



**HAL**  
open science

# Optimisation non différentielle pour la prise en compte de cahier des charges générique en automatique

Bilal Lassami

► **To cite this version:**

Bilal Lassami. Optimisation non différentielle pour la prise en compte de cahier des charges générique en automatique. Automatique / Robotique. Université Paris Sud - Paris XI, 2008. Français. NNT : . tel-00352962

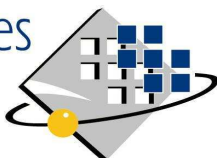
**HAL Id: tel-00352962**

**<https://theses.hal.science/tel-00352962v1>**

Submitted on 14 Jan 2009

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

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



**THÈSE DE DOCTORAT**

**SPECIALITE : PHYSIQUE**

*Ecole Doctorale « Sciences et Technologies de l'Information des  
Télécommunications et des Systèmes »*

Présentée par :

**Bilal LASSAMI**

Sujet :

**OPTIMISATION NON DIFFERENTIABLE POUR LA PRISE EN COMPTE  
DE CAHIER DES CHARGES GÉNÉRIQUE EN AUTOMATIQUE**

Soutenue le 16 Juillet 2008 devant les membres du jury :

M. Y. CHITOUR

Président

M. S. FONT

Directeur de thèse

M. V. FROMION

Rapporteur

Mme H. SIGUERDIDJANE

Co- Directeur de thèse

M. M. ZASADZINSKI

Rapporteur



## *Remerciements*

*Nombreux sont ceux que je voudrais remercier pour m'avoir aidé, soutenu ou accompagné durant mes années de thèse. C'est pour leur montrer toute ma reconnaissance que je leur dédie ces quelques lignes.*

*J'aimerais, avant toute autre personne, remercier Monsieur Patrick BOUCHER, Chef du Département Automatique de Supélec, pour m'avoir accueilli et pour avoir mis à ma disposition tout ce qui a été nécessaire à la bonne réalisation de ce travail.*

*Cette thèse ne pourrait pas avoir été menée à terme sans la confiance, la patience et la générosité du Professeur Stéphane FONT, à qui je veux apporter mes remerciements tout particuliers. Il a su m'encadrer avec efficacité, clairvoyance et beaucoup de patience durant tout le temps que m'a exigé ce travail. Je lui exprime ma profonde reconnaissance pour son soutien et encouragement face à la difficulté mais aussi pour sa disponibilité et le temps passé à la relecture et la correction du mémoire de thèse. Je lui apporte ma plus sincère gratitude pour le temps précieux qu'il m'a accordé tout au long de ces années.*

*Je tiens également à témoigner ma reconnaissance à Madame Houria SIGUERDIDJANE qui a co-dirigé cette thèse, pour la patience et la disponibilité dont elle a su faire preuve tout au long de nos discussions.*

*Je remercie également Monsieur Y. CHITOUR, Professeur à l'Université Paris-Sud XI de m'avoir fait l'honneur de présider mon jury de thèse.*

*Mes remerciements s'adressent ensuite à Monsieur V. FROMION, Directeur de recherche à l'INRA et Monsieur M. ZASADZINSKI, Professeur à l'Université Henri Poincaré- Nancy I, pour avoir accepté d'évaluer mes travaux en qualité de rapporteurs. Leurs questions et leurs commentaires pertinents m'ont permis de rendre plus claire ma rédaction et m'ont donné de nouvelles pistes de réflexion.*

*Que tous les membres du Département Automatique trouvent ici l'expression de ma gratitude. Une pensée particulière à Guillaume SANDOU, Professeur au Département Automatique, qui m'a beaucoup aidé pour finaliser mon mémoire de thèse et à Madame Josiane DARTRON, secrétaire du Département d'Automatique pour son amitié, son efficacité, son sourire en toutes circonstances.*

*Je ne saurai terminer mes remerciements sans une pensée pour ma famille. Je m'adresse à ma mère, mes frères, ma sœur, ma grande mère ainsi qu'à toute ma famille que je porte toujours avec moi dans ma pensée pour leur exprimer ma profonde reconnaissance pour leur soutien pendant toutes ces années de thèse. Sans leur confiance immense en moi, sans leur aide et leur amour, je n'aurais pas pu aller au bout de mes projets. Qu'ils trouvent en moi l'enfant redevable toute sa vie.*

*Enfin, je remercie tout particulièrement ma femme Karima qui a toujours cru en moi et qui m'a apporté son aide et son réconfort.*





