un système linéaire temps invariant (LTI), les équations (1.19) et (1.20) deviennent :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.22)$$

$$y(t) = Cx(t) + Du(t) \quad (1.23)$$

La fonction de transfert peut être obtenue à partir de la représentation d'état par la relation suivante :

$$G(s) = C(sI - A)^{-1}B + D \quad (1.11)$$

Comme suit :

Soit la représentation d'état suivante d'un système LTI.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.12)$$

$$y(t) = Cx(t) + Du(t) \quad (1.13)$$

En appliquant la transformé de Laplace pour (1.12) et(1.13), et en considérant les conditions initiales nulles, on aura :

$$sX(s) = AX(s) + BU(s) \quad (1.14)$$

$$Y(s) = CX(s) + DU(s) \quad (1.15)$$

$$(1.14) \Rightarrow X(s)(sI - A) = BU(s) \Leftrightarrow X(s) = (sI - A)^{-1}BU(s) \quad (1.16)$$

Remplaçant (1.16) dans (1.15) :

$$Y(s) = C(sI - A)^{-1}BU(s) + DU(s) \Leftrightarrow Y(s) = (C(sI - A)^{-1}B + D) U(s)$$

$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D$$

$$G(s) = \frac{Y(s)}{U(s)}$$

Finalement :

$$G(s) = C(sI - A)^{-1}B + D$$

$G(s)$ est la fonction de transfert. Elle facilite la résolution du système sans avoir recours au calcul complexe qu'offre l'équation différentielle.

Remarque 1.3

Le passage d'une représentation d'un système à une autre est possible, comme on vient de le démontrer pour la relation entre équation différentielle et fonction de transfert et encore la relation entre représentation d'état et fonction de transfert.

III. Généralité sur les systèmes de commande :

L'objectif de l'automatisation a conduit à une nouvelle théorie qu'est la théorie de contrôle et qui vise à analyser et synthétiser des systèmes dit : **système de commande** ou **de contrôle**. C'est un système qui doit être conçu de façon à commander le comportement d'un système [9].

III.1. Notion du système de commande :

Le terme commande désigne toute action exercée sur un système pour influencer son évolution dynamique selon certains critères désirés, appelés, cahier de charge. Cette commande est traduite par un signal, dit signal de commande qui affecte le système commandé et qui est issu d'un sous-système appelé système de commande, connu sous le terme correcteur.

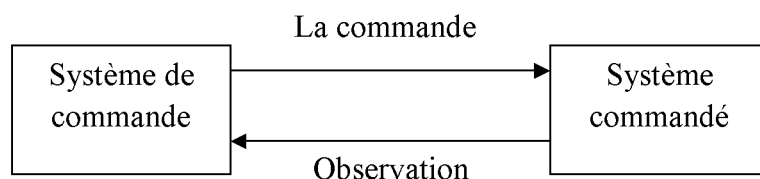


Figure 1.9. Schéma représentatif d'un système de commande.

III.2. Propriétés d'un système de contrôle :

Un système de contrôle doit assurer en boucle fermée les propriétés suivantes :

- **Stabilité** : La grandeur de sortie doit converger vers une valeur finie si le signal d'entrée est aussi borné.
- **Précision** : La grandeur à mesurer doit être la plus proche de celle désirée.
- **Rapidité** : Il doit répondre rapidement à une excitation.

III. 3. Type de commande :

Quelque soit le système étudié, on distingue deux types de commande : la commande en boucle ouverte et commande en boucle fermée, d'où la possibilité de la classification suivante :

- Système de commande en **Boucle Ouverte**.
- Système de commande en **Boucle Fermée**, appelé aussi, commande à contre-réaction ou à rétroaction.

III.3. 1. Commande en boucle ouverte :

On parle de commande en **Boucle Ouverte**, lorsque le système de commande n'a aucune information sur l'évolution et le comportement de la grandeur à régler. Donc il n'est pas possible d'évaluer la conformité des objectifs désirés.

La figure (1.9) représente un schéma fonctionnel d'un système en BO.

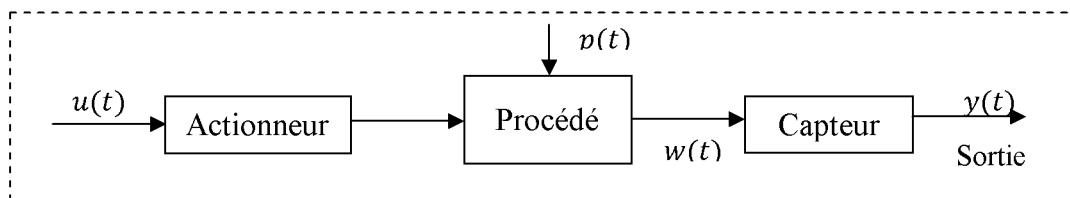


Figure 1.10. Représentation par schéma-bloc d'un procédé en boucle ouverte.

➤ Problèmes liés à la boucle ouverte :

- Il est impossible de commander un système instable (pendule inversé, sous-marin ...).

- Pour obtenir la sortie souhaitée, le système à commander doit être parfaitement connu afin de pouvoir calculer avec précision la valeur de $u(t)$ à appliquer. D'autre part, aucune perturbation $p(t)$ ne doit affecter le fonctionnement du système.

III.3. 2. Commande en boucle fermée :

On parle de système **bouclé** ou de système en **boucle fermée** par opposition au système en boucle ouverte. C'est donc une structure où le signal de commande $u(t)$ dépend d'une façon ou d'une autre du signal de sortie $y(t)$.

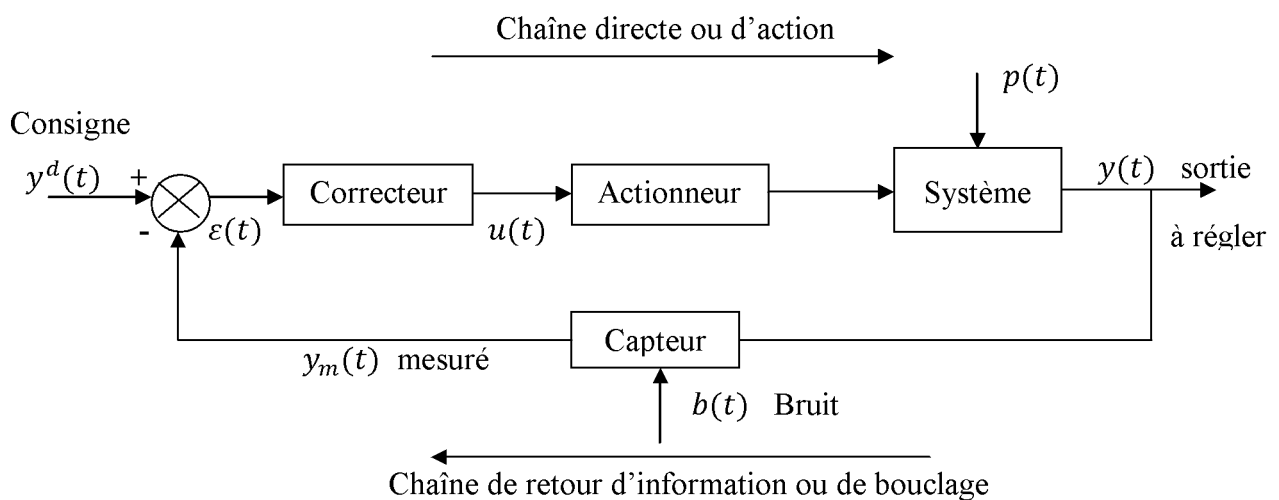


Figure 1.11. Structure générale d'un Système en boucle fermée.

$u(t)$: Signal de commande.

$\varepsilon(t)$: Ecart entre la consigne (valeur désirée) et la valeur mesurée.

$y^d(t)$: Consigne qui représente la valeur désirée.

Comme on le constate sur la figure précédente, un système de commande à rétroaction comporte les éléments principaux suivants :

- **Actionneur** : L'actionneur est un organe d'action qui apporte l'énergie nécessaire au système pour produire l'effet souhaité, il peut être un organe mécanique, hydraulique ou électrique. Par exemple une vanne, vérin... etc.
- **Capteur** : La mesure d'une grandeur de sortie s'effectue par un capteur. Le rôle du capteur est de délivrer un signal analogique qui soit le plus fidèle possible de la

grandeur à mesurer. La précision et la rapidité sont deux caractéristiques importantes du capteur.

- **Correcteur** : c'est un organe de réglage qui prend en compte, généralement, le signal d'écart $\varepsilon(t)$ entre la consigne (valeur souhaitée pour la sortie réelle $y^d(t)$) et la sortie mesurée $y(t)$. Le signal d'écart $\varepsilon(t)$ constitue la variable d'entrée du correcteur.

Remarque 1.4

Le correcteur peut jouer les deux rôles suivants :

- **Suiveur** : son rôle est de ramener le système commandé à suivre la consigne.
- **Régulateur** : son rôle est de rejeter les perturbations et corriger les imperfections du système pour lui assurer un certain niveau de stabilité souhaité.

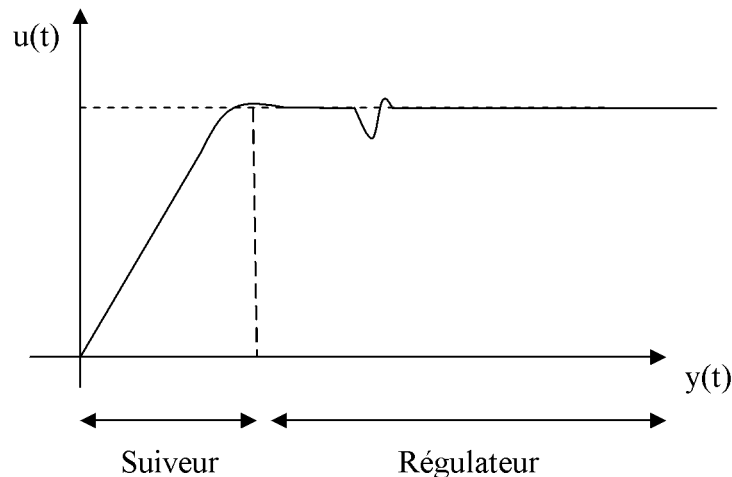


Figure 1.12. Schéma représentant les deux fonctions d'un correcteur.

➤ Problèmes liés à la boucle fermée :

La boucle fermée offre l'avantage de précision au système en le corrigeant, elle a aussi ses inconvénients :

- Il est nécessaire d'avoir des capteurs de mesures fidèles qui sont parfois très onéreux, ce qui augmente le coût d'une installation.
- Le problème de la stabilité et de la précision des systèmes en boucle fermée se pose de manière plus complexe que dans le cas de la structure de commande en boucle ouverte. Par exemple, une commande par contre-réaction peut rendre instable un système qu'est initialement stable en boucle ouverte.

- Il faut veiller à ce que le correcteur génère une commande $u(t)$ qui soit admissible par le système (ex. problème de saturation des actionneurs à prendre en compte).

III.4. Asservissement et régulation :

Un système de commande a deux atouts majeurs : asservir la sortie (grandeur à régler) pour lui imprimer les performances désirées (suivi de la trajectoire et la rapidité), ou la réguler pour la rendre stable en dépit des perturbations qui persistent.

- **Problème d'asservissement ou de poursuite ($y^d(t) = y(t)$) :**

Il consiste à asservir la sortie du système par une consigne $y^d(t)$ variante dans le temps et si des perturbations existent, elles sont considérées comme fixe. Le correcteur est dit suiveur. A titre d'exemple, un missile qui poursuit une cible.

- **Problème de régulation ($y^d(t) = 0$) :**

Il consiste à maintenir la sortie mesurée du système à une valeur constante en compensant l'effet des perturbations. Dans ce cas, $y^d(t) = 0$ et l'écart $\varepsilon(t)$ est dû aux perturbations que l'on doit rejeter en appliquant une commande $u(t)$ via le correcteur qu'est appelé dans ce cas régulateur. A titre d'exemple, réguler le niveau d'eau dans un réservoir.

Remarque 1.5

Avec le développement de la théorie de commande, la commande par rétroaction de la sortie s'est étendue à la commande par retour d'état, connue par la commande moderne.

III.5. Commande classique :

Le terme automatique ou commande classique regroupe les premières méthodes de commande, graphique pour la plupart, développées vers les années 1940 et 1950. Ces méthodes s'appliquent à des systèmes mono-variables et elles s'ouvrent à l'étude fréquentielle. Les critères d'encerclement de Nyquist, la conception sur le plan de Nicols ou le lieu géométrique des racines d'Evans sont des méthodes qui permettent de déduire des conclusions sur la BF à partir des caractéristiques de la BO. Il faut ajouter à cela, le critère algébrique de Routh-Hurwitz, qui sert à déduire la stabilité de la BO.

III.6. Commande moderne :

Les années 1960 et 1970 ont vu le développement d'une méthode alternative, la commande par variable d'état, aussi appelée commande moderne. La théorie de Wiener et de Kalman, les pionniers de cette théorie, entre autre, ont permis d'élaborer une théorie complète applicable à des signaux multivariables (MIMO).

Les problèmes de commande ont été reformulés pour des signaux échantillonnés plutôt que des signaux continus. La théorie de cette commande permet une étude temporelle et s'applique dans le domaine des signaux continus ou discrets. Dans ce dernier cas, on parle de commande numérique.

IV. CONCLUSION :

Dans ce chapitre nous avons vu les différentes généralités sur les systèmes et système de commande, cependant, en vue de la satisfaction du cahier des charges, l'automatique, notamment la théorie de la commande qui analyse et synthétise les systèmes de commande, elle aussi a pris le chemin vers la perfection des commandes établies.

Avec la commande moderne, la théorie de contrôle s'est penchée vers l'étude interne des systèmes par variables d'état au lieu de l'étude externe (entrée-sortie) offerte par la commande classique. Néanmoins, les modèles utilisés ne sont que des représentations imparfaites de la réalité, ce qui présente des écarts (incertitude) entre le modèle et le système réel et conduit à des commandes moins performantes.

Cependant, vers les années 1980, une nouvelle commande qui tient compte des ces incertitudes et qui réconcilié entre les deux commandes précédentes (la richesse de l'analyse fréquentielle (commande classique) et la simplicité et puissance des modèles et méthodes de synthèse par variables d'état (commande moderne)) fut apparaitre. Cette dernière, est connue par la commande robuste et elle fera objet de chapitre suivant.

Chapitre 02

Généralités sur la Commande Robuste

I. INTRODUCTION :

Au cours des dernières décennies, une grande partie de la pensée s'est penchée vers la modélisation de "l'incertitude" des systèmes. Bien avant, l'hypothèse était que le "système réel" appartenait à une certaine sphère centrée autour d'un modèle nominal [11]. Cependant, ce dernier ne reflète pas exactement le comportement du système réel, à cause des hypothèses de simplification et de réduction en phase de modélisation et de la présence de non linéarités, les perturbations et les bruits de mesure. Pour contourner ce problème, la nouvelle idée est d'incorporer ces incertitudes lors de la démarche de synthèse de la loi de commande.

Dans ce chapitre, nous allons aborder la notion de système incertain et nous discuterons comment les incertitudes sont, généralement, décrites de sorte qu'elles peuvent être bien prises en compte dans l'analyse de la robustesse des systèmes et/ou lors de la synthèse de la loi de commande. Nous aborderons, aussi, la notion de commande robuste et ses classifications, ainsi que quelques méthodes de synthèse, comme nous allons introduire une notion, très importante, considérée comme une limite pour la commande robuste et qu'est la complexité algorithmique.