*À la mémoire de mon père  
À ma famille*



# Table des matières

<b>Chapitre 0 Introduction.....</b>	<b>0-1</b>
Motivations .....	0-1
Contexte et problématique .....	0-2
Organisation du manuscrit .....	0-3
<b>Chapitre 1 Formalisme d'un cahier des charges générique pour un problème de commande .....</b>	<b>1-1</b>
1.1. La théorie de la contre-réaction.....	1-1
1.2. Vie d'un cahier des charges .....	1-4
1.3. Les étapes d'un problème de commande.....	1-4
1.4. Le problème fondamental de la commande.....	1-6
1.5. Les spécifications d'un cahier des charges.....	1-7
1.5.1. La stabilité.....	1-8
1.5.1.1. Stabilité entrée/sortie (stabilité externe) .....	1-9
1.5.1.2. Stabilité externe des systèmes linéaires.....	1-9
1.5.1.3. La stabilité au sens de Lyapunov (stabilité interne) .....	1-9
1.5.1.4. La passivité.....	1-11
1.5.1.5. Stabilité interne des systèmes linéaires continus invariants .....	1-12
1.5.1.6. Critères de stabilité des systèmes linéaires.....	1-13
1.5.2. Les spécifications en performances.....	1-16
1.5.2.1. Suivi de trajectoire de référence (consigne) .....	1-17
1.5.2.2. Rejet/atténuation de signaux de perturbation. ....	1-21
1.5.2.3. Atténuation des bruits de mesure. ....	1-22
1.5.2.4. Commande modérée.....	1-23
1.5.2.5. Dépendance des spécifications et limite de performances .....	1-23
1.5.2.6. Performances des systèmes linéaires multivariables .....	1-28
1.5.3. Les spécifications en robustesse.....	1-30
1.5.3.1. La robustesse en stabilité.....	1-32
1.5.3.2. La robustesse en performance : .....	1-37
1.5.4. Les spécifications de la loi de commande .....	1-39
1.6. Classification des méthodes de synthèse.....	1-39
1.6.1. Méthodes de synthèse synthétiques.....	1-39
1.6.2. Méthodes de synthèse analytiques .....	1-40
1.6.3. Méthodes de synthèse par optimisation.....	1-40
1.6.3.1. Classification (convexité et dimension du problème) .....	1-42
1.7. Vers une approche d'optimisation non linéaire.....	1-44
1.8. Conclusion .....	1-44
<b>Chapitre 2 Formulation et analyse de cahier des charges par optimisation .....</b>	<b>2-1</b>
2.1. Concepts fondamentaux pour une approche générique .....	2-2
2.2. Cahiers des charges et critères mathématiques .....	2-3
2.3. Traduction du cahier des charges .....	2-4
2.3.1. Expression des contraintes temporelles E/S du système .....	2-5
2.3.1.1. Formulation directe des spécifications temporelles.....	2-5
2.3.1.2. Formulation des spécifications temporelles sous forme de gabarits.....	2-6
2.3.2. Faisabilité des cahiers des charges temporels .....	2-8
2.3.2.1. Etude de la faisabilité des spécifications temporelles par une approche E/S .....	2-9
2.3.2.2. Etude de la faisabilité des spécifications temporelles par une approche paramétrique générale.....	2-20
2.3.3. Expression des contraintes fréquentielles du système .....	2-26
2.3.3.1. Formulation des spécifications fréquentielles à base d'indicateurs de robustesse.....	2-26
2.3.3.2. Formulation des spécifications fréquentielles sous forme de gabarits .....	2-30
2.3.3.3. Formulation des spécifications fréquentielles modales .....	2-32
2.3.4. Faisabilité des cahiers des charges fréquents .....	2-33

2.3.4.1. Etude de la faisabilité des spécifications fréquentielles par une approche par opérateurs.....	2-33
2.3.4.2. Etude de la faisabilité des spécifications fréquentielles par une approche paramétrique .....	2-48
2.4. Bilan et approche paramétrique générale pour l'analyse de cahiers des charges .....	2-50
2.4.1. Formulation du problème d'optimisation global de l'approche paramétrique générale.....	2-52
2.4.2. Transformation en un problème d'optimisation non linéaire sans contraintes.....	2-53
2.5. Sensibilité des critères et contraintes d'une approche paramétrique générale.....	2-53
2.6. Conclusions.....	2-58
<b>Chapitre 3 Techniques d'optimisation non linéaire .....</b>	<b>3-1</b>
3.1. Contexte du travail .....	3-1
3.2. Principes généraux .....	3-2
3.2.1. Problème d'optimisation .....	3-2
3.2.2. Conditions d'optimalité.....	3-4
3.2.2.1. Conditions d'optimalité en l'absence de contraintes .....	3-4
3.2.2.2. Conditions d'optimalité en optimisation avec contraintes.....	3-4
3.3. Prise en compte des contraintes par pénalisation .....	3-6
3.3.1. Types de pénalisations .....	3-7
3.3.1.1. Pénalisations extérieures .....	3-8
3.3.1.2. Pénalisations intérieures.....	3-11
3.3.1.3. Pénalisations exactes .....	3-13
3.4. Méthodes globales versus méthodes locales .....	3-14
3.5. Méthodes d'optimisation sans contraintes.....	3-15
3.5.1. Les méthodes stochastiques .....	3-16
3.5.1.1. Métaheuristiques .....	3-16
3.5.1.2. Principe .....	3-17
3.5.1.3. Classification.....	3-18
3.5.1.4. Exemples de métaheuristiques .....	3-18
3.5.2. Les méthodes de descente .....	3-19
3.5.2.1. Calcul de la direction de descente .....	3-21
3.5.2.2. Calcul de la longueur de descente (recherche linéaire) .....	3-26
3.5.3. Les méthodes mixtes .....	3-32
3.5.4. Les méthodes de recherche directe.....	3-33
3.5.4.1. Méthodes de recherches par motifs généralisés .....	3-34
3.5.4.2. Directions conjuguées (algorithme de Powell).....	3-34
3.6. Conclusions.....	3-35
<b>Chapitre 4 Algorithmes d'optimisation développés .....</b>	<b>4-1</b>
4.1. Algorithme du simplexe modifiée.....	4-2
4.1.1. Algorithme de Nelder-Mead (Simplexe).....	4-2
4.1.2. Détection et traitement des dégénérescences .....	4-5
4.1.3. Prise en compte des bornes .....	4-6
4.1.3.1. Prise en compte des bornes par projection .....	4-7
4.1.3.2. Prise en compte des bornes par reparamétrisation.....	4-8
4.2. Méthodes du sous-différentiel .....	4-14
4.2.1. Généralisation du gradient .....	4-14
4.2.1.1. Définitions.....	4-14
4.2.1.2. Propriétés.....	4-15
4.2.1.3. Stationnarité .....	4-17
4.2.2. Méthodes de descente non différentiables.....	4-17
4.2.3. Le sous-différentiel de Clarke est ses propriétés.....	4-20
4.2.4. Algorithme du epsilon-sous-différentiel (AESD) .....	4-23
4.2.4.1. Notations .....	4-23
4.2.4.2. AESD .....	4-24
4.2.5. Algorithme du epsilon-sous-différentiel modifié (AESDM).....	4-27
4.2.5.1. Notations .....	4-28
4.2.5.2. AESDM.....	4-28
4.2.6. Algorithme de gradient Universel .....	4-29
4.2.6.1. Notations .....	4-30
4.2.6.2. AGU .....	4-31
4.3. Évaluation et comparaison des algorithmes développés .....	4-33

4.3.1. Les problèmes test.....	4-33
4.3.2. Résolution des problèmes test sans contraintes d'encadrement .....	4-34
4.3.2.1. Paramètres des algorithmes .....	4-35
4.3.2.2. Analyse des résultats numériques.....	4-36
4.3.3. Résolution des problèmes test avec contraintes d'encadrement.....	4-43
4.3.3.1. Paramètres des algorithmes .....	4-44
4.3.3.2. Analyse des résultats numériques.....	4-44
4.4. Conclusions .....	4-49
<b>Chapitre 5 Analyse du <math>\varepsilon</math>-sous différentiel et calcul du gradient.....</b>	<b>5-1</b>
5.1. Estimation du $\varepsilon$ -sous-différentiel de Clarke .....	5-2
5.1.1. Échantillonnage uniforme dans une hyperboule .....	5-2
5.1.2. Choix du nombre d'échantillons $m$ .....	5-4
5.1.3. Détection de zones non différentiables et ouverture d'un $\mathcal{E}$ -sous-différentiel .....	5-9
5.2. Calcul et estimation du gradient.....	5-10
5.2.1. Différences finies .....	5-10
5.2.2. Méthodes de fonctions de sensibilité.....	5-15
5.2.2.1. Sensibilités des systèmes non linéaires .....	5-17
5.2.2.2. Sensibilités des systèmes linéaires .....	5-17
5.2.2.3. Sensibilité paramétrique fréquentielle.....	5-19
5.2.3. État adjoint .....	5-23
5.2.4. Différentiation automatique et code adjoint.....	5-27
5.2.4.1. Mode direct .....	5-27
5.2.4.2. Mode inverse.....	5-28
5.2.4.3. Principes de différentiation automatique par code adjoint .....	5-28
5.2.5. Dérivation complexe .....	5-33
5.2.5.1. Principe de la méthode .....	5-33
5.2.5.2. Approximation des dérivées d'ordre supérieur .....	5-35
5.2.5.3. Procédure d'implémentation (fonctions complexes de base et opérateurs en Matlab).....	5-36
5.3. Conclusion .....	5-39
<b>Chapitre 6 Analyse et validation de cahiers des charges pour des problèmes de commande .....</b>	<b>6-1</b>
6.1. Problèmes de stabilisation linéaire .....	6-2
6.1.1. Problème de stabilisation pure .....	6-2
6.1.2. Optimisation de l'abscisse spectrale.....	6-6
6.1.3. Problème de stabilisation simultanée (Application au problème de chocolat belge) .....	6-11
6.2. Synthèse $H_\infty$ à structure fixe.....	6-18
6.2.1. Problème de synthèse $H_\infty$ .....	6-19
6.2.2. Problème de stabilisation robuste (optimisation du rayon de stabilité complexe).....	6-20
6.3. Asservissement PID d'un moteur à courant continu .....	6-26
6.3.1. Présentation du système .....	6-26
6.3.2. Cahier des charges.....	6-28
6.3.3. Formulation du problème .....	6-28
6.3.4. Résolution, simulation et validation expérimentale .....	6-29
6.3.5. Minimisation du temps de réponse.....	6-32
6.3.6. Modification de la structure PID .....	6-34
6.4. Stabilisation robuste d'un système oscillatoire .....	6-37
6.4.1. Présentation du problème .....	6-37
6.4.2. Cahier des charges.....	6-37
6.4.3. Formulation en un problème d'optimisation .....	6-37
6.5. Commande Backstepping d'une suspension magnétique.....	6-39
6.5.1. Modélisation du système .....	6-40
6.5.2. Cahier des charges.....	6-42
6.5.3. Synthèse de la commande Backstepping en vue d'une validation expérimentale .....	6-43
6.5.4. Synthèse de la commande Backstepping.....	6-44
6.5.5. Synthèse de la commande Backstepping par retour de sortie.....	6-55
6.5.6. Synthèse d'une commande Backstepping à action intégrale par retour de sortie.....	6-61
6.5.7. Commande Backstepping par retour de sortie adaptative .....	6-65
6.5.8. Prise en compte de la dynamique de l'actionneur .....	6-69
6.6. Commande d'un système de forage pétrolier.....	6-72

6.7. Commande d'un missile fortement manœuvrant .....	6-74
6.7.1. Objectifs de la commande .....	6-74
6.7.2. Modèle du missile .....	6-75
6.7.3. Cahier des charges.....	6-76
6.7.4. Commande du missile par un correcteur PI .....	6-76
6.7.5. Commande linéarisante par retour d'état dynamique.....	6-81
6.8. Conclusion .....	6-87
<b>Chapitre 7 Conclusions et perspectives .....</b>	<b>7-1</b>
Apports scientifiques et originalité du travail .....	7-1
Perspectives.....	7-4
<b>Chapitre 8 Annexes .....</b>	<b>8-1</b>
8.1. Annexe 1 : Commande et observateur par Backstepping .....	8-1
8.2. Annexe 2 : Commande d'une suspension magnétique.....	8-11
8.3. Annexe 3 : Commande d'un système de forage pétrolier .....	8-17

**Références Bibliographiques.**

# Chapitre 0

## Introduction

### Motivations

Malgré la diversité des méthodes existantes, face à un problème industriel ou même académique comprenant des spécifications précises et variées, la synthèse de loi de commande reste très difficile.