II. Notion de système incertain :

Les systèmes incertains ou systèmes avec incertitude est une grande classe comprenant des systèmes de nature diverse (continu / discret, linéaire / non linéaire, etc), caractérisé par la présence de paramètres inconnus ou mal connus. Dans ces systèmes, l'objectif est généralement d'effectuer des tâches d'analyse et de synthèse pour une famille de valeurs admissibles de l'incertitude, afin de déterminer différents comportements possibles du système, ainsi que des stratégies de conception de contrôle robuste.

Un modèle incertain est un modèle où les incertitudes du système figurent dans la structure de son modèle. Du fait de la prise en compte de ces dernières, le modèle élaboré approche au mieux le système réel plus qu'un modèle nominal où les paramètres sont supposés d'une valeur connue et précise.

Représenté sous sa structure basique ou sous forme LFR (représentation linéaire fractionnaire (voir annexe)), le modèle faisant intervenir explicitement ces incertitudes (paramétrique et/ou non paramétrique) dans sa structure est bien le modèle linéaire à paramètre variant LPV. A titre d'exemple (dans l'espace d'état) :

$$\dot{x} = f(x(t), u(t), \Delta(t), t) \quad (2.1)$$

Avec : $\Delta(t)$ représente les incertitudes

III. Notion de l'incertitude :

L'incertitude est la différence entre la sortie du système et celle du modèle d'étude pour des conditions initiales identiques. Autrement dit, elle est définie comme les "différences" possibles entre le modèle et le système réel.

III. 1. Sources des incertitudes :

Il existe diverses sources d'incertitudes (d'erreurs) qui affectent le modèles, mais principalement, elles se résument en ces quatre sources fondamentales [23] :

III.1. 1. Perturbations externes négligées :

Chaque système dynamique dans le monde réel est constamment soumis à des perturbations de nature aléatoire qui ne peuvent pas toujours être modélisées (non mesurables). En l'absence de leurs modélisations, leurs effets sont considérés être une incertitude classée comme perturbation exogène.

III.1. 2. Non linéarités négligées :

La majorité des systèmes physiques possèdent des non linéarités, ce qui conduit à des structures de modèles non linéaires et rend leurs analyses complexes. Pour les étudier ou lors de la conception, on procède à leur linéarisation, autour du point de fonctionnement, et cela par la négligence des non linéarités ou des termes non linéaires du modèle. Citons par exemple, la linéarisation par l'expansion de la série Taylor, les effets non linéaires qui se produisent sous forme de termes d'ordre supérieur sont négligés afin d'obtenir un modèle linéaire. Mais cela conduit à un décalage entre le système physique et le modèle mathématique, et cet écart est traité comme une perturbation pour le système.

III.1. 3. Dynamique non modélisée :

Les divergences qui se produisent lorsque certains états ne sont pas considérés ou inclus dans le modèle dynamique (tels que ceux qui se produisent dans les techniques de réduction du modèle) sont considérés comme incertitudes et classés comme la dynamique non modélisée. En outre, dans certains cas, il est possible qu'il soit assez difficile de modéliser un phénomène complexe dans un cadre mathématique, ce qui se traduit par une erreur de modélisation.

III.1. 4. Variations des paramètres réels :

Lors de la modélisation d'un système, certains paramètres physiques tels que la masse sont supposés être d'une certaine valeur. Cependant, ces paramètres peuvent différer des valeurs réelles en raison soit de l'inexactitude dans la détermination ou un changement sur une période de temps. Une telle variation des paramètres réels peuvent être considérée comme une incertitude ou une perturbation pour le système.

III. 2. Classification et représentations des incertitudes :

Généralement les incertitudes sont regroupées selon deux classes : les incertitudes structurées et les incertitudes non structurées.

III.2. 1. Incertitudes structurées ou incertitudes paramétriques :

Ces perturbations peuvent être représentées par des variations de certains paramètres du système sur certaines plages de valeurs possibles (complexes ou réels) [12]. Ils affectent la performance de gamme basse fréquence.

Ici, les paramètres ζ_i sont non connus d'une valeur précise, mais, supposés bornés $\zeta_i \in [\zeta_i^-, \zeta_i^+]$.

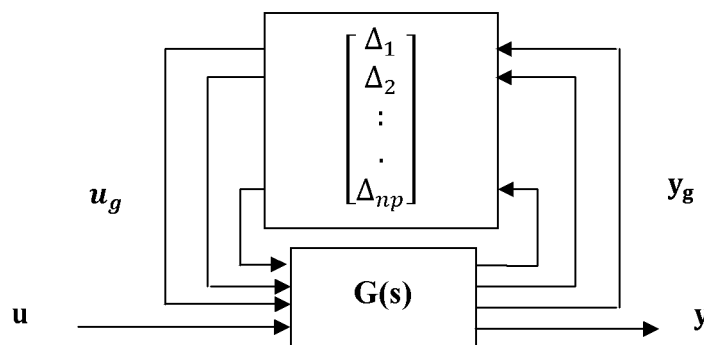


Figure 2.1. Représentation des incertitudes paramétrique.

III.2. 2. Incertitudes non structurées ou incertitudes dynamique :

Les représentations d'incertitude non structurées sont utiles pour décrire la dynamique des systèmes non modélisés ou négligés [12, 21]. Ces incertitudes complexes s'effectuent habituellement à la gamme des hautes fréquences et peuvent inclure des retards non modélisés (de temporisation), couplage parasite, hystérésis et d'autres non-linéarités.

Elles peuvent se représenter à l'aide des schémas blocs du **Figure 2.2**.

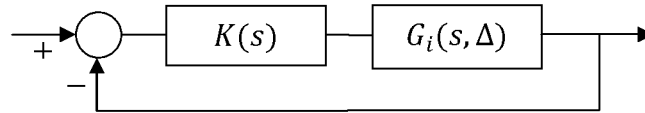


Figure 2.2. Représentation générale des incertitudes non structurées.

Le bloc $G_i(s, \Delta)$ peut avoir différentes structures qui se résument dans le **tableau 2.1**

Type de structure	Schéma de bloc $G_i(s, \Delta)$	Equation de la structure
Incertitude additive		$G_i(s, \Delta) = G(s) + \Delta_a(s)$
Incertitude multiplicative en entrée		$G_i(s, \Delta) = G(s)(I + \Delta_e(s))$
Incertitude multiplicative en sortie		$G_i(s, \Delta) = G(s)(I + \Delta_s(s))$
Incertitude multiplicative inverse en entrée		$G_i(s, \Delta) = G(s)(I + \Delta_{i_e}(s))^{-1}$

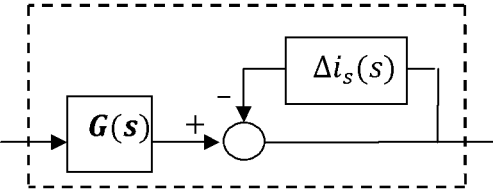
<p>Incertitude multiplicative inverse en sortie</p>		$G_i(s, \Delta) = G(s)(I + \Delta i_s(s))^{-1}$
--	---	---

Tableau 2.1 Représentation des incertitudes non paramétriques.

Exemple d'un système incertain

Considérons le schéma électromécanique suivant :

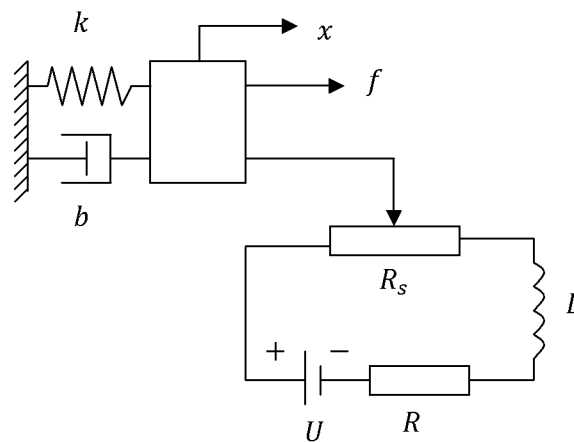


Figure 2.3. Schéma d'un système électromécanique.

▪ **Modélisation du système :**

Le système, comme représenté, ci-dessus, comporte deux sous-systèmes (mécanique et électrique) combinés. Pour la modélisation, on procède à la modélisation de chaque sous-système avant de les combinés pour avoir le modèle de système global.

- Pour le système mécanique :

D'après la deuxième loi de Newton, on a :
$$\sum \vec{F}_{ex} = m \vec{a} \tag{2.2}$$

Où : \vec{a} désigne l'accélération de la masse tel que :
$$\vec{a} = \frac{dx^2(t)}{dt^2} = \ddot{x}(t) \tag{2.3}$$

$$m \ddot{x}(t) = -k x(t) - (f_r - b)\dot{x}(t) + f(t) \tag{2.4}$$

En négligeant les frottements ($f_r = 0$), on aura finalement :

$$m \ddot{x}(t) + b \dot{x}(t) + k x(t) - f(t) = 0 \quad (2.5)$$

- Pour le système électrique :

En appliquant la loi des mailles :

$$-u(t) + L \frac{di(t)}{dt} + (R_s + R)i(t) = 0 \quad (2.6)$$

Finalement : $L \ddot{q}(t) + (R_s + R)\dot{q}(t) - u(t) = 0 \quad (2.7)$

Où : $i(t) = \frac{dq(t)}{dt} \quad (2.8)$

Il en résulte le modèle du système global (2.9), regroupant (2.5) et (2.7), qui sont liées par la relation $R_s = F(x(t))$, sachant que R_s est un paramètre incertain pour le système puisque : $R_{s_{min}} \leq R_s \leq R_{s_{max}}$:

$$\begin{cases} m \ddot{x}(t) + b \dot{x}(t) + k x(t) - f(t) = 0 \\ L \ddot{q}(t) + (R_s + R)\dot{q}(t) - u(t) = 0 \end{cases} \quad (2.9)$$

IV. Notion de robustesse :

L'analyse de robustesse est considérée comme étant une étape de validation à posteriori de la loi de commande. La robustesse d'un système est traduite par la capacité de ce dernier à préserver certaines de ses caractéristiques essentielles, telle que la stabilité et/ou les performances, en présence de bruit et des perturbations inconnues, ainsi que des incertitudes du modèle.

En analyse de robustesse, on s'intéresse d'abord à la stabilité du système et on parle alors de *robustesse en stabilité* ou de *stabilité robuste*. On peut également, aller plus loin, et analyser si le système incertain satisfait un critère de performance on parle, alors, de *robustesse en performance* ou de *performance robuste*. Ces analyses peuvent être menées à deux niveaux :

- On peut se limiter à la question de savoir si le critère de performance est satisfait par l'ensemble des incertitudes prévues. Le résultat d'un tel test est binaire (oui ou non). C'est un problème d'analyse et aussi problème de décision.
- On peut aussi chercher à connaître la dilatation que l'on peut appliquer au domaine de variation initial pour amener le système à la limite de robustesse. Le taux de dilatation est aussi appelée marge de robustesse (problème de synthèse ou problème d'optimisation).

Ainsi, on parle de commande robuste si les incertitudes sont explicitement prises en compte lors de la synthèse de la loi de commande et le correcteur est dit robuste s'il garantit les marges de robustesse, spécifiées même lorsque le système s'écarte du modèle.

V. Commande robuste :

La commande robuste est une discipline consistant à établir une loi de commande qui puisse être implantée sur un système dont le modèle est entaché d'une incertitude.

La loi de commande établie, doit garantir la stabilité du système et un certain niveau de ses performances malgré les perturbations du son environnement et des incertitudes de son modèle.

Ainsi, on dit d'un système qu'il est robuste s'il conserve sa stabilité et des performances acceptables malgré la présence des incertitudes [20].

V. 1. Classification des méthodes (commandes) robustes :

Généralement la synthèse d'un correcteur robuste se ramène à résoudre un problème d'optimisation qui dépend des incertitudes. La plupart des problèmes de conception sont formulés dans le domaine fréquentiel. Mais des techniques algébriques sont aussi disponibles.

Prendre les incertitudes en considération dans la conception du correcteur est généralement impossible et rend le problème en considération non tractable, c'est-à-dire de complexité NP. Dans cette optique, des méthodes probabilistes sont proposées dans la littérature [11]. Ainsi, les commandes robustes sont classées comme suit :

V.1. 1. Méthodes déterministes :

Ces méthodes essaient de résoudre le problème d'optimisation en prenant en considération la totalité des incertitudes mais conduisent généralement à des problèmes d'optimisation semi-infinie. Des solutions algébriques sont encore disponibles qui permet de réduire la complexité mais souffrent toujours du problème de la complexité lorsque l'ordre du système est grand.

V.1.1. 1. Quelques méthodes déterministes :

A. Méthode 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 performance et la stabilité du système bouclé avec un certain taux d'atténuation de l'influence des entrées exogènes sur les sorties du système.

A.1. Définition de H_∞ :

H_∞ est une méthode qui sert à la conception de commandes, elle est utilisée dans la théorie de commande pour synthétiser des correcteurs consistant à trouver un compromis permettant de garantir au mieux la stabilité et les performances demandées pour une large plage d'incertitudes.

A.2. Notion du petit gain : [11], [18]

Théorème 2.1 : (théorème du petit gain).

La commande H_∞ est bien connue pour ses propriétés de robustesses issues du théorème du petit gain. Ce dernier, permet d'affirmer que le système présenté sur la figure 2.4 est stable, pour toute incertitude vérifiant que : $\|\Delta(s)\|_\infty \leq \gamma$ (respectivement

$\|\Delta(s)\|_\infty < \gamma$) et si est seulement si $\|M(s)\|_\infty \leq \frac{1}{\gamma}$ (respectivement $\|M(s)\|_\infty < \frac{1}{\gamma}$) .

Si $\|M(s)\|_\infty \leq \gamma^{-1}$ alors $\|\Delta(s)M(s)\|_\infty \leq \|\Delta(s)\|_\infty \gamma^{-1} \leq \gamma \cdot \gamma^{-1} \leq 1$

Le système bouclé est aussi stable si : $\|\Delta(s)M(s)\|_\infty \leq 1$ (1.10)

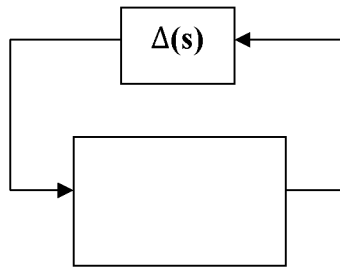


Figure 2.4. Schéma d'analyse de la robustesse de la stabilité.

Remarque 2.1

La norme infinie est la taille de la plus grande valeur singulière de la fonction de transfert sur toutes les fréquences. De même, toute norme induite entraînera les mêmes résultats.

A.3. Synthèse H_∞ :

La synthèse H_∞ doit être plutôt considérée comme une façon particulière de calculer un correcteur, sans que toutes les demandes de robustesse soient prises en compte à priori. Elle permet de modéliser différents transferts du système asservi, et ainsi de garantir des marges de stabilité globales (marge de gain, marge de phase ...etc.)[20].

Pour aboutir aux objectifs de synthèse, on doit passer par ces deux approches suivantes :

A.3.1. Introduction de fonction de pondération.

Pour atteindre les objectifs de la synthèse, on peut introduire des pondérations sur les différents signaux, qui prendront la forme de filtres permettant, suivant le signal auquel ils s'appliquent, de privilégier un domaine de fréquence particulier.

Le schéma suivant dans lequel ε est pondérée par le filtre $W_1(s)$ la commande u par $W_2(s)$ et l'entrée de perturbation b est la sortie d'un filtre $W_3(s)$.

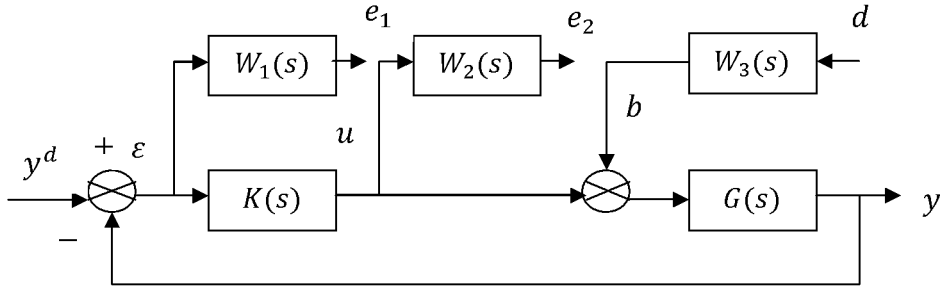


Figure 2.5. Système soumis aux pondérations.

En considérant y^d et d comme des entrées et e_1 , e_2 comme les signaux à surveiller, on obtient alors : (avec $S(s)$ la fonction de sensibilité (voir l'annexe))

$$\begin{bmatrix} E_1(s) \\ E_2(s) \end{bmatrix} = \begin{bmatrix} W_1(s)S(s) & W_1(s)S(s)G(s)W_3(s) \\ W_2(s)K(s)S(s) & W_2(s)K(s)S(s)G(s)W_3(s) \end{bmatrix} \begin{bmatrix} Y^d(s) \\ D(s) \end{bmatrix} \quad (2.11)$$

A.3.1.1. Choix de la fonction des pondérations :

Le problème H_∞ standard qui en découle se base sur la détermination d'un scalaire $\gamma > 0$ et le correcteur $K(s)$ stabilisant le système bouclé et assurant :

$$\left\| \begin{bmatrix} W_1(s)S(s) & W_1(s)S(s)G(s)W_3(s) \\ W_2(s)K(s)S(s) & W_2(s)K(s)S(s)G(s)W_3(s) \end{bmatrix} \right\|_\infty < \gamma \quad (2.12)$$

Les propriétés de la norme H_∞ assurent en effet que si la condition (2.12) est vérifiée, alors les 4 conditions suivantes le sont aussi.

$$\|W_1S\|_\infty < \gamma \Leftrightarrow \forall \omega \in \mathcal{R} |S(j\omega)| < \frac{\gamma}{|W_1(j\omega)|} \quad (2.13)$$

$$\|W_2KS\|_\infty < \gamma \Leftrightarrow \forall \omega \in \mathcal{R} |K(j\omega)S(j\omega)| < \frac{\gamma}{|W_2(j\omega)|} \quad (2.14)$$

$$\|W_1SGW_3\|_\infty < \gamma \Leftrightarrow \forall \omega \in \mathcal{R} |S(j\omega)G(j\omega)| < \frac{\gamma}{|W_1(j\omega)W_3(j\omega)|} \quad (2.15)$$

$$\|W_2KSGW_3\|_\infty < \gamma \Leftrightarrow \forall \omega \in \mathcal{R} |K(j\omega)S(j\omega)G(j\omega)| < \frac{\gamma}{|W_2(j\omega)W_3(j\omega)|} \quad (2.16)$$

On voit donc que la réponse fréquentielle de chacune des fonctions S , KS , SG et KSG est contrainte par un modèle qui dépend des filtres choisis.

A.3.1.2. Mise sous forme standard :

En pratique, on choisit les filtres $W_1(s)$, $W_2(s)$ et $W_3(s)$ d'après les considérations précédentes et on résout le problème H_∞ correspondant, qui donne la valeur de γ et le correcteur. Bien sûr la valeur de γ n'est pas connue à l'avance, elle intervient dans les modèles et on oriente le choix des filtres de façon à avoir une valeur de γ proche de 1. Une fois les filtres choisis, il reste à mettre le problème sous la forme standard

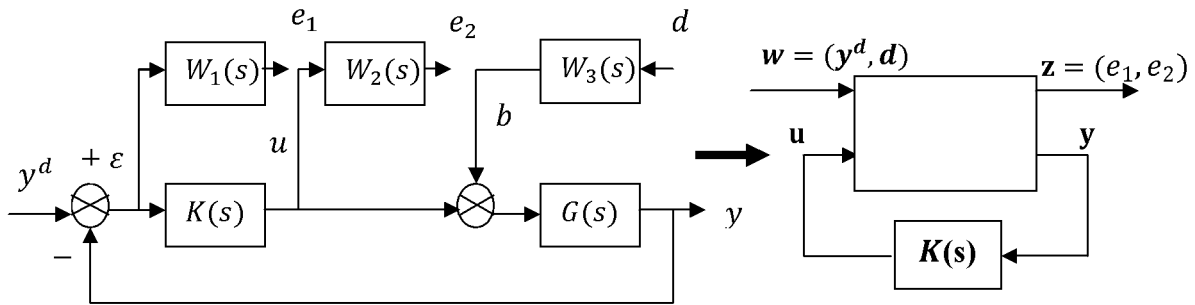


Figure 2.6. La transformation du problème sous une forme standard.

A.4. Notion de forme 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. 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 (2.6).

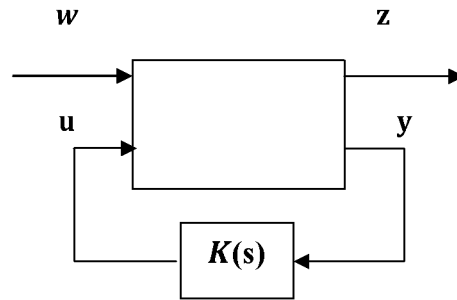


Figure 2.7. Forme standard du problème H_∞ .

Dans cette figure $M(s)$ représente le modèle augmenté, et $K(s)$ le correcteur. Le système $M(s)$ est formé des quatre transferts multivariables existantes 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) :

$$\begin{bmatrix} Z(s) \\ Y(s) \end{bmatrix} = M(s) \begin{bmatrix} W(s) \\ U(s) \end{bmatrix} = \begin{bmatrix} m_{11}(s) & m_{12}(s) \\ m_{21}(s) & m_{22}(s) \end{bmatrix} \begin{bmatrix} W(s) \\ U(s) \end{bmatrix} \quad (2.17)$$

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 $u = Ky$ connue, $T_{zw}(s)$ peut se calculer de la façon suivante :

$$T_{zw}(s) = F_l(M, K) = m_{11}(s) + m_{12}(s) \cdot K(s) (sI - m_{21}(s)K(s))^{-1} m_{21}(s) \quad (2.18)$$

A.4.1. Problème H_∞ standard :

Étant donné $\gamma > 0$, trouver un compensateur $K(s)$ qui stabilise le système de manière interne et assure que :

- Le système $F_l(M, K)$ soit asymptotiquement stable (Tous les pôles du système en boucle fermée sont à partie réelle négatives).
- On définit la performance d'un système si la norme H_∞ de : $\|F_l(M, K)\|_\infty < \gamma$.

B. Méthode de Kharitonov:

La méthode de Kharitonov est très répandue dans le domaine de contrôle robuste qui cherche à construire un système robuste aux incertitudes paramétrique dues aux mesures, changement de conditions du système, ... etc. Elle étudie la robustesse en stabilité du système continu et incertain. Les plages des variations des paramètres du système sont supposées connues.

Théorème 2.2 (théorème de Kharitonov)

Dans cette section nous considérons que les paramètres incertains sont de type :

$$\zeta = [\zeta_0 \dots \zeta_l]^T$$

Où : chaque $\zeta_i (i = 0, 1, \dots, l)$ est borné dans l'intervalle $[\zeta_i^-, \zeta_i^+]$, et ζ_i^-, ζ_i^+ désignent respectivement la borne inférieure et la borne supérieure de l'intervalle contenant ζ_i , et notons que ces intervalles sont indépendants.

Kharitonov considère le polynôme \mathcal{P} défini comme suit :

$$\mathcal{P} = \{ p(s, \zeta) = \zeta_0 + \zeta_1 s + \dots + \zeta_n s^n : \zeta_i \in [\zeta_i^-, \zeta_i^+], \quad i = 0, 1, \dots, l \} \quad (2.19)$$

Alors, quatre polynômes caractéristiques dit polynômes de Kharitonov seront définis comme suit :

$$p_1(s) = \zeta_0^- + \zeta_1^- s + \zeta_2^+ s^2 + \zeta_3^+ s^3 + \zeta_4^- s^4 + \zeta_5^- s^5 + \zeta_6^+ s^6 + \dots \quad (2.20)$$

$$p_2(s) = \zeta_0^+ + \zeta_1^+ s + \zeta_2^- s^2 + \zeta_3^- s^3 + \zeta_4^+ s^4 + \zeta_5^+ s^5 + \zeta_6^- s^6 + \dots \quad (2.11)$$

$$p_3(s) = \zeta_0^+ + \zeta_1^- s + \zeta_2^- s^2 + \zeta_3^+ s^3 + \zeta_4^+ s^4 + \zeta_5^- s^5 + \zeta_6^- s^6 + \dots \quad (2.12)$$

$$p_4(s) = \zeta_0^- + \zeta_1^+ s + \zeta_2^+ s^2 + \zeta_3^- s^3 + \zeta_4^- s^4 + \zeta_5^+ s^5 + \zeta_6^+ s^6 + \dots \quad (2.23)$$

Ainsi, le système est stable si et seulement si les quatre polynômes de Kharitonov sont stables. (La stabilité sera étudiée par le critère de Routh-Hurwitz ou une autre approche d'étude de stabilité).

C. Méthode du loop-shaping :

Le domaine fréquentiel est un outil puissant pour l'analyse et la synthèse des correcteurs. Les performances désirées (rejet de perturbation, rapidité, etc.) peuvent être assurées en imposant une certaine forme pour le lieu fréquentiel du système en BF. Parmi les techniques les plus répandues on trouve les méthodes de modelage ou de mise en forme de boucle (loop-shaping).

On appelle spécification loop-shaping le fait de spécifier la boucle ouverte de l'asservissement à travers les divers spécifications désiré en BF, car il est plus facile de travailler sur un unique transfert que sur une multitude de transfert.

Les problèmes de rejet de perturbation peuvent être théoriquement résolu simplement en plaçons des réseaux passe bas et passe haut dans le modelage de la boucle ouverte. Le choix de pulsation ω_b et ω_h ($W_1(s)$ et $W_2(s)$) dépend des applications spécifiques et des informations qu'on possède sur les perturbations et les bruits.

La synthèse peut être menée sur la fonction de transfert ou les fonctions de sensibilité du système :

- **Synthèse du correcteur par modelage de la fonction de transfert du procédé :**

Une possibilité est de commencer par la fonction de transfert en boucle du procédé et de modifier plus ou en changeant le gain et l'ajout de pôles et les zéros au contrôleur jusqu'à ce que la forme désirée soit obtenue.

- **Synthèse du correcteur par modelage de la fonction de sensibilité :**

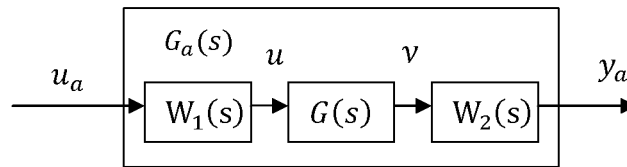
Une stratégie différentes consiste à de définir les spécifications directement sur les fonctions de sensibilité et de reformuler le problème de synthèse en un problème d'optimisation. En effet, la sensibilité est en particulier un bon indicateur des performances du système.

Lorsque le système est sujet à des incertitudes, il est nécessaire d'associer une commande robuste à la procédure de modelage.

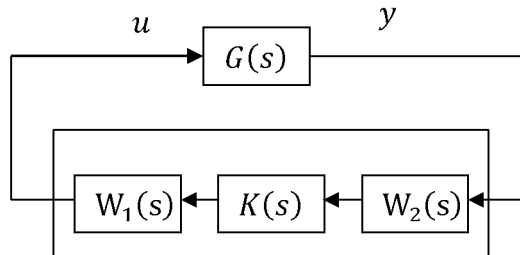
C.1. Synthèse H_∞ par la méthode de loop-shaping

Le réglage des performances est donc obtenu en effectuant un modelage en boucle ouverte du processus avant de calculer le correcteur. La démarche est la suivante [20, 21] :

(i) Ajouter à la matrice $G(s)$ du système à réguler un pré-compensateur $W_1(s)$ et/ou un post-comparateur $W_2(s)$, de sorte que le système augmenté $G_a(s) = W_2(s)G(s)W_1(s)$ présente un comportement en fréquence satisfaisant : typiquement, en assurera un grand gain en basse fréquence (par exemple au moyen d'une action intégral), de faible gain dans les hautes fréquences, et en choisira la fréquence de passage à 0dB qui correspond à peu près à la bande passante du système bouclé.



a) Modelage du processus.



b) Structure de commande.

Figure 2.8. Synthèse par loop-shaping.

La structure finale de correction est obtenue en combinant le correcteur K et les compensateurs $W_1(s)$ et $W_2(s)$

En effet l'étape (i) est suivie d'une synthèse H_∞ qui, si nécessaire, améliorera les marges de stabilité, en pratique les compensateur $W_1(s)$ et $W_2(s)$ sont des réseaux classiques de type PI, avance ou retard de phase, filtre passe bas, ... etc.

Pour conclure, la simplicité de l'approche étant par ailleurs un atout : le modelage d'un seul transfert (la boucle ouverte) permet la mise en œuvre de bouclage performant vis-à-vis de signaux externe et robuste vis-à-vis des incertitudes.

V.1.2. Méthodes probabilistes :

Le principe de ces méthodes est le même que celui des méthodes déterministes mais pour échapper à la complexité, on procède par relaxation des problèmes d'optimisation en réduisant le nombre de contraintes par la méthode de Monte Carlo. L'objectif est de ramener le nombre de contraintes infini en un nombre fini. Cette étape est basée sur des choix à effectuer avec certaines probabilités de distribution sur les incertitudes.

V.2. Inconvénient des commandes robustes :

La première question critique du paradigme de la robustesse est la complexité de calcul. En particulier, divers problèmes de contrôle robustes sont classés NP-dur (NP-difficile) [11].

V.3. Notion de complexité algorithmique :

Les problèmes se classent en deux grandes catégories : les problèmes résolubles et non résoluble, selon qu'il existe ou non un algorithme de résolution [7]. Les problèmes algorithmiques ou bien résolubles, c'est-à-dire qui peuvent être traités par ordinateur, répondent à une structure algorithmique : *hypothèses*, *données* et une *question*, et ils se répartissent, de leurs parts, en deux grandes catégories :

- Les problèmes de décision : ils posent une question dont la réponse est *oui* ou *non* ;
- Les problèmes d'existence ou de recherche d'une solution ou encore dit de calcul : ils comportent une question ou plutôt une injonction de la forme « *trouver un élément tel que ...* » dont la réponse consiste à fournir l'élément attendu.

V.3.1. Théorie de la complexité algorithmique :

Cette théorie est un domaine des mathématiques, plus précisément, de l'informatique théorique qui étudie, formellement, la *quantité de ressources* (en temps de calcul et en espace mémoire) nécessaire pour résoudre un problème au moyen de l'exécution d'un algorithme.

V.3.2. Classe de complexité :

On s'intéressera à la complexité en temps, c'est-à-dire la mesure du temps de calcul d'exécution d'un algorithme. Selon la fonction que satisfait le temps en fonction de la taille du problème, plusieurs classes d'algorithme seront tirées. Par exemple un algorithme pour qui le temps du calcul croît en une fonction polynomiale (en fonction de la taille de problème) est

appelé : complexité polynomiale, et ainsi, on distingue la complexité exponentielle, quadratique, factoriel... etc.

Remarque 2.2

Les algorithmes fonctionnant en un temps polynomial sont dit « efficaces », et les problèmes résolus par ces derniers sont dit polynomiaux ou facile.

V.3.3. Classification des problèmes :

Selon le temps de calcul, la théorie de complexité classe les problèmes comme suit [7]:

V.3.3.1. Classe P (Polynomial time) :

C'est la classe des problèmes pouvant être résolus en un temps polynomial, en fonction de taille des données, sur une machine de Turing déterministe. Les problèmes de cette classe sont dit faciles.