Pourquoi ? Peut-être ce cahier des charges est-il mal défini, incompatible avec les performances atteignables par le système, son contenu est-il insuffisant ou bien encore n'est-il pas correctement formulé pour être exploitable ? Probablement un peu toutes ces raisons à la fois, mais surtout, très souvent, parce qu'il est "tout simplement" très bien défini et très complet.

En effet, dans le cas général, les méthodes de calcul de correcteurs ne peuvent prendre en compte directement toutes les demandes d'un cahier des charges. Elles nécessitent des reformulations, des validations à posteriori et des pratiques de type essais/erreurs pour obtenir un bon résultat.

Les problèmes liés au cahier des charges génériques ne sont pourtant pas délaissés. Les praticiens et théoriciens en Automatique montre régulièrement leur intérêt pour ce sujet :

- définition et transformation de cahier des charges (les plus complets possibles) sous des formes mathématiques adaptées à un traitement efficace
- prise en compte (plus ou moins directe) de spécifications variées lors d'une synthèse de correcteurs
- problèmes de reformulations (convexe, multicritère...)
- étude de la faisabilité du cahier des charges
- logiciel d'aide à la conception intégrée

Beaucoup d'études choisissent une catégorie particulière de critère et étudient les possibilités de résolution directe (éventuellement par optimisation convexe) et les extensions possibles des critères. En effet, le problème étant par nature très complexe, il n'est pas possible de proposer un d'outil miracle qui résout le problème universel.

Dans notre problématique nous développons une approche complémentaire, qui, quitte à perdre certaines propriétés, conserve la possibilité de traiter un cahier des charges génériques efficacement.



## Contexte et problématique

Les différentes étapes de conception d'un système de commande n'ont pas connues le même développement. Alors que plusieurs techniques de modélisation, d'identification, d'analyse et de synthèse de lois de commande existent et ne cessent de se développer, très peu de recherches ont été effectuées pour formuler et analyser le cahier des charges d'un problème de commande. Ceci est certainement dû à la difficulté de réunir la conception d'un cahier des charges et la synthèse d'une loi de commande dans un même formalisme générique, précis et solvable.

Comme exemple de ce type de difficultés, nous pouvons citer le problème du choix de la structure de commande : c'est une des problématiques les plus difficiles lors de l'élaboration d'un cahier des charges qui peut avoir de fortes conséquences sur les performances du système de commande. En effet, l'amélioration des performances en modifiant une structure de commande peut être beaucoup plus importante que celle observée en retouchant une loi de commande particulière.

Par ailleurs, un bilan des méthodes de synthèses nous montre que pour la plupart, elles consistent à exprimer des cahiers des charges d'un type donné sous forme d'optimisation convexe (LQG,  $H_2$ ,  $H_\infty$ ,  $H_2/H_\infty$ , Placement de pôles, "multicritères"... ) [Sch95]. Les algorithmes correspondants sont efficaces en terme de résolution mais ils ne traitent que des cahiers des charges particuliers adaptés à chaque méthode considérée.

Pour une meilleure prise en compte d'un cahier des charges, il est possible de ne pas partir d'un critère particulier mais d'utiliser un critère général regroupant l'ensemble des spécifications. Dans ce cas, les demandes sont formulées plus précisément mais les problèmes d'optimisation obtenus sont non convexes et présentent des minimums locaux. Que faire pour rendre une telle démarche viable ?

**Le contexte général** de la thèse est la recherche d'une réponse au problème de prise en compte d'un cahier des charges générique par une approche de synthèse par optimisation non linéaire. Étant donné un processus et un ensemble de spécifications (en général de natures très diverses) sur le comportement souhaité pour le système bouclé, peut-on développer une formulation générique qui permet via des algorithmes d'optimisation adaptées, de répondre aux spécifications du cahier des charges ?

Bien évidemment, nous ne pourrons pas changer les problèmes liés à la complexité et garantir de trouver un optimum global. **Dans notre problématique**, nous nous intéresseront à l'efficacité de la mise en œuvre :

- Efficacité des méthodes tant dans le choix de la formulation (type de formulation des critères et des contraintes) que dans le choix des algorithmes.
- Adaptation ou création d'algorithmes spécifiques.
- Et surtout, prise en compte des régularités des problèmes pour obtenir des méthodes efficaces et faciles à mettre en œuvre.

Ce travail de recherche se propose d'étudier ces problématiques dans le prolongement des travaux portant sur les formulations convexes [Boy91, Hin98, Hba02a, Hba02b] en utilisant différentes

méthodes de l'optimisation non linéaire avec prise en compte de l'information a priori : correcteur initial intelligent, utilisation de changement de paramètres pour le conditionnement (tenant compte des comportements attendus pour la solution), hiérarchisation du problème global en sous problèmes présentant des qualités de résolution supérieures.

C'est dans cette perspective que s'inscrit notre travail. Il s'est déroulé dans la cadre d'un contrat ministériel de recherche MENRT au sein du Département Automatique (EA1399) de Supélec. A la base, cette étude a été menée suivant la thématique définie dans le groupe de travail : GdR Analyse de la performance des systèmes non linéaires visant à étendre les résultats obtenus précédemment [Hba02a, Hba02b] dans un cadre plus générique.

## **Organisation du manuscrit**

Le manuscrit est organisé en trois grandes parties.

La première partie (chapitres 1, 2) contient tous les rappels et développements mathématiques. Y sont détaillés les critères et les procédures de mise en œuvre spécifiques au problème de commande dans le cadre de notre approche (formulation générique pour l'optimisation). Afin de montrer la généralité de l'approche proposée, cette dernière est parallèlement développée aux approches convexes déjà existantes [Hba02a].

La seconde partie (chapitres 3, 4 et 5) est dédiée à l'optimisation : contexte, développements des algorithmes spécifiques et outils de calcul numérique associés.

La dernière partie (chapitre 6) présente des applications des développements théoriques des précédentes parties. Elles sont du domaine de l'Automatique et concernent des problèmes de synthèse et/ou de retouche de loi de commande.

### ***Chapitre 1***

Nous commencerons donc cette étude par un bref rappel des différentes phases du cycle de développement d'un système de commande. Ceci nous conduira à nous focaliser sur la formulation de deux phases particulière de ce cycle : la définition du cahier des charges et la synthèse de la loi de commande. Ces deux phases correspondent à deux étapes voisines situées au cœur du cycle. Nous expliquons pourquoi les relations qu'elles entretiennent sont délicates et comment elles peuvent être gérées dans le cadre d'une approche "expert" par des retouches de loi de commande.

Nous présentons ensuite les notions et concepts d'Automatique indispensables à l'approche générique. Nous exposons les résultats clefs qui régissent le problème fondamental de la commande et rappelons les critères les plus courants qui permettent de traduire les objectifs de stabilité, sensibilité, performances et robustesse d'un cahier des charges générique.

Nous détaillons, également, les différentes démarches par optimisation existantes menant à l'élaboration et la validation d'un cahier des charges en Automatique. Une étude bibliographique détaillée est également présentée.

## ***Chapitre 2***

Ce chapitre est consacré aux problématiques de formulation et d'analyse du cahier des charges. Deux approches concernant la problématique de faisabilité d'un cahier des charges sont traitées. Notons que chaque approche peut aborder des problèmes dans les domaines temporels et/ou fréquentiels.

La première se base sur la formulation en un problème d'optimisation convexe obtenu soit par une analyse de trajectoires E/S du système linéaire à commander, soit par une paramétrisation convexe des transferts en boucle fermée. Les problèmes d'optimisation formulés sont convexes mais de dimensions infinies. Ils ne peuvent être résolus directement ; mais des approximations finies du problème peuvent l'être.

La deuxième approche se base sur une formulation paramétrique globale sur les paramètres de commande. Elle vise à formuler finement les spécifications du cahier des charges. En contrepartie, elle ne fournit aucune garantie de résolution globale des problèmes d'optimisation formulés. Cette difficulté peut être compensée, par exemple, par une bonne initialisation des paramètres de la loi de commande ou par une utilisation "experte" de l'outil. Par exemple en faisant des synthèses de plus en plus complètes par augmentation progressive de la complexité.

Notons que ces deux approches nécessitent des informations a priori pour une résolution efficace et qu'elles sont complémentaires de par leurs qualités et défauts respectifs :

- Résolution d'un problème convexe mais prise en compte de cahier des charges spécifiques pour la première.
- Prise en compte de cahier des charges générique mais résolution de problème d'optimisation non linéaire sans garantie de trouver l'optimum global pour la seconde.

Par la suite, ce sont les problèmes d'optimisation non convexes de la deuxième approche qui seront considérés afin d'identifier leurs caractéristiques communes : celles qui peuvent être exploitées pour obtenir une meilleure résolution. Nous verront que ces problèmes ne sont pas différentiables partout mais restent presque partout différentiables.

## ***Chapitre 3***

Ce chapitre est consacré à l'exposition des principales techniques d'optimisation non linéaire. Nous avons insisté sur les approches qui se sont avérées intéressantes pour être adaptées aux spécificités des problèmes différentiables presque partout.

Le concept de pénalisation est introduit. Il permet d'unifier les résolutions des problèmes d'optimisation avec et sans contraintes dans un même formalisme afin qu'elles puissent être traitées avec un même algorithme d'optimisation sans contraintes. Un état de l'art des méthodes d'optimisation non linéaire est dressé et une classification est proposée.

## ***Chapitre 4***

Nos travaux nous ont amené à nous intéresser aux méthodes déterministes qui présentent des propriétés de robustesse vis-à-vis des erreurs de calcul des critères et des directions de descente. En particulier, nous développons les méthodes de recherche directe de type simplexe de Nelder-Mead et les méthodes de descente à base des généralisations du gradient.

Ces deux méthodes sont introduites et puis les différentes versions développées pour prendre en compte les problèmes de non différentiabilité sont détaillées :

- L'algorithme de Nelder-Mead est amélioré pour surmonter son problème de dégénérescence via une série de réinitialisations bien adaptées.
- L'algorithme de plus profonde descente est généralisé pour le cas des critères non différentiables en utilisant la notion du  $\varepsilon$ -sous-différentiable de Clarke. Cette généralisation présente de bonne propriété de calcul avec une estimation efficace et simple du  $\varepsilon$ -sous différentiel via un nombre fini de gradients échantillonnés.

Le développement de ces algorithmes s'est fait en plusieurs étapes afin d'améliorer progressivement leur efficacité tout en conservant leur principale propriété de robustesse.

En restant dans la même classe de problèmes, les contraintes paramétriques de bornes sont prises en compte soit par projection soit par changement de variable.

Afin d'évaluer leurs performances, les algorithmes résultants sont soumis à une série de problèmes test de complexités variées : mixte, mal conditionné, non différentiable, non Lipschitz et contraints. Ces difficultés sont semblables, du point de vue de l'optimisation, à ceux d'un problème de cahier des charges.

### *Chapitre 5*

Ce chapitre traite la problématique du calcul des gradients qui seront utilisés pour l'estimation du  $\varepsilon$ -sous-différentiel de Clarke. Le calcul du gradient devant être fait de nombreuses fois pour obtenir de bonne estimation des  $\varepsilon$ -sous gradient, il doit être précis et rapide.

Pour ce faire, nous faisons une synthèse du calcul de gradient prenant en compte les spécificités des problèmes posés par les critères d'Automatique (voir le premier chapitre). La majorité de ces critères dépendent implicitement des paramètres de commande et leurs gradients nécessitent ainsi l'emploi de la règle de chaîne de dérivation.