V.3.3.2. Classe NP (Non Deterministic Polynomial time) :

Ensemble de problèmes pouvant être résolus sur une machine de Turing non-déterministe en un temps non polynomial, mais, si une solution existe, on peut vérifier qu'elle répond bien à une complexité polynomiale. Ces problèmes sont de type de décision et ils sont plus durs que ceux de la classe P, mais qui sont résolubles.

V.3.3.3. Classe NP-complet :

Elle représente les problèmes de la classe NP qui sont liés. Un problème est *NP-Complet* s'il est dans la classe NP et si on peut le ramener, par une transformation polynomiale, à un autre problème connu de la classe NP. Il faut noter que se sont des problèmes de décision, mais difficiles et non résolubles par une complexité polynomiale.

V.3.3.4. Classe NP-difficile ou NP-dur :

On ne s'intéresse plus aux problèmes de décision seulement, mais aussi aux problèmes d'optimisation qui sont des problèmes de calcul. Un problème NP-difficile est un problème d'optimisation associé à un problème de décision de classe NP-complet [4]. Une autre manière

de définir un problème NP-difficile est si un problème est au moins aussi dur que le problème le plus difficile dans la classe NP alors nous disons que ce problème est NP-difficile [14].

Un problème NP-dur est généralement considéré comme un «problème insoluble». Ce type de problème est très difficile à résoudre et sa résolution, de façon optimale en un temps raisonnable fait, souvent, appel à des algorithmes qui accélèrent la résolution comme la 'Programmation Dynamique', et parfois, on se penche vers des méthodes approchées donneront une solution aussi proche que possible de l'optimum en un temps «raisonnable» comme les "Heuristiques" et les "Métaheuristiques".

Remarque 2.3

Les termes déterministe et non déterministe décrivent la machine utilisée lors de l'exécution et non pas le temps.

Pour conclure, la théorie de complexité nous permet de :

- classer les algorithmes selon leur efficacité ;
- classer les problèmes selon leur difficultés ;
- prévoir le temps d'exécution d'un algorithme ;

Et ainsi, elle nous donne une idée sur degré de la difficulté à laquelle on est confronté et nous permet de mieux choisir l'algorithme de résolution qui convient avant de procéder à son implémentation.

VI. Conclusion :

La théorie de la commande robuste regroupe plusieurs méthodes d'analyse et de synthèse de contrôleurs réduisant l'écart entre le système réel et le modèle identifié. Elles sont classées en méthodes déterministes et méthodes probabilistes. En outre, de nombreux problèmes de synthèse des correcteurs robustes sont souvent formulés comme des problèmes d'optimisation [15].

Le chapitre suivant aborde l'optimisation et en particulier l'optimisation robuste qui modélise les incertitudes.

Chapitre 03

Optimisation robuste par la méthode des scénarios

I. Introduction :

Comme indiqué dans le chapitre précédent, synthétiser un correcteur pour un système incertain se ramène généralement à résoudre un problème d'optimisation robuste.

Ce présent chapitre, porte sur les notions de base de *l'optimisation*, ainsi sur les *aspects généraux des méthodes de résolution* où on présentera *les algorithmes génétiques*, avec *un exemple illustratif*, comme l'une des méthodes les plus efficaces pour la résolution d'un problème d'optimisation. Ensuite, l'intention se focalise sur *l'optimisation en présence d'incertitudes* connue par *l'optimisation robuste*. Toutefois, les problèmes survenant sont d'une complexité grandiose, voire NP-dur. Pour contourner cette difficulté '*l'approche des scénarios*' basée sur la relaxation du problème par randomisation du domaine d'incertitude selon une certaine loi probabiliste peut être utilisée.

II. Généralité sur l'optimisation :

II.1. Optimisation :

L'optimisation est une discipline combinant plusieurs domaines de compétences : les mathématiques décisionnelles, les statistiques et l'informatique. Cette méthode scientifique a pour but de maximiser ou de minimiser un objectif. En pratique l'optimisation est souvent utilisée pour augmenter la rentabilité et/ou diminuer les coûts.

II.2. A quoi sert l'optimisation ?

L'optimisation vise à déterminer une solution optimale selon un critère, prenant en compte un ensemble de contraintes et de variables inhérentes au problème. Dans la pratique, il n'existe pas de solution optimale, parce que la représentation mathématique d'un problème n'est jamais exacte. Il faut donc déterminer de bonnes solutions en se basant sur de bons modèles qui se doivent d'être fiables et robustes. Et donc, pour décrire et éventuellement résoudre un problème d'optimisation donné, nous utilisons la modélisation mathématique.

II.3. Mise en forme mathématique d'un problème d'optimisation :

La démarche de modélisation d'un problème d'optimisation, comporte les étapes suivantes [1] :

- **Etape 1 :** Analyser le problème en choisissant les variables de décision, noté x , qui sont inconnus à déterminer. Si on estime un nombre fini noté $n \in \mathbb{N}^*$ de ces variables, donc elles seront représentées par un vecteur $x = [x_1 x_2 x_n]^T$;
- **Etape 2.** Déterminer le critère à optimiser. Ceci revient mathématiquement à se donner une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ appelée fonction-objectif et que nous voulons rendre la plus petite ou la plus grande possible ;
- **Etape 3.** Décrire les contraintes que les variables de décision vont satisfaire. Ceci, revient à définir un ensemble $\Omega \rightarrow \mathbb{R}^n$ et imposer d'avoir $x \in \Omega$.
- **Etape 4.** Simplifier le problème (par élimination des variables ou des contraintes)
- **Etape 5.** Choisir la méthode de résolution adéquate, selon le type du problème.

II.4. Quelques définitions :

II.4.1. Notion d'optimum :

Le mot optimum désigne un maximum et/ou un minimum et on distingue deux types d'optimum : optimum local et optimum global.

Plus formellement, pour une fonction donnée $f(x) : \Omega \rightarrow \mathbb{I} \subset \mathbb{R}$.

- La fonction $f(x)$ admet un minimum, dit local, noté \hat{x} , (respect. un maximum local) si $\forall x \in \vartheta(\hat{x}) \subset \Omega$, ($\vartheta(\hat{x})$ voisinage de \hat{x}), alors $f(\hat{x}) \leq f(x)$ (respect. $f(\hat{x}) \geq f(x)$) (3.1)

- La fonction $f(x)$ admet un minimum global, noté x^* , (respect. un maximum global) si $\forall x \in \Omega$, on a : $f(x^*) \leq f(x)$ (respect. $f(x^*) \geq f(x)$). (3.2)

Remarque 3.1 : L'optimum global n'est pas forcément unique.

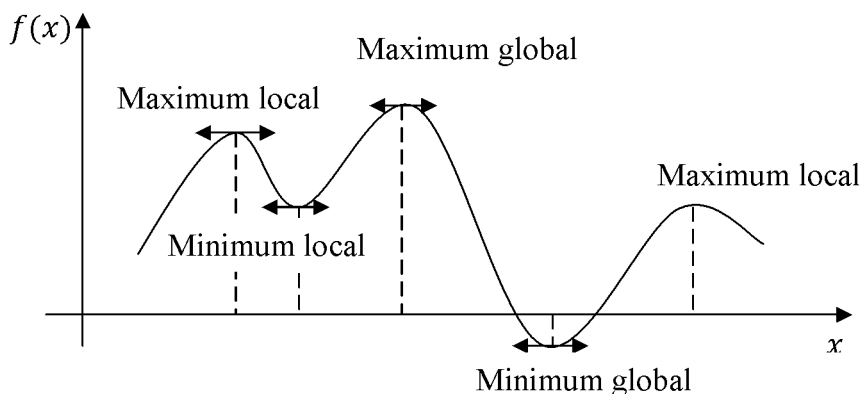


Figure 3.1. Illustration du type d'optimum

II.4.2. Qu'est-ce qu'une contrainte ?

Ensemble des enjeux et des limites physiques (ex. sécurité/stabilité, limite de ressources) relatives au problème qui sont traduit mathématiquement par équations ou inéquations composées de variables de décision et qui limitent le modèle réalisable de la fonction-objectif.

On distingue trois types de contrainte : contraintes égalités, contraintes inégalités supérieures et contraintes inégalités inférieures.

- **Contraintes égalités** : Représente un ensemble d'équations $g_i(x) = 0 \quad ; i = 1, \dots, p$
- **Contraintes inégalités** : Représente un ensemble d'inéquation et on distingue deux types d'inéquation :

$$\text{Inégalités supérieurs : } h_j(x) \geq 0 \quad ; j = 1, \dots, q \quad (3.3)$$

$$\text{Inégalités inférieurs : } h_j(x) \leq 0 \quad ; j = 1, \dots, q \quad (3.4)$$

Remarque 3.2

Il existe un autre type de contrainte, dit bornée et qui se rapporte à deux contraintes inégalités.

II.5. Formulation mathématique d'un problème d'optimisation :

La forme mathématique générale d'un problème d'optimisation est la suivante :

$$\text{Optimum}_{x \in \Omega \subset \mathbb{R}^n} \quad f(x) \quad (3.5)$$

Soumise à :

$$g_i(x) = 0 \quad ; i = 1, \dots, p \quad (3.6)$$

$$h_j(x) \leq 0 \quad ; j = 1, \dots, q \quad (3.7)$$

Où :

- Optimum : désigne le type de problème (problème de minimisation ou de maximisation).
- $f(x)$: est appelée fonction de coût, fonction objectif, ou encore critère d'optimisation.
- x : vecteur de variable de décision ou d'optimisation. $x = [x_1 x_2 \dots x_n]^T$
- $g_i(x)$ et $h_j(x)$ sont respectivement les contraintes d'égalité et d'inégalité.
- p , q et n sont respectivement le nombre de contraintes égalités, nombre de contraintes inégalités et le nombre de variable de décision.

- Ω : est l'ensemble admissible de variable de décision.

II.6. Comment classifier un problème d'optimisation ?

La résolution d'un problème d'optimisation, notamment le choix d'une méthode adéquate, se fait par rapport au type et la classe du problème considéré, et comme il existe un grand nombre de classes, nous optons aux modes de classification de ces derniers.

A. Type du problème, on distingue :

Problèmes de maximisation et problèmes de minimisation.

B. La fonction-objectif (le nombre de fonction) :

- Optimisation mono-objectif ;
- Optimisation Multi-objectifs {Ici, la fonction-objectif contient deux ou plusieurs fonctions ou bien dite fonctionnelle $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ }.

C. Présence ou non présence de contraintes, alors deux grandes classes seront distinguées :

- Optimisation sans contraintes ;

Cette classe contient seulement le critère à optimiser.

- Optimisation sous contraintes.

Le problème est soumis à des contraintes, et il est du même classé selon :

- Le type de contraintes rencontrées :

- Problème d'optimisation avec contraintes égalités/inégalités ;
- Problème d'optimisation avec contraintes mixtes (c.à.d. contraintes égalités et inégalités en même temps).

- Le nombre de contraintes rencontrées, on distingue :

- Problème d'optimisation avec un nombre contrainte fini ;
- Problème d'optimisation avec un nombre contrainte infini.

D. La nature des variables de décision x :

- Sont réelles : on parle alors d'optimisation continue ;
- Sont entières, voire booléennes (0 ou 1) (optimisation combinatoire ou discrète).

Et dans le cadre même de l'optimisation continue, on catalogue les problèmes suivant les propriétés (linéarité, convexité, différentiabilité, etc.) des fonctions f, g_i et h_j . On distingue ainsi des problèmes :

- Linéaires et non linéaires ;
- Convexes et non convexes ;
- Quadratiques ;
- Différentiables ou non différentiables
- Déterministes ou stochastiques ;

Remarque 3.3

- Un problème d'optimisation sous contraintes peut toujours être ramené à un problème d'optimisation sans contrainte. Et qu'un problème de minimisation peut se ramener à un problème de maximisation puisque $(\max_{x \in \Omega} f(x) = -(\min_{x \in \Omega} -f(x)))$ et vice versa [2].
- Lorsque les variables de décision et/ou les paramètres des fonctions dépendent du temps c.à.d. $x(t)$, alors, ces derniers ne seront plus des variables statiques et l'optimisation est dite optimisation dynamique, qu'est connue par la commande optimale [1].

III. Résolution d'un problème d'optimisation :

A cause de la diversité des problèmes d'optimisation rencontrée, la recherche de l'optimum correspondant, fait appel à un arsenal de méthodes, mais on remarque bien que la totalité de ces méthodes, se résument en ces quatre aspects généraux qui suivent [3, 4] :

III.1. Méthodes globales et méthodes locales :

Selon l'optimum recherché, on distingue deux types de méthodes : locale et globale. La tâche des méthodes locales, consiste à déterminer l'optimum local, propre au problème parmi son ensemble de solution admissible, alors que, les méthodes globales se penchent à trouver l'optimum global (autrement dit ; rechercher le minimum des minima locaux ou le maximum des maxima locaux), ce qu'est l'objectif désiré et utile pour, la plus part, des problèmes dans divers domaines.

III.2. Méthodes déterministes :

Sont des algorithmes qui ne font pas intervenir le hasard dans la recherche de l'optimum, et généralement, elles nécessitent le choix d'un point de départ et/ou la connaissance de la fonction-objectif pour explorer son gradient. Ce type d'optimisation regroupe : les heuristiques, les méthodes locales, les méthodes globales classiques qui

s'appuient sur le gradient ainsi que beaucoup de méthodes spécialisées comme le branch & bound, ... etc.

III.3. Méthodes stochastiques :

Contrairement aux méthodes déterministes ces méthodes font des explorations aléatoires et reposent sur des processus stochastiques, cependant elles font appel à des propriétés probabilistes dans la recherche d'optimum. Elles sont moins performantes (du point de vue rapidité) que les méthodes déterministes, mais elles peuvent trouver, en principe, un optimum global difficile à atteindre. Le grand avantage de ces méthodes réside dans le peu d'hypothèses qu'on exige sur la fonction-objectif et le non spécificité de l'instance du départ. Généralement, ces méthodes sont des méta-heuristiques distribuées.

III.3.1. Méthodes Métaheuristiques :

Sont des méthodes polyvalentes (applicables à de nombreux problèmes) et en majorité s'inspirent des systèmes naturels. Elles ont une grande capacité à extraire l'optimum global mais demandent un coût de recherche important.

Parmi les méta-heuristiques, on peut différencier les méta-heuristiques "de voisinage", qui font progresser une seule solution à la fois (recuit simulé, recherche tabou, etc.), et les méta-heuristiques "distribuées", qui manipulent en parallèle toute une population de solutions (algorithmes génétiques, etc.) et essaient d'améliorer leurs qualités au cours des itérations.

Ces méthodes sont d'origine combinatoire, mais avec le besoin, elles ont été adaptées au cas continu. Cependant, cette extension ou adaptation, la plus courante en ingénierie, fait recours à une stratégie de discrétisation des variables continues. Le pas de discrétisation doit s'adapter en cours d'optimisation, pour garantir à la fois la régularité de la progression vers l'optimum et la précision du résultat.

Parmi les méthodes stochastiques les plus utilisées, on trouve les Algorithmes Génétiques et qu'on abordera en détail dans la partie suivante.

IV. Algorithmes Génétiques :

Entre les années 1950 et 1960 les chercheurs ont commencé à exploiter le mécanisme de l'évolution pour proposer des techniques, des stratégies et des algorithmes capable de résoudre des problèmes complexes. Dans les années 1960, un chercheur Jean Holland a fondé les bases des algorithmes génétiques qui sont une technique stochastique inspirée par la sélection naturelle de Charles Darwin mais qui est capable à nous offrir des solutions optimales à nos problèmes d'optimisation.

IV.1. Qu'est ce qu'un algorithme génétique ?

Les algorithmes génétiques (AGs) sont des algorithmes d'optimisation stochastique leur fonctionnement est extrêmement simple. On part avec une population de solutions potentielles (chromosomes) initiales arbitrairement choisies. On évalue leur performance (fitness) relative. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. On recommence ce cycle jusqu'à ce que l'on trouve une solution satisfaisante.

IV.2. Principe des AGs :

Le principe des AGs est schématisé sur la figure 3.2 :

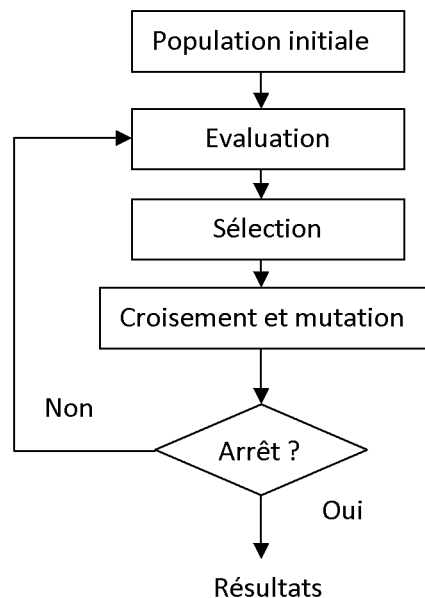


Figure 3.2. Organigramme des algorithmes génétiques.

IV.3. Fonctionnement des AGs :

Avant d'expliquer en détail le fonctionnement d'un algorithme génétique, nous allons présenter quelques mots clés utilisés dans un algorithme génétique.

- **Gène** : un chromosome est composé de gènes. On binaire un gène se code par 0 ou 1.
- **Génotype** ou **chromosome** : c'est un ensemble de gènes, et c'est une autre façon de dire "individu".
- **Individu** : correspond au codage sous forme de gènes d'une solution potentielle à un problème d'optimisation.
- **Une population** : elle est représentée par un ensemble d'individus (i.e. l'ensemble des solutions du problème).
- **Une génération** : est une succession d'itérations composées d'un ensemble d'opérations, permettant le passage d'une population à une autre.
- **La probabilité de croisement**: Elle représente la probabilité d'échange de patrimoine entre deux individus (ou plus). Plus elle est grande, plus la génération des nouveaux descendants est meilleurs que leurs parents.
- **La probabilité de mutation**: Elle est en général faible, Il a de fait un double rôle : celui d'effectuer une recherche locale et/ou de sortir d'une trappe (recherche éloignée).

IV.3.1. Population initiale :

La population initiale d'un AG représente la population de départ de l'algorithme. Si la position de l'optimum dans l'espace de solutions est totalement inconnue, il est naturel de faire une génération aléatoire des individus. Si, en revanche, des solutions sont disponibles, il paraît évident de générer les individus à partir de ces solutions ; mais le problème qui se pose c'est comment choisir la taille m de cette population, de telle façon qu'on puisse satisfaire les deux critères, c'est-à-dire, qu'elle sera hétérogènes et qu'elle ne prenne pas beaucoup de temps de calcul. La seule contrainte concernant cette population initiale est qu'elle doit appartenir au domaine des solutions admissibles.

IV.3.2. Codage et population initiale :

Les algorithmes génétiques utilisent le codage des paramètres, et non les paramètres eux-mêmes. Chaque variable de décision sera codée sous forme d'une chaîne de symboles

appelée chromosome, on distingue trois types principaux : binaire, *gray* ou réel, le codage binaire est le plus utilisé, chaque gène dispose du même alphabet binaire (0, 1).

IV.3.3. Fonction d'évaluation :

La fonction d'évaluation, ou encore fitness, quantifie la qualité de chaque chromosome par rapport au problème. Elle est généralement utilisée pour sélectionner les chromosomes pour la reproduction. Les chromosomes ayant une bonne qualité ont alors plus de chances d'être sélectionnés pour la reproduction, faisant en sorte que la prochaine génération de la population hérite de leur matériel génétique. La fonction d'évaluation produit la pression qui permet de faire évoluer la population de l'AG vers des individus de meilleure qualité.

IV.3.4. Opérateurs des algorithmes génétiques :

On distingue trois opérateurs :

IV.3.4.1. Opérateur de sélection

L'opérateur de sélection tend à augmenter l'importance de l'utilisation des solutions de bonne qualité pour la génération de descendants. Une pression de sélection est généralement associée à un opérateur de sélection. Elle représente un terme informel pour décrire la force du mécanisme de la phase de sélection. Cette pression calcule le ratio de la meilleure fitness par rapport à la moyenne de la population.

$$p_i = \frac{f(x_i)}{\sum_{i=1}^m f(x_i)} \quad (3.8)$$

Ainsi on choisit un nombre $\alpha \in [0,1]$. Si $f(x_i) > \alpha$ sélectionner x_i , à partir des éléments sélectionnés, on forme une nouvelle population dont la taille est la même que celle de la population initiale (m).

IV.3.4.2. Opérateur de Croisement :

Le croisement permet la création de nouveaux individus selon un processus fort simple. Il permet donc l'échange d'information entre les chromosomes (individus). Tout d'abord, deux individus, qui forment un couple, sont tirés au sein de la nouvelle population issue de la reproduction. Puis un (potentiellement plusieurs) point de croisement est tiré aléatoirement. Enfin, selon une probabilité p_c que le croisement s'effectue, les segments

finiaux (dans le cas d'un seul point de croisement) des deux parents sont alors échangés autour de ce point (voir la figure 3.3).

Exemple 3.1

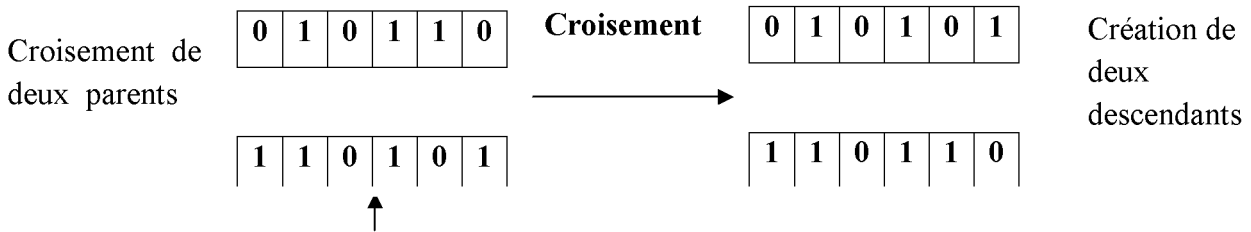


Figure 3.3. Le croisement en codage binaire

IV.3.4.3. Operateur de mutation :

Avant de finir, on peut procéder à une mutation au niveau des gènes d'un individu. Pour cela, on commence par choisir au hasard un certain nombre d'individus (en général, la probabilité de sélection pour la mutation est faible de l'ordre de 0.001). Ensuite, pour chaque individu sélectionné, on choisit au hasard le gène où aura lieu la mutation. Finalement, à cette position, on change le 0 en 1 et réciproquement.

Remarque 3.4

On recommence cette procédure jusqu'à ce que l'on atteigne l'un des critères d'arrêt.

IV.3.5. Condition d'arrêt des AGs :

Les conditions d'arrêt peuvent être :

- Un nombre de génération au préalable ;
- Une valeur de la fonction d'évaluation à ne pas dépasser ;
- Un nombre de génération maximal sans améliorer la solution ;
- L'obtention de chromosomes ayant atteint un certain degré d'homogénéité (un grand nombre de chromosomes possèdent des gènes identiques aux mêmes positions).

IV.3.6. Création d'une nouvelle population :

Dans cette dernière étape, les nouvelles solutions sont ajoutées à la population courante en remplacement (total ou partiel) des anciennes solutions (parents). Généralement, les meilleures solutions remplacent les plus mauvaises, et il en résulte une amélioration de la

population. Lorsque la nouvelle population n'est constituée que des nouvelles solutions (enfants), l'AG est dit *générationnel*.

IV.4. Avantage et inconvénient

Les algorithmes génétiques se basent sur une recherche globale ce qui leur permet d'échapper au problème de la convergence vers l'optimum local et augmente leur possibilité de fournir des solutions de bonnes qualités. Son avantage principal est qu'elles permettent une bonne combinaison entre l'exploitation des solutions et de l'espace de recherche, et elles peuvent être appliquées quelque-soit le cas traité, éventuellement, on peut les considérer comme une boîte noire. Malheureusement, de nombreux inconvénients s'imposent, en effet, il n'y a pas de garantie quant à l'obtention de la solution optimale au problème posé en un temps fini. L'utilisation de ces algorithmes est souvent coûteuse en temps de calculs (lentes).

Exemple illustratif 3.2

Considérons le problème de maximisation suivant :

$$\max_x \quad f(x) = x^2 \quad (3.9)$$

Sujet à :

$$1 \leq x \leq 31 \quad (3.10)$$

1/ Codage du problème :

Chaque individu représentera une valeur de x . Son génotype sera la valeur de x exprimée en binaire, alors il suffit de 5 bits, car 31 se note 11111.

2/ Population initiale :

On choisit ici une population de 4 individus. On détermine la population initiale en tirant au hasard la valeur de chaque allèle. Supposons que l'on obtienne les individus suivant :

Numéro de l'individu	Génotype					Phénotype
1	0	1	1	1	0	14
2	1	1	1	0	1	29
3	0	1	0	0	1	9
4	1	0	0	0	0	16

Tableau 3.1 Création de la population initiale.

3/ Evaluation de l'adaptation :

Dans ce cas, la fonction d'adaptation (objectif) est simple. Elle s'agit de la fonction :

$$f(x) = x^2 \quad (3.11)$$

Ainsi pour calculer l'adaptation du premier individu, il suffit de calculer le carré de 14. On obtient ainsi :

Numéro de l'individu	Génotype					Phénotype	Adptation	$p_i(\alpha = 0.1)$
1	0	1	1	1	0	14	196	0.1426
2	1	1	1	0	1	29	841	0.6120
3	0	1	0	0	1	9	81	0.0589
4	1	0	0	0	0	16	256	0.1863

Tableau 3.2. Calcul de la fonction d'adaptation.

4/ Sélection des parents

Il suffit de tirer au hasard 4 individus parmi la population en tenant compte de leurs adaptations respectives. On obtient, ainsi, une nouvelle population comprenant par exemple :

- Une copie de l'individu 1.
- Une copie de l'individu 4.
- deux copies de l'individu 2.
- aucune copie de l'individu 3.

C'est-à-dire :

Numéro d'individu	Génotype					Phénotype	Adaptation
1	0	1	1	1	0	14	196
2	1	1	1	0	1	29	841
2	1	1	1	0	1	29	841
3	1	0	0	0	0	16	256

Tableau 3.3. Sélection des individus.

5/ Recombinanson(croisement)

On choisit ici, pour le croisement de déterminer au hasard les deux parents, et de couper les chromosomes au hasard : la première partie du chromosome ira au premier descendant, alors que la seconde ira à l'autre. Supposons que l'individu 1 se recombine avec l'individu 2 et l'individu 3 avec 4 (on note || les point de croisement).

Numéro d'individu	Avant					Après				
1	0	1	1	1	0	0	1	1	0	1
2	1	1	1	0	1	1	1	1	1	0
3	1	1	1	0	1	1	1	1	0	0
4	1	0	0	0	0	1	0	0	0	1

Tableau 3.4. Croisement des individus.

6/ Seconde génération

Dans cet exemple, on choisit de limiter la durée de vie de chaque individu à une génération. Donc, la nouvelle génération sera composée exclusivement des enfants :

Numéro d'individu	Génotype					phénotype	Adaptation	$p_i(\alpha = 0.1)$
1	0	1	1	0	1	13	169	0.0788
2	1	1	1	1	0	30	900	0.4201
3	1	1	1	0	0	28	784	0.3660
4	1	0	0	0	1	17	289	0.1349

Tableau 3.5. Reproduction d'une nouvelle génération.

Si on calcul la moyenne des valeurs d'adaptation, on choisit 344 pour la génération initiale, contre 1926 pour la seconde. On se rapproche bien de la solution. Si on réitère le processus, on obtiendra des valeurs de plus en plus grandes, jusqu'à l'obtention de la solution.

Remarque 3.5

En pratique, le choix d'un algorithme n'est pas toujours un problème simple. Car il ne consiste pas seulement en la recherche d'un l'algorithme convenant au type de problème et donnant la meilleure précision sur les résultats obtenus, mais aussi, consiste en celui qui minimise l'encombrement mémoire et le temps de calcul.

V. Optimisation en présence d'incertitude.

Sur le plan pratique, il est difficile de trouver des exemples de systèmes à optimiser qui ne sont pas complexes et qui ne comprennent pas un certain niveau d'incertitude sur les valeurs d'au moins quelques paramètres, alors d'autre types de problèmes d'optimisation en présence d'incertitudes sont apparus, et qui sont connu par l'optimisation robuste.

V.1. Optimisation Robuste :

L'*optimisation robuste* est une nouvelle technique qui traite les problèmes d'optimisation soumis à l'incertitude. La première formulation d'une solution robuste à un problème de programmation mathématique à paramètres incertains peut être attribuée à Soyster au début des années 70, mais le domaine a pris son essor beaucoup plus tard, à la fin des années 90, grâce au travail des pionniers de domaine Ben-Tal et Nemirovski, mais également d'El-Gahoui et Lebret. Depuis, l'optimisation robuste est devenue un des domaines de recherche les plus actifs en optimisation.

Cette technique vise à fournir une solution aux problèmes de cette classe, en des temps de calcul raisonnables, là où les approches traditionnelles, basées sur la formulation d'un modèle probabiliste, se heurtent à des difficultés numériques considérables, voire, même l'impossibilité de cette formulation.

Pour contourner ces difficultés, l'optimisation robuste suppose des hypothèses minimales sur le domaine de variation des paramètres incertains appelé ensemble d'incertitude Δ . Ainsi le calcul de modèle probabiliste est évacué, et remplacé par la notion de «pire des cas» traduite par la notion « minimax ».

La formulation générale de problème d'optimisation robuste est la suivante :

$$\min_{x \in \Omega} \max_{\zeta \in \Delta} f_{\zeta}(x) \quad (3.12) \quad (P_1)$$

Où : ζ : vecteur des paramètres incertains

Δ : représente l'ensemble des incertitudes

Malgré que la considération de la notion de 'pire des cas' nous a ramené à éviter des structures de modèle complexes, la complexité de résolution réside encore du fait que le modèle obtenu devient de type semi-infini.

V.2. Qu'est-ce- qu'une optimisation semi-infinie ?

L'optimisation semi-infinie, connue par la famille des problèmes *SIP* (de la langue anglaise *Semi-Infinite Programming*), est l'appellation donnée aux types de problèmes d'optimisation à un nombre fini de variables de décision et à nombre infini de contraintes.

La forme générale d'un *SIP* est donnée par le problème (P_2) suivant :

$$\text{Optimum}_{x \in \Omega \subset \mathbb{R}^n} f(x) \quad (3.13)$$

Soumise à : (P_2)

$$h_j(x, \zeta) \leq 0 \quad (3.14)$$

Même si la fonction-objectif ne présente aucune structure complexe, comme la non linéarité, ce type de problème est difficile à résoudre, et souvent, de type NP-difficile vu l'infinité de contrainte qu'il présente.