En particulier, sur le plan numérique, deux méthodes d'évaluation sont présentées :

- *La méthode des sensibilités paramétriques* : elle concerne les spécifications temporelles. Théoriquement, elle est exacte. Elle permet de ramener le calcul du gradient à une résolution d'un système différentiel qui peut être évalué simultanément avec la résolution du système en boucle fermée.
- *La méthode de dérivation complexe* : elle est basée sur une propriété des fonctions analytiques (holomorphes). Cette méthode approchée permet d'éviter le choix délicat du paramètre de perturbation utilisé lors des estimations du gradient par différences finies.

L'efficacité de ces deux techniques est mesurée et comparée à d'autres méthodes via plusieurs exemples numériques.

## **Chapitre 6**

Ce chapitre est consacré à l'application de l'approche par optimisation non linéaire développée pour une série de problèmes de commande et de retouche de correcteurs pour des cahiers des charges variés. Le caractère générique de l'approche développée permet de traiter plusieurs problèmes (linéaire, non linéaire, temporel, fréquentiel) dans un même et seul formalisme.

En particulier, une application de synthèse et de retouche d'une commande par Backstepping pour un système de suspension magnétique est présentée. Ainsi, sur le même banc d'essais incluant la suspension magnétique, et reprenant les solutions élaborées pendant les développements de ce mémoire, des lois de commande sont progressivement élaborées et testées, en commençant par un cahier des charges initial, puis dans un cadre de retouche et en analysant ensuite les conséquences de l'évolution du cahier des charges sur les performances atteintes.

## **Conclusion**

Un bilan est formulé. Il reprend l'ensemble des points forts dégagés dans les différents chapitres. Bien que disposant maintenant d'approches très efficaces pour les problèmes traités, de nombreux points restent encore à étudier. Des perspectives sont indiquées. Elles peuvent donner lieu à des développements court terme ou long terme et ceci dans des domaines appliqués ou théoriques.

Les travaux présentés dans ce mémoire ont donné lieu aux différentes communications scientifiques suivantes :

B. Lassami, F. Abdulgalil, S. Font and H. Siguerdidjane, Parametric adjustment of a backstepping controller by nonsmooth optimization: Application to a rotary drilling system, International Journal of Tomography & Statistics. Special Issue on Control Applications of Optimisation. Volume 6, No. S07, Summer 2007, page 134-139.

B. Lassami, S. Font, Backstepping controller retuning using epsilon subdifferential optimization: Application to a magnetic suspension system, 44th IEEE Conference on Decision and Control CDC and European Control Conference ECC, Seville, Espagne, December 2005.

B. Lassami, F. Abdulgalil, S. Font and H. Siguerdidjane, Parametric adjustment of a backstepping controller by nonsmooth optimization: Application to a rotary drilling system, 13th IFAC Workshop on Control Applications of Optimization, Paris, France, April 2006.

B. Lassami, S. Font and H. Siguerdidjane, Improvement of the temporal performance of a nonlinear missile autopilot by nonconvex nonsmooth optimization, IET International Control Conference 2006, Glasgow, Scotland, August 2006.

B. Lassami, S. Font, Computer-aided design and retuning of linear control systems: Nonsmooth optimization approach, 6th Asian Control conference, Bali, Indonesia, July 2006.

B. Lassami, S. Font, Linear controller retuning approach based on nonconvex nonsmooth optimization, IEEE CCA/CACSD/ISIC conferences, Munich, Germany, October 2006.

B. Lassami, S. Font and H. Siguerdidjane, Nonsmooth optimization for nonlinear missile autopilot: Improvement under time domain constraints, IEEE CCA/CACSD/ISIC conferences, Munich, Germany, October 2006.

B. Lassami, S. Font, Optimisation générale pour la retouche de correcteurs : Approche par sous-gradient, Journées Doctorale d'Automatique JDMACS/JNMACS, Lyon, France, Septembre 2005.

G. Sandou, B. Lassami, Optimisation par essaim particulière pour la synthèse ou la retouche de correcteurs, 7ième Conférence Internationale de Modélisation et Simulation MOSIM, Paris, France, Mars 2008.

B. Lassami, S. Font, Optimisation pour la prise en compte de cahiers des charges génériques en automatique : Approche par sous-gradient, Groupe de travail Méthodes et Outils pour la Synthèse et l'Analyse en Robustesse (MOSAR), Lille, France, Juin 2005.



# Chapitre 1

## Formalisme d'un cahier des charges générique pour un problème de commande

Que ce soit pour une résolution globale d'un problème de commande ou pour une retouche de correcteur, l'utilisation de l'optimisation non linéaire pour un problème de commande est pleinement justifiée quand il s'agit de prendre en compte des cahiers des charges génériques : multicritère, complexe, temporel et fréquentiel... La formulation d'un problème générique nécessite donc d'avoir une bonne vue d'ensemble des formulations des cahiers des charges.

Dans ce chapitre, nous aborderons le problème de la formulation des spécifications d'un cahier des charges en termes d'objectifs de synthèse : stabilité, sensibilité, performances et robustesse, de même que les outils et les critères mathématiques permettant de quantifier ces propriétés. Seront également définis les principaux concepts et outils communs à tout le manuscrit.

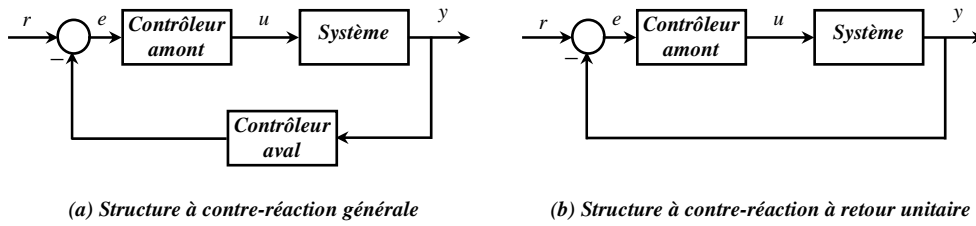
Quiconque souhaite formuler un problème générique peut utiliser ce chapitre comme une aide à la formulation en le parcourant comme une check-list. Une étude bibliographique détaillée permet également un approfondissement en retrouvant le sens "historique" et complet des notions utilisées.