Remarque 3.6

- Un problème à nombre infini de variables de décision et à nombre fini de contrainte fait, aussi, parti de la famille des *SIP*.
- La fonction $f_{\zeta}(x)$ et $f(x, \zeta)$ sont équivalentes.
- Le problème (P₂) peut être réécrit sous la forme (P₁) comme suit :

$$\min_{x, \zeta \in \Delta} \quad R \quad (3.15) \tag{P_3}$$

Soumise à :

$$l_{\zeta}(x) \leq 0 \quad \forall \zeta \in \Delta \quad (3.16)$$

Avec : $R = \max_{\zeta \in \Delta} f_{\zeta}(x) \quad (3.17)$

$$l_{\zeta}(x) = f_{\zeta}(x) - R \quad (3.18)$$

Ce problème est très difficile à traiter vu le nombre infini de contraintes imposées par les incertitudes. Plusieurs travaux de recherche ont été menés pour surmonter cette difficulté, et l'approche des scénarios fait partie des méthodes proposées.

VI. Approche des scénarios :

La méthode des scénarios est le fruit des recherches menées par Giuseppe Calafiore et Marco C. Campi (et qui a été publiée en 2005), dans un but basique, surmonter la complexité due aux contraintes qu'imposent les problèmes d'optimisation convexes semi-infinie.

Cette méthode suppose une description probabiliste des incertitudes appartenant à un ensemble. La méthode repose sur un échantillonnage aléatoire de contraintes, et représente un outil puissant pour résoudre une variété de problèmes de conception rencontrés dans la théorie des systèmes et de commande.

VI.1. Fondement de la méthode :

L'idée principale de la méthode s'appuie sur la réduction de nombre infini des contraintes à un nombre fini, lors de la recherche de la solution, et cela en extrayant

aléatoirement N instances ou « scénarios » des paramètres incertains ζ . Cette extraction est faite en se basant sur une probabilité de distribution P_r des incertitudes.

VI.2. Optimisation par l'approche des scénarios :

Après formulation du problème à optimiser sous forme d'un problème robuste (P_3) dont le nombre de contraintes est infinie, on extrait N échantillons (des paramètres incertains) indépendants et identiquement distribués (iid) $(\zeta^{(1)}, \dots, \zeta^{(N)})$ selon la probabilité P_r , puis on résout le problème d'optimisation suivant :

$$SP_N: \quad \min_x \quad R \quad (3.19)$$

s.à :

$$f_{\zeta^{(i)}}(x) \leq 0 \quad i = 1, \dots, N \quad (3.20)$$

Du fait de cette procédure d'échantillonnage, seulement les contraintes correspondantes aux paramètres incertains $\zeta^{(i)}$ extrais seront considérées. Par conséquent, le problème d'optimisation robuste dont le nombre de contraintes est infini devient un problème d'optimisation avec un nombre de contraintes fini.

Cependant, deux questions évidentes viennent à l'esprit et qui sont :

- Comment choisir la valeur de nombre d'échantillons N ?
- Quel degré de robustesse présente la solution trouvée, puisque la procédure de sa recherche était restreinte à un nombre fini de contraintes, tandis que les autres étaient ignorées ?

Le théorème suivant répond à ces questions.

Théorème 3.1 : [11]

Fixez un « paramètre de risque » $\epsilon \in [0,1]$ et un « paramètre de confiance » $\beta \in [0,1]$

Si
$$N \geq \frac{2}{\epsilon} \left(\ln \frac{1}{\beta} + n \right) \quad (3.21)$$

Alors, avec une probabilité $Pr \geq 1 - \beta$, la solution x_N^* satisfait toutes les contraintes sur Ω mais sauf pour ϵ -fraction.

Avec : n représente le nombre de variables de décision.

Pour conclure, la méthode des scénarios est une nouvelle technique innovante, simple à utiliser et propose divers champs d'application (control robuste, stabilisation robuste, conception de la commande robuste H_2 , réduction du modèle,...) et son utilisation se résume en ces trois points essentiels :

- Formulation de problème d'optimisation comme un problème d'optimisation robuste avec une infinité de contraintes.
- Echantillonnage aléatoire des contraintes et résolution du problème d'optimisation, à nombre de contraintes fini, par des méthodes d'optimisation classique.
- Evaluation de niveau de satisfaction de la solution obtenu pour toutes les contraintes avec le théorème 3.1.

VII. Génération des nombre aléatoire :

Comme, on le constate soit dans les algorithmes génétiques ou dans la méthode des scénarios, l'utilisation des variables aléatoires est une étape évidente, cependant, l'implémentation de tels algorithmes sur ordinateur requiert une génération de ces derniers.

VII.1. Nombre aléatoire [11]

Une variable aléatoire est une variable dont la suite des valeurs n'est pas prédictible. Aussi longue que soit la série générée, il impossible de trouver une équation permettant de prédire le reste de la série à partir de celles déjà générées.

Une suite de nombres aléatoires est un échantillon issu d'une population distribuée selon la loi statique dont les éléments sont indépendants les uns des autres. Ils peuvent être engendrés par traitement d'une source d'entropie extérieure (par exemple un bruit).

VII.2. Propriétés des nombres aléatoires :

Considérons une séquence de N nombres x_i tirés à partir d'une distribution uniforme de la variable aléatoire x sur $[0, 1]$. Cette séquence de nombres doit satisfaire deux propriétés :

- **Uniformité** : ils sont répartis sur $[0, 1]$.
- **Indépendance** : la valeur x_i ne doit avoir aucune dépendance par rapport à la valeur x_{i-1}

Comme nous l'avons vu précédemment, le hasard est une notion relativement difficile à définir. La nécessité de produire de longues séries de nombres simulant le hasard, tout en ayant d'autres qualités, comme une répartition aussi uniforme que possible est une préoccupation permanente des informaticiens, pour cela nous avons eu le recours à la méthode de Monte Carlo pour garantir ces besoins recherchés.

VII.3. Méthode de Monte Carlo : [11], [8]

Cette méthode a été fondée par le mathématicien américain Stanisław Ulam plus précisément en 1949, le nom de Monte-Carlo fait explicitement référence aux casinos de Monte-Carlo et aux jeux de hasard, ce qui a simulé beaucoup cette méthode et le projet de Manhattan, qui a vu la naissance de la bombe atomique, l'idée c'était de pouvoir simuler des réactions en chaîne avec ces algorithmes au lieu de faire des essais avec des vraies bombes. Et elle désigne toute méthode de calcul basée sur l'utilisation des nombres aléatoires.

VII.3.1. Principe de la méthode de MC :

La méthode de Monte Carlo est utilisée dans de nombreux cadres, cette dernière a pour but de simuler des systèmes déterministes avec des paramètres ou des entrées stochastiques. Elle s'appuie sur l'échantillonnage des distributions et des quantités incertaines, autrement dit sur la capacité de générer des nombres aléatoires.

VII.3.2. Concept de la méthode de MC [8]

La méthode de Monte Carlo nécessite à la base de définir un générateur de nombres aléatoire uniforme à valeur réelles entre 0 et 1. A ce stade, deux remarques s'imposent concernant le caractère aléatoire des nombres générés, et la corrélation entre valeurs générées.

De façon générale, il est très important d'être prudent avec les générateurs intégrés ; certains peuvent en effet présenter des périodes courtes, ce qui génère des corrélations entre les variables. Toute la difficulté est donc de trouver un générateur offrant des valeurs suffisamment aléatoire et de période la plus grande possible.

L'utilisation d'un ordinateur pour générer des nombres aléatoires peut paraître en effet déconcertante, n'importe quel programme informatique ne peut exécuter une suite d'instructions seulement si cette dernière est décrite à l'avance, dans ce contexte le générateur sera forcément déterministe et non véritablement aléatoire, on parlera alors d'une génération de variables « pseudo-aléatoire ».

VII.4. Générateur de nombre pseudo- aléatoire :

Formellement, c'est un dispositif qui génère une suite de nombres pseudo-aléatoires, possédant les mêmes propriétés qu'une suite de nombres aléatoires, mais générée à travers une procédure déterministe. Ils sont générés par des fonctions mathématiques ou par des algorithmes. La plupart des algorithmes pour la génération de nombres pseudo-aléatoires ont pour but de produire des suites uniformément distribuées et qui puisse apparaître comme non déterministe. Le générateur, le plus couramment, utilisé est celui des congruences linéaires inventé par D. H Lehmer (1948).

VII.4.1. Générateur congruentiel linéaire (LCG) :

Le générateur LCG génère une séquence x_η de nombre entier non négatifs grâce à la formule de récurrence suivante (aussi connue par l'algorithme de Lehmer) [8, 11].

$$x_\eta = (a_m x_{\eta-1} + c_i) \text{ mod } m_t \quad \forall \eta \geq 1 \quad (3.23)$$

Où : $\text{mod } m_t$ est le reste de division de $(a_m x_{\eta-1} + c_i)$ par m_t . m_t est un entier positif appelé terme et a_m , c_i et x_0 sont appelés respectivement multiplicateur, incrément et germe.

La suite de nombres aléatoire uniforme est donné par la suite suivante :

$$u_\eta = \frac{x_\eta}{m_t} \quad \forall \eta \geq 0 \quad (3.24)$$

Exemple 3.3.

Choisissons comme titre d'exemple les paramètres a_m , c_i et m_t comme suit :

$$x_\eta = (25 x_{\eta-1} + 16) \bmod 256 \quad (3.25)$$

Pour $x_0 = 12$, les nombres produits sont : 60, 236, 28, 204, 252, 172

Ils sont tous pairs !

Pour $x_0 = 11$, les nombres produits sont : 35, 123, 19, 235, 3, 91, 243. Ils sont tous impair !

Pour $x_0 = 10$, les nombres produits sont : 10, 10, 10, 10 cela produit toujours la même valeur.

Remarquons que la formule est simple mais le choix des trois paramètres doit répondre à certains critères, car un mauvais choix de ces paramètres peut rendre le générateur complètement trivial. Nous présentons ci-dessous l'un des critères les plus utilisés, celui de Knuth.

VII.4.1.1. Critère de Knuth :

Knuth fait la démonstration des critères que doivent remplir les paramètres : a_m , c_i et m_t pour cela :

- c_i et m_t doivent être premiers entre eux ;
- $a_m - 1$ doit être un multiple de N_p , pour tout N_p nombre premier diviseur de m_t ;
- $a_m - 1$ doit être un multiple de 4 si m_t est un multiple de 4.
- si m_t est une puissance de 2, le bit de poids faible des nombres produits vaut alternativement 0 et 1.

Remarque 3.7

- Si on désire produire toujours la même séquence, on va utiliser la même valeur de x_0 .
- Si on préfère que la séquence soit toujours différente, on initialise x_0 avec une valeur différente à chaque fois qu'on refait la génération. Dans tous les cas, les nombres de la suite sont compris entre 0 et $m_t - 1$.

VII.4.2. Avantage de l'approche pseudo-aléatoire :

L'approche pseudo-aléatoire est souvent préférable puisque elle est plus rapide, facile à générer et permet la contrôlabilité et la répétitivité de l'expérience si nécessaire (par exemple pour des raisons de débogage).

VIII. Conclusion :

Dans ce chapitre nous avons présenté des généralités sur l'optimisation statique. Nous avons aussi introduit la notion de l'optimisation robuste, c'est-à-dire en présence des incertitudes. Un problème d'optimisation robuste est caractérisé par un nombre de contraintes infinies. Ce problème peut être relaxé en utilisant l'approche des scénarios qui permet de rendre le nombre de contraintes fini.

Dans le chapitre suivant, on s'intéressera à l'utilisation de méthode des scénarios dans la synthèse d'un correcteur robuste pour un système ARMA

Chapitre 04

Synthèse des correcteurs robustes par la méthode des scénarios.

I. Introduction :

Dans ce chapitre l'approche des scénarios est adoptée pour la conception d'un correcteur robuste assurant une variance nulle pour la sortie d'un système incertain sujet à une entrée aléatoire. Le problème de conception est formulé sous forme d'un problème d'optimisation robuste résolu par les algorithmes génétiques. Pour illustrer les performances du correcteur robuste, une étude comparative des résultats obtenus avec ce dernier et ceux obtenus avec un correcteur nominal est présentée.

II. Structure ARMA (Auto-Regressive Moving-Average):

Les modèles ARMA sont une généralisation des modèle MA et AR ils consistent à avoir une partie autorégressive et une partie à moyenne mobile [24].

- **Le modèle AR :**

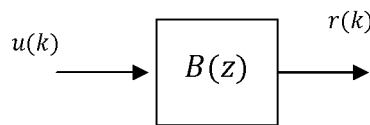


Figure 4.1. Modèle de structure AR.

- **Le modèle MA :**

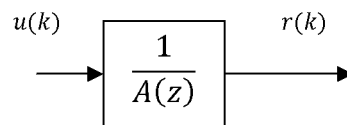


Figure 4.2. Modèle de structure MA.

- **Le modèle ARMA :**

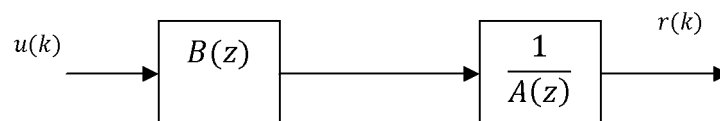


Figure 4.3. Modèle de structure ARMA.

On a :

$$r(k) = \frac{B(z)}{A(z)} u(k). \quad (4.1)$$

Supposant que le système dynamique est caractérisé par la relation entrée-sortie :

$$r(k) = z^{-d} \frac{B(z^{-1})}{A(z^{-1})} u(k) \quad (4.2)$$

Où : A et B sont des polynômes en z^{-1} , z^{-1} étant l'opérateur retard défini par

$$z^{-1} u(k) = u(k - 1) \quad (4.3)$$

Le signal $r(k)$ désigne la sortie non bruitée du système. $u(k)$ est la commande et d le retard de l'entrée par rapport à la sortie avec $d \geq 1$. Par ailleurs, les polynômes $A(z^{-1})$ et $B(z^{-1})$ sont définies par:

$$A(z^{-1}) = 1 + \sum_{i=1}^{n_i} a_i z^{-i} = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_i} z^{-n_i} \quad i = 1, \dots, n_i \quad (4.4)$$

$$B(z^{-1}) = \sum_{j=0}^{m_j} b_j z^{-j} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{m_j} z^{-m_j} \quad j = 1, \dots, m_j \quad (4.5)$$

Les perturbations agissant sur le système se réduisent à un seul signal $v(k)$ ce qui donne pour la sortie du système l'expression suivante :

$$y(k) = r(k) + v(k) \quad (4.6)$$

En outre la perturbation $v(t)$ est considérée comme étant la sortie d'un filtre excité par un bruit blanc comme suit :

$$v(k) = \frac{C(z^{-1})}{A(z^{-1})} w(k) \quad (4.7)$$

Sachant bien que l'opérateur $C(z)$ s'écrit sous la forme suivante :

$$C(z) = \sum_{p=1}^{c_p} c_p z^{-p} = c_1 z^{-1} + c_2 z^{-2} + \dots + c_p z^{-p} \quad p = 1, \dots, c_p \quad (4.8)$$

Où $w(k)$ est une séquence indépendante et non corrélée, de moyenne nulle et de variance finie. Une écriture standard du modèle du système est alors la suivante :

$$A(z^{-1}) y(k) = z^{-1} B(z^{-1}) u(k) + C(z^{-1}) w(k) \quad (4.9)$$

Cette forme est connue dans la littérature sous le nom de modèle Auto-Régressive Moving Average ou modèle ARMA. Il faut rappeler que les statisticiens désignent par modèle ARMA le modèle où la commande $u(k)$ est nulle et l'entrée se réduit à un processus aléatoire.

III. Application de l'Approche des Scénarios pour la synthèse d'un correcteur robuste pour un système ARMA :

Considérons le système le système Autorégressif à moyenne mobile (*Auto-Regressif Moving-Average* _ ARMA) représenté sous sa forme récurrente (équation aux différences) suivant :

$$y(k + 1) = a y(k) + b u(k) + c_1 w(k) + c_2 w(k - 1) \quad (4.10)$$

Où : $u(k)$ est l'entrée du système, $y(k)$ est la sortie et $w(k)$ est la perturbation représenté par un bruit blanc avec une moyenne nulle et une variance unitaire $WN(0,1)$, et a, b, c_1 et c_2 sont des paramètres réels {avec $|a| < 1$ (condition de stabilité) et $b \neq 0$ (condition de contrôlabilité)}, qui ne sont pas connus avec précision (paramètres incertains).

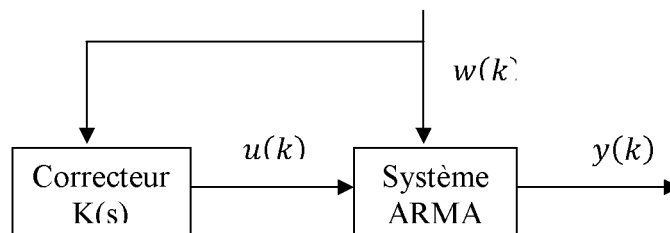


Figure 4.4. Schéma du système de commande.

Notre objectif est de concevoir un correcteur en boucle ouverte de structure :

$$u(t) = K_1 w(k) + K_2 w(k - 1) \quad (4.11)$$

Avec K_1 et K_2 les paramètres du correcteur, et pour cela, nous supposons que $w(k)$ sont mesurables et le correcteur sera calculé par la minimisation de la variance asymptotique de $y(k)$, c'est-à-dire :

$$\min_{K_1, K_2} E [y_t^2] \quad (4.12)$$

$$E[y^2(k)] = \frac{((c_1 + bK_1)^2 + (c_2 + bK_2)^2 + 2a(c_1 + bK_1)(c_2 + bK_2))}{1 - a_1^2} \quad (4.13)$$

III.1 Synthèse du correcteur nominal :

Si les conditions sont nominales, alors les paramètres du correcteur K_1 et K_2 sont facilement obtenus pour :

$$E[y^2(k)] = 0 \quad (4.14)$$

On obtient : $K_1 = -\frac{c_1}{b}$ et $K_2 = -\frac{c_2}{b}$

Finalement, le correcteur nominal est donné par la loi de commande suivante :

$$u(k) = -\frac{c_1}{b} w(k) + -\frac{c_2}{b} w(k - 1) \quad (4.15)$$

Pour les valeurs nominales $a^{nom} = 0,45, b^{nom} = 1, c_1^{nom} = 0,2, c_2^{nom} = 0,5$ on aura :

$$u(k) = -0,2w(k) + -0,5 w(k - 1) \quad (4.16)$$

III.2. Synthèse du correcteur robuste :

Chapitre 04 : Synthèse d'un correcteur robuste par l'Approche des Scénarios

Dans ce cas, on suppose que les paramètres du système sont incertains. Ainsi, les paramètres ne sont que partiellement connus, et ils prennent la valeur dans un ensemble Δ d'incertitude bien déterminé. L'objectif est de concevoir un correcteur K_1 et K_2 assurant de bonne performances malgré les incertitudes.

Dans notre problème, l'ensemble de l'incertitude Δ est définie par le couple $(\theta_1, \theta_2) \in [-1/3, 1/3]^2$ comme suit:

$$\Delta = \{a, b, c_1, c_2 : a = a^{nom} + 0,5 \cdot (1 - e^{-8 \cdot 10^3 (\theta_1^2 + \theta_2^2)}); \quad (4.17)$$

$$b = b^{nom} + \theta_2^2; \quad (4.18)$$

$$c_1 = c_1^{nom} + (\theta_2 + \sin(\theta_2) + 0,1) \cdot \sin(2\pi\theta_2); \quad (4.19)$$

$$c_2 = c_2^{nom} + \theta_1^2 \cos(\theta_2); \quad (4.20)$$

$$(\theta_1, \theta_2) \in [-\frac{1}{3}, \frac{1}{3}]^2 \quad \} \quad (4.21)$$

Où :

$$a^{nom} = 0,45, \quad b^{nom} = 1, \quad c_1^{nom} = 0,2, \quad c_2^{nom} = 0,5$$

Dans ce cas, le problème de synthèse peut être formulé comme un problème d'optimisation minimax comme suit :

$$\min_{K_1, K_2} \max_{a, b, c_1, c_2 \in \Delta} E [y^2(k)] \quad (4.22)$$

En considérant : $l_{a, b, c_1, c_2}(K_1, K_2) = E [y^2(k)] \quad (4.23)$

Le problème d'optimisation prend la forme :

$$\min_{K_1, K_2} \max_{a, b, c_1, c_2 \in \Delta} l_{a, b, c_1, c_2}(K_1, K_2) \quad (4.24)$$

La formulation min max permet de minimiser la variance de la sortie quelle que soit l'incertitude sur les paramètres du système.

Chapitre 04 : Synthèse d'un correcteur robuste par l'Approche des Scénarios

Donc pour synthétiser le correcteur robuste, il suffit de résoudre le problème d'optimisation (4.24) en calculant les paramètres K_1 et K_2 . Pour se faire on utilisera l'approche des scénarios.

L'application de cette approche se résume comme suit :

Etape 01 : La formulation du problème (4.24) sous forme d'un problème d'optimisation à dont le nombre de contraintes est infini. La procédure est la suivante :

$$\text{Posons : } K_3 = \max_{a,b,c_1,c_2 \in \Delta} l_{a,b,c_1,c_2}(K_1, K_2) \quad (4.25)$$

Puisque K_3 est la valeur maximale, il en résulte que toutes les contraintes satisfont la propriété suivante :

$$l_{a,b,c_1,c_2}(K_1, K_2) \leq K_3 \quad (4.26)$$

$$\text{La relation (4.26) est équivalente à : } l_{a,b,c_1,c_2}(K_1, K_2) - K_3 \leq 0 \quad (4.27)$$

Notons que K_3 constitue un autre inconnu du problème.

Alors, le problème d'optimisation robuste (4.24) s'écrit comme suit :

$$\begin{aligned} \min_{K_1, K_2, K_3} \quad & K_3 \\ \text{Sujet à :} \quad & \\ & l_{a,b,c_1,c_2}(K_1, K_2) - K_3 \leq 0 \end{aligned} \quad (4.28)$$

Remarque : Le paramètre K_3 est un paramètre qui aide à la résolution du problème (4.28) mais qui n'a pas du sens physique. (C'est-à-dire il n'est pas un paramètre du correcteur).

Etape 02 : échantillonnage aléatoire des contraintes et résolution du problème.

Cette étape suppose une discrétisation aléatoire des contraintes, autrement, échantillonner l'ensemble des incertitudes Δ .

Chapitre 04 : Synthèse d'un correcteur robuste par l'Approche des Scénarios

Le nombre d'échantillons est prédit par la relation :

$$N \geq \frac{2}{\epsilon} \left(\ln \frac{1}{\beta} + n \right) \quad (4.29)$$

Avec :

Paramètre de risque : $\epsilon = 0.01$

Paramètre de confiance : $\beta = 10^{-10}$

Nombre de variable de décision (K_1, K_2, K_3): $n = 3$

Ces paramètres donnent $N = \frac{2}{0.01} \left(\ln \frac{1}{10^{-10}} + 3 \right) = 5205.17018 \Rightarrow N = 5206$ (4.30)

Donc pour avoir un correcteur robuste, on considère 5206 valeurs aléatoires pour les incertitudes, c'est-à-dire 5206 contraintes. Le problème d'optimisation à résoudre est :

$$\min_{K_1, K_2, K_3} K_3$$

Sujet à : (4.31)

$$l_{a^{(i)}, b^{(i)}, c_1^{(i)}, c_2^{(i)}}(K_1, K_2) - K_3 \leq 0 \quad i = 1, \dots, 5206$$

Les échantillons générés doivent être indépendants et identiquement distribués, de plus, on doit générer un grand nombre de variable aléatoires ($N = 5206$) dans un petit intervalle $(\theta_1, \theta_2) \in \left[-\frac{1}{3}, \frac{1}{3}\right]^2$.

Le problème (4.31) est un problème d'optimisation dont la fonction objectif est linéaire et les contraintes sont quadratiques. Ce problème d'optimisation a été résolu avec Matlab en utilisant les Algorithmes Génétiques qui ont la capacité de converger vers un optimum global. Les résultats obtenus sont :

$$K_1^* = -0.5186$$

$$K_2^* = -0.5109$$

et

$$K_3^* = 1.1842$$

Finalement, on parvient au correcteur robuste suivant :

$$u(k) = -0.5186 w(k) - 0.5108 w(k - 1) \quad (4.32)$$

Etape 03 : Evaluation de niveau de satisfaction de la solution obtenu.

D'après le Théorème 3.1, le correcteur $u(k) = K_1^*w(k) + K_2^*w(k - 1)$ garantit $E [y^2(k)] = l_{a,b,c_1,c_2}(K_1^*, K_2^*) \leq K_3^* = 1.18$ avec une probabilité de $1 - \beta = 1 - 10^{-10}$ (en pratique avec une probabilité qui égale à 1) sauf pour une petite fraction de taille $\epsilon = 0,01$ au maximum. Cette propriété de robustesse est démontrée par la figure 4.5 qui représente le tracé $l_{a,b,c_1,c_2}(K_1^*, K_2^*)$ en fonction de la reparamétrisation θ_1, θ_2 de a, b, c_1 et c_2 . La surface plane représente $K_3^* = 1.18$.

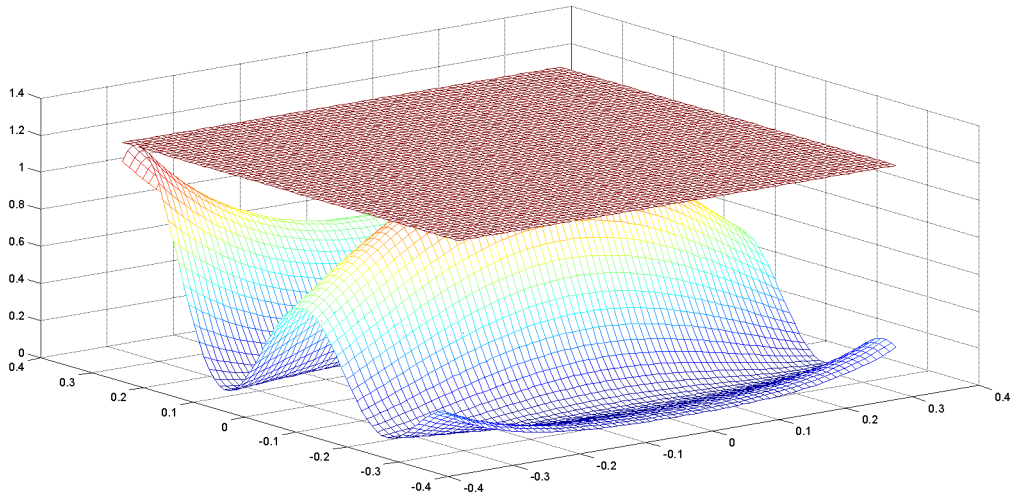


Figure 4.5. Robustesse du correcteur.

Pour mettre en évidence les capacités du correcteur robuste obtenu, on compare les résultats obtenus par les deux correcteurs nominal et robuste.

Résultat pour les conditions nominales (paramètres nominaux) :

La **figure 4.6** donne la sortie du système obtenue lorsque les paramètres du système sont nominaux. On constate que le correcteur nominal présente de meilleures performances par rapport au correcteur robuste. Les résultats sont évidents vu que les paramètres du correcteur nominal sont calculés en utilisant les valeurs nominales des paramètres du système.

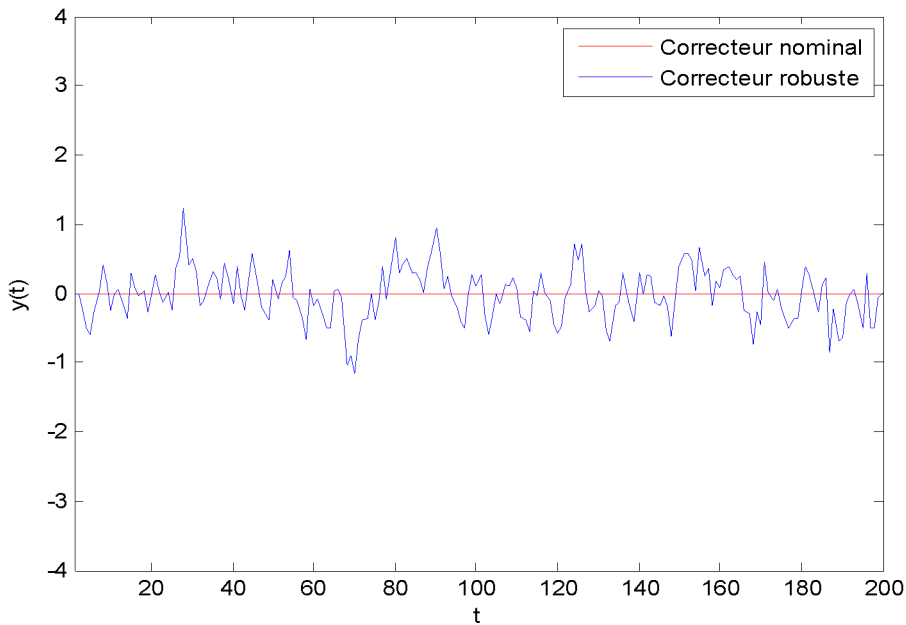


Figure 4.6. Représentation de la sortie du système avec le correcteur robuste, et avec le correcteur nominal

Résultat pour les conditions non-nominales (paramètres incertains) :

Les résultats suivants sont obtenus pour les valeurs $(\theta_1, \theta_2) = (-0.21, -0.32)$:

La **figure 4.7** montre bien qu'en présence d'incertitudes, la sortie du système avec le correcteur robuste approche une moyenne nulle, alors que celle du correcteur nominal

s'éloigne du zéro (moyenne nulle). Ce test mis en évidence les capacités du correcteur robuste dans le rejet des perturbations dues aux incertitudes paramétriques.

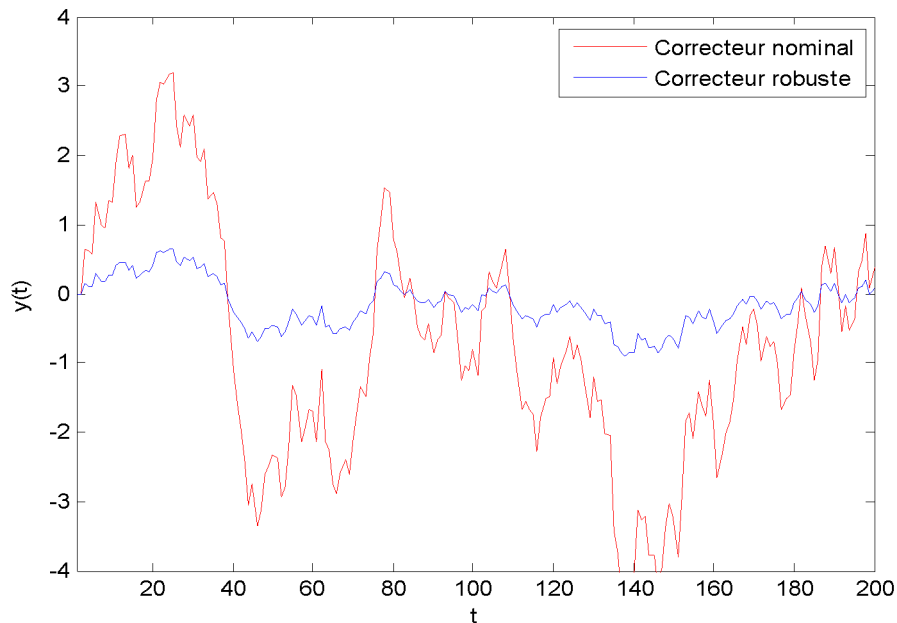


Figure 4.7. Evolution de la sortie du système.