Nous débutons ce manuscrit par un rappel des concepts génériques de l'Automatique (en détaillant quelques propriétés liées au cas linéaire) qui régissent le problème fondamental de la commande. Ensuite, l'accent est mis sur le choix de la structure de commande qui s'avère primordiale pour une approche par optimisation. Nous détaillerons, également, les différentes démarches par optimisation existantes menant à l'élaboration et la validation d'un cahier des charges en Automatique.

### 1.1. La théorie de la contre-réaction

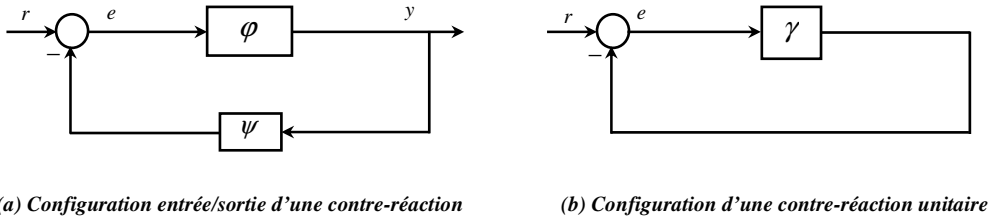
L'idée de la "structure bouclée" est ancienne et remonte aux travaux de Black dans les années 1930 [Bla34]. Elle représente aujourd'hui le noyau commun pour tout outil et technique de commande et de diagnostic des systèmes [May70]. Cette idée de contre-réaction aboutit à des implications d'une grande portée [Kwa91]. La structure de boucle générale est représentée sur la figure 1.1(a). La figure 1.1(b) représente une configuration simplifiée de la boucle de contre-réaction générale, dite à retour unitaire.





**Fig. 1.1** Structure d'un système à contre-réaction

La configuration de la figure 1.1 peut être ramenée à celle de la figure 1.2. Le système ainsi que le correcteur en amont représentent la chaîne directe de cette boucle de commande, représentée par une relation entrée/sortie (E/S) avec un opérateur  $\varphi$ . La chaîne de retour (le correcteur en aval) est également assimilée à un opérateur E/S  $\psi$ . Les différents signaux temporels de la boucle fermée sont reliés entre eux via ces opérateurs E/S.



**Fig. 1.2** Configuration E/S d'une contre-réaction

Sous l'hypothèse d'une boucle fermée est bien posée, le système à contre-réaction est régi par les équations suivantes :

$$\begin{cases} y = \varphi(e) \\ e = r - \psi(y) \end{cases} \quad (1-1)$$

Le signal d'erreur  $e$  satisfait à l'équation E/S de la boucle donnée par :

$$e + \psi(\varphi(e)) = e + \gamma(e) = r \quad (1-2)$$

Le schéma équivalent à cette équation est donné par la figure 1.2(b)

La commande à contre-réaction est à son efficacité maximale si la relation E/S de la boucle possède un grand gain [Bod40]. L'une des conséquences importantes est que la relation E/S entre  $r$  et  $y$  est approximativement l'inverse de l'opérateur  $\psi$ . D'où le fait que la relation de l'entrée de référence  $r$  à la sortie du système  $y$  est quasiment indépendante de la chaîne directe (définie par  $\varphi$ ) et par conséquent indépendante du système à commander :

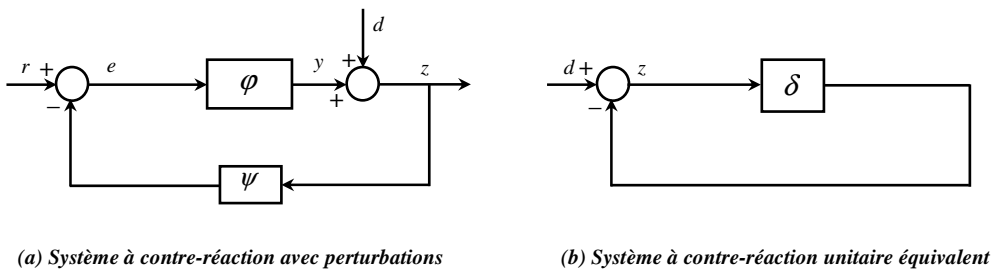
$$y \approx \psi^{-1}(r) \quad (1-3)$$

sous l'hypothèse que  $\psi^{-1}$  existe.

La structure à contre-réaction présente aussi une propriété très intéressante de robustesse vis-à-vis des erreurs de modélisation du système à commander. En effet, la relation (1-3) reste vraie aussi longtemps que la boucle fermée (relation (1-2)) possède une solution pour tout signal  $r$  et que le gain de la fonction  $\gamma$  est grand. La relation E/S de la chaîne de retour doit souvent être implantée avec précision. Ceci conduit à une bonne précision de la relation E/S du système à contre-réaction aussi longtemps que le gain est grand, même si le procédé est mal connu où s'il possède des propriétés peu favorables (cf. paragraphe 1.5.3).

La contre-réaction peut produire d'autres effets favorables que la robustesse, parmi lesquels l'amélioration de la bande passante, et l'atténuation des perturbations [Ous94]. L'amélioration de la bande passante est aussi une conséquence de la propriété de grand gain. Si le compensateur de retour est un gain unité, la relation E/S du système à contre-réaction est proche de l'unité pour les pulsations où le gain de boucle est important. Ceci augmente la bande passante du système bouclé.

En ce qui concerne l'atténuation des perturbations, considérons la figure 1.3 où  $d$  est une perturbation.



**Fig. 1.3 Configuration E/S d'une contre-réaction**

Si nous posons  $r = 0$ , le système à contre-réaction est alors décrit par l'équation :

$$z = d + \varphi(-\psi(z)) = d - \delta(z) \tag{1-4}$$

Par analogie avec la configuration de la figure 1.2(b), il s'ensuit que si le gain est important au sens que  $\|\delta(z)\| \gg \|z\|$ , nous avons :

$$\|z\| \ll \|d\| \tag{1-5}$$

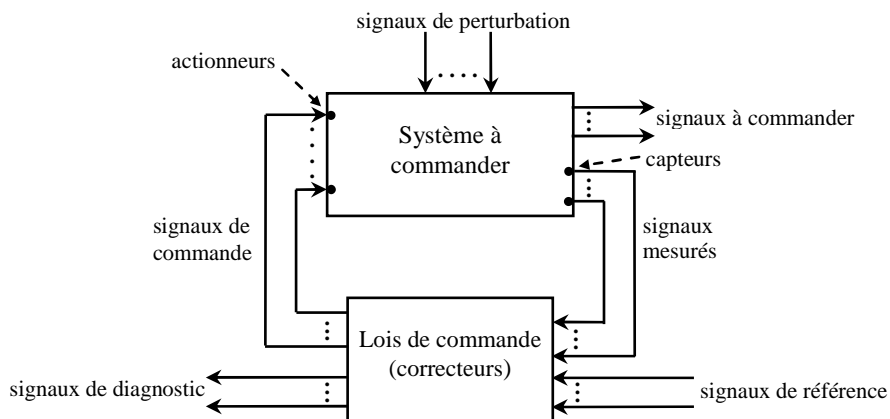
Cela signifie que la sortie  $z$  du système à contre-réaction est petite comparée à la perturbation  $d$ , et donc que l'effet de la perturbation est grandement réduit, Tout ceci reste vrai pourvu que l'équation (1-4) ait une solution bornée  $z$  quel que soit la perturbation  $d$  bornée.

Malgré tous les atouts exposés, l'utilisation d'une structure à contre-réaction nécessite quelques précautions car si on augmente naïvement le gain du système, on risque d'obtenir un système à contre-réaction instable. Et même si le système est stable, un grand gain risque d'aboutir à des entrées trop importantes, que le système à commander ne peut pas tolérer. Il en résultera une réduction du gain et une baisse de performance correspondante. D'autre part, la contre-réaction implique que l'on mesure la sortie au moyen d'un capteur. Les erreurs de mesure et bruits de mesure qui y sont associés risquent d'en diminuer la précision. Dans la suite de ce chapitre, ces points seront repris, avec plus de détails.

## 1.2. Vie d'un cahier des charges

Le formalisme d'un cahier des charges en Automatique est fortement lié à l'évolution du problème de commande, son élaboration représente une des étapes récurrentes d'un processus dynamique compliqué où le but est de synthétiser une loi de commande qui permet de satisfaire toutes les exigences.

Dans le cadre des applications et de la conception globale de systèmes, un cahier des charges n'est jamais définitivement fixé et peut évoluer au cours des évolutions du système ou des performances désirées. Par exemple, dans le but d'améliorer la conduite d'un système, l'ingénieur automaticien a souvent recours à augmenter la structure du système à commander par l'ajout de capteurs, contrôleurs et actionneurs.



*Fig. 1.4 Schéma fonctionnel d'un système de commande*

Le schéma fonctionnel général d'un système de commande est représenté par la figure 1.4. Ce schéma moderne est une généralisation du schéma d'origine. Dans de nombreuses formulations, les "signaux à commander" sont souvent représentatifs des erreurs et doivent être "minimisés". Dans le cadre d'une formulation générique pour l'optimisation, ce peut être également des sorties devant respecter un gabarit temporel ou fréquentiel, ou des contraintes variées (énergétiques par exemple).

Dans la suite, nous détaillerons la formalisation du problème de commande ainsi que les éventuelles démarches à suivre pour la validation d'un cahier des charges

## 1.3. Les étapes d'un problème de commande

La synthèse d'une loi de commande est rarement une opération directe : plusieurs étapes sont nécessaires pour obtenir un résultat satisfaisant. Voici un résumé ordonné de ces étapes :

**Etape 1. Modélisation et identification** : Pour agir sur un système, il est nécessaire de connaître son comportement et ses interactions avec le milieu extérieur c'est-à-dire le lien entre les différents signaux d'entrée (commande, perturbations, bruits) et les différents signaux de sortie (sortie à commander, mesure). Cette connaissance prend la forme d'un modèle mathématique quantitatif,

acquis théoriquement par un ensemble de lois physiques (modélisation) et/ou expérimentalement par une série d'observations E/S (identification). Le choix d'un bon modèle dépend essentiellement de sa complexité et sa précision à reproduire le comportement E/S. Cela signifie que l'on doit disposer d'un modèle mathématique réalisant un compromis entre sa fidélité de comportement qualitatif et quantitatif et sa simplicité de mise en œuvre à des fins d'analyse et de synthèse. Bien souvent, dans les exercices académiques par exemple, la modélisation s'arrête là. Pour avoir toutes les informations nécessaires à une reformulation générique, le modèle doit également comprendre des informations sur les classes de signaux susceptibles d'être appliqués en entrée (perturbations, bruits). Des scénarios test et des signaux de références doivent être également définis et relèvent souvent également de la modélisation.

**Etape 2. Etablissement d'un cahier des charges :** Le cahier des charges doit contenir la définition de l'ensemble des spécifications du système. Ces spécifications traduisent les performances désirées (objectifs) ainsi que toutes