IV. Conclusion :

Dans ce chapitre, nous avons appliqué l'approche de scénarios pour la synthèse d'un correcteur robuste pour un système dynamique caractérisé par des incertitudes paramétriques.

Le problème de conception du correcteur a été formulé sous forme d'un problème d'optimisation dont le nombre de contraintes est infini. Puis, en utilisant l'approche des scénarios, ce problème a été simplifié pour avoir un problème d'optimisation avec un nombre de contraintes fini. Ce dernier a été résolu par les algorithmes génétiques. L'apport du correcteur robuste a été mis en évidence par simulation.

Conclusion Générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la conception de correcteurs robustes pour les systèmes incertains. Le problème de conception est formulé sous forme d'un problème d'optimisation dont le nombre de contraintes est infini. L'objectif consiste à introduire l'approche de scénarios et de l'utiliser pour la résolution du problème d'optimisation formulé.

Ainsi, après avoir rappelé certaines notions relatives aux systèmes incertains et la commande robuste, nous avons introduit les différentes approches d'optimisation statique en particulier l'approche des scénarios utilisée pour la résolution des problèmes d'optimisation robuste. La fin du mémoire a été consacrée à une application de cette approche pour la conception d'un correcteur robuste pour un système incertain. L'objectif est d'assurer une moyenne nulle pour la sortie lorsque le système est soumis à une entrée externe aléatoire.

L'étude réalisée permet de conclure que l'approche des scénarios permet de simplifier davantage la conception de correcteurs robustes formulée sous forme d'un problème d'optimisation robuste dont le nombre de contraintes est infini. Cette approche permet de rendre le nombre de contraintes fini en considérant un nombre bien déterminé de contraintes choisis selon une loi de distribution.

Les résultats de simulation présentés démontrent les performances du correcteur robuste synthétisé en utilisant l'approche des scénarios et motive son exploitation pour la résolution d'autres problèmes par exemple :

- Des correcteurs PID robustes,
- Des commandes optimales en présence d'incertitudes.

Annexe

1. Théorème de superposition : [10]

Si pour chaque entrée $e_i(t)$ est associée une sortie $y_j(t)$ comme sa réponse alors :

$$\text{Si : } a_1 e_1(t) + a_2 e_2(t) + \dots + a_n e_n(t) = a_1 y_1(t) + a_2 y_2(t) + \dots + a_n y_n(t)$$

Alors le système est dit linéaire.

2. Norme l_2 : Pour un signal $x(t)$ a valeur dans R^n , la norme l_2 est définie par :

La norme l_2 est définie comme suit :

$$\|x(t)\|_{l_2} = \sqrt{\int_0^{\infty} x^T x dt}$$

3/ Norme euclidienne :

Soit un vecteur $x \in R^n$, on définit la norme euclidienne de ce vecteur comme suit :

$$\|x\| = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

4/ Norme H_2 : On appelle norme H_2 du transfert $G(s)$ entre w et z le réel positif défini par :

$$\|G(s)\|_2^2 = \frac{1}{2\pi} \sqrt{\int_{-\infty}^{+\infty} \text{Trace} [G(-jw)^T G(jw) dw] d(w)}$$

On peut écrire aussi:

$$\sup_{w \neq 0} \frac{\|z(s)\|_2}{\|w(s)\|_\infty}$$

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

5/ Norme H_∞ : on appelle norme H_∞ du transfert $G(s)$ entre w et z le réel positif défini par :

$$\|G(jw)\|_\infty = \sup_{w \neq 0} \frac{\|z(s)\|_2}{\|w(s)\|_2}$$

On peut écrire aussi:

$$\|G(j\omega)\|_{\infty} = \sup \sigma_{\max}(G(j\omega))$$

Où : σ_{\max} représente la valeur singulière maximale.

6/ Rappel sur les relations de bouclage :

Soit le schéma principal d'un système à rétroaction suivant :

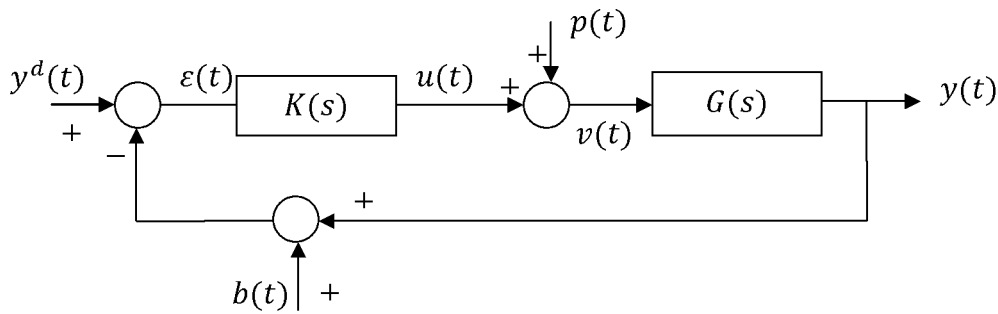


Figure A.1 schéma fondamental d'un système à rétroaction.

6.1. Fonctions de sensibilité

Les relations algébriques élémentaires définies dans le schéma fonctionnel précédent s'écrivent :

$$\begin{aligned} Y(p) &= G(p)V(p) \\ V(p) &= P(p) + U(p) \\ U(p) &= K(p)E(p) \\ E(p) &= Y^d(s) - B(p) - Y(p) \end{aligned}$$

Cela permet de déterminer les relations en boucle fermée entre la sortie exogène et les entrées exogènes :

$$Y(s) = (1 + G(s)K(p))^{-1} G(p)P(s) + (1 + G(s)K(p))^{-1} G(s)K(s) (Y^d(s) - B(s))$$

6.2. Définition : (Sensibilité et sensibilité complémentaire)

Les fonctions de transfert $S(s)$ et $T(s)$ sont appelées respectivement fonction de sensibilité et fonction de sensibilité complémentaire. La fonction de sensibilité complémentaire est en fait la fonction de transfert en boucle fermée usuelle.

$$S(s) = \frac{1}{1 + K(s)G(s)} \quad \text{et} \quad T(s) = \frac{K(s)G(s)}{1 + K(s)G(s)}$$

Il est à noter que $T(s) = G(s)K(s)S(s)$ et donc : $S(s) + T(s) = 1$

Un signal permettant d'analyser la précision du système est celui de l'erreur $\varepsilon(t)$:

$$E(s) = S(s)Y^d(s) - S(s)G(s)P(s) + T(s)B(s)$$

Afin de conserver un certain contrôle sur l'amplitude de la commande délivrée et de la commande reçue, il est également nécessaire de connaître les transferts de ces commandes aux entrées exogènes :

$$U(s) = K(s)S(s)(Y^d(s) - B(s)) - T(s)P(s)$$

$$V(s) = K(s)S(s)(Y^d(s) - B(s)) + S(s)P(s)$$

Le schéma fonctionnel précédent fait donc apparaître quatre fonctions de transfert importantes appelées fonctions de sensibilité : $S(s)$, $S(s)G(s)$, $T(s)$, $S(s)K(s)$ où :

- $S(s)$ représente le transfert entre la référence et l'erreur et entre la perturbation et la commande délivrée.
- $T(s)$ représente l'influence des bruits de mesure sur la sortie et l'erreur et l'influence de la perturbation sur la commande délivrée.
- $S(s)G(s)$ représente l'influence de la perturbation sur la sortie et l'erreur.
- $S(s)K(s)$ représente l'influence de la référence et des bruits de mesure sur les commandes.

6.3. Objectifs de commande

Compte tenu des relations établies précédemment et de l'analyse qui les suit, la synthèse du correcteur K doit obéir aux différents objectifs suivants :

- Un bon suivi du signal de référence implique une fonction de sensibilité $S(s)$ faible en amplitude.
- Une bonne rejection des perturbations sera obtenue pour $S(s)G(s)$ faible.
- Les bruits de mesure seront atténués si l'amplitude de $T(s)$ est faible sur la plage de fréquence concernée.
- L'effort de commande est faible si l'amplitude de $S(s)K(s)$ et de $T(s)$ est faible.
- La commande délivrée au système est faible si $S(s)$ et $S(s)K(s)$ sont d'amplitude faible.

Ces objectifs de commande sont contradictoires et devront faire l'objet de compromis lors de la synthèse du système de commande.

7. Structure Transformation Linéaire Fractionnaire (LFR) :

De manière générale, ce qu'on appelle Représentation Linéaire Fractionnaire (LFR) ou Transformation Linéaire Fractionnaire (LFT) désigne le bouclage d'un système avec un autre système. Dans le cadre de la modélisation des systèmes incertains, cette dénomination désigne une structure qui se compose d'un modèle nominal $G(s)$ (appelé aussi modèle de référence) de structure LTI et des incertitudes de modèle représentées par une matrice $\Delta(s)$ qui englobent les écarts entre le comportement nominal du modèle et le comportement réel du procédé. La **Figure 2.3** donne une représentation standard schématisée d'une structure LFR.

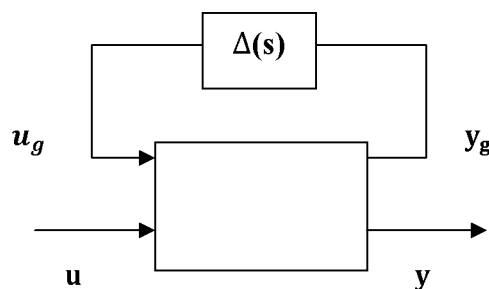


Figure A.2. Structure LFR pour un système incertain.

8. Stabilité / performance :

- Si le système est stable pour l'ensemble des variations des paramètres, on dit qu'il est robuste en stabilité ou robustement stable
- Si le système respecte les critères de performance pour l'ensemble des variations des paramètres, on dit qu'il est robuste en performance
- La robustesse en performance inclut la robustesse en stabilité

9. Marge de robustesse :

- On définit la marge de robustesse comme la dilatation que l'on peut imposer sur le domaine de variations des paramètres tout en conservant la robustesse
- Le système est robuste si la marge de robustesse est supérieure ou égale à 1.

Références bibliographiques

Références bibliographiques

- [1] A. Maidi. Cours d'Optimisation et commande optimale. Département d'Automatique, Faculté Génie Electrique et d'Informatique, Université Mouloud MAMMERRI Tizi Ouzou, 2015/2016.
- [2] M. Bierlaire. Introduction à l'optimisation différentiable. Presses Polytechnique et Universitaire Romandes, Lausanne, 2006
- [3] Y. Collette et P. Siarry. Optimisation multi-objectif. Editions Eyrolles, Paris, 2002.
- [4] A. Gherboudj. Méthodes de résolution de problèmes difficiles académiques. Thèse de Doctorat 3ème cycle LMD, Spécialité : Informatique, Département d'Informatique Fondamentale et ses Applications, Faculté des Technologies de l'Information et de la Communication, Université de Constantine 2, 2013.
- [5] M. C. Campi, S. Garatti and M. Prandini. The scenario approach for systems and design. Annual Reviews in Control, Vol. 33, No. 2, pp., 149-157, 2009.
- [6] A. Ben-Tal, L. El Ghaoui and A. Nemirovski. Robust Optimization, Princeton University Press, 2009.
- [7] F. Jedrzejewski. Introduction aux méthodes numériques, Deuxième édition, édition Springer-Verlag France, Paris, 2005.
- [8] B. Tuffin. La simulation de Monte Carlo, Hermès/Lavoisier, Paris, 2010
- [9] A. El Jai. Eléments d'analyse et de contrôle des systèmes, collection d'études, Presses universitaires des Perpignan, Perpignan, 2004.
- [10] D. Bensoussan. Commande moderne : approche par modèles continus et discrets, Presses inter Polytechnique, 2008.
- [11] R. Tempo, G. Calafiore and F. Dabbene. Randomized Algorithms for Analysis and Control of Uncertain Systems. Edition Springer-Verlag, London, 2013.

- [12] D.W. Gu, P. H. Petkov and M. M. Konstantinov. Robust control design with matlab, Edition Springer, 2013
- [13] D. Arzelier, J. Bernussou, D. Henrion et D. Peaucelle. Optimisation et théorie de la commande robuste, EDP Sciences, 2003.
- [14] V. D. Blondel, J. N. Tsitsiklis. A survey of computational complexity results in systems and control, Automatica, Vol. 36, pp. 1249–1274, 1999.
- [15] S. Garatti and M C.Campi. Modulating Robustness in Control Design: Principles and Algorithms. IEEE Control Systems Magazine, April 2013
- [16] P. Apkarian. Elément de la théorie de la commande robuste. pierre.apkarian.free.fr/COURS/polysae.pdf (Dernière consultation : 31 août 2016)
- [17] D. Alazard, P. Apkarian, C. Cumer, G.Ferreres et M.Gauvrit. Robustesse et commande optimale, Edition Cépaduès, 1999.
- [18] G. Scorletti, V. Fromion. Introduction à la commande multi variable des systèmes : méthode de synthèse fréquentielle H_∞ . Cours de l'ENSI de Caen, 2007.
- [19] D. Arzelier. Cours de Première année Représentation et analyse des systèmes linéaires, Toulouse, France, 2010.
- [20] J. Bernussou. Commande robuste : développement et application, édition Hermès, Paris, 1996.
- [21] P.Feyel. La commande robuste par loop-shaping, Edition Lavoisier, Paris, 2013.
- [22] K. Ghédira. Optimisation combinatoire par méta-heuristique, Editions Technip, 2007.

Références bibliographiques

- [23] R. K. Yedavalli. Robust Control of Uncertain Dynamic Systems: A Linear State Space Approach. Edition Springer, 2013.
- [24] A. Rachid et D. Mehdi. Réalisation, réduction et commande des systèmes linéaires, Edition Technip, Paris, 1997.

Résumé :

Notre travail se rapporte sur la conception des correcteurs pour les systèmes incertains. Cependant, étudier un système incertain fait appel aux différentes commandes robustes assurant la robustesse du système.

De plus, la plus part des systèmes sont formulés comme étant des problèmes d'optimisation, et inclure les incertitudes dans le modèle impose une infinité de contraintes, ce qui rend le système difficile à résoudre, voire irrésoluble (NP-difficile). Cependant, pour résoudre ce type d'optimisation robuste on a introduit l'approche des scénarios comme une méthode qui se base sur la réduction de nombre de contrainte infini à un nombre fini en extrayant aléatoirement N valeurs du domaine incertain selon une loi de distribution. La génération de ces nombres aléatoire avec les exigences imposées sera facilement assurée par la simulation de Monte Carlo. Ainsi le problème qu'était irrésoluble devient dilué et peut être résolu par une des méthodes standards, pour cela on a eu recours aux Algorithmes Génétiques (AGs) qu'ont l'avantage de converger vers un optimum globale difficile à trouver. L'application présentée dans ce mémoire était une application de cette approche pour la conception d'un correcteur robuste pour un système incertain. L'objectif est d'assurer une moyenne nulle pour la sortie lorsque le système est soumis à une entrée externe aléatoire. Et les résultats de simulation présentés démontrent les performances du correcteur robuste synthétisé en utilisant l'approche des scénarios ce qui motive son exploitation pour la résolution d'autres problèmes de monde actuel.

Mots clés :

Commande robuste, système, système de commande, système incertain, incertitude, Modèle, complexité algorithmique, NP-difficile, optimisation, optimisation robuste, optimisation minimax, correcteur, Approche des Scénarios, Simulation de Monte Carlo, Auto Regressif Moving Average (ARMA), Robustesse, méthode de Kharithonov, commande robuste H_∞ , commande robuste par Modelage de Boucle, Algorithmes Génétiques, déterministe, stochastique, optimum, variable aléatoire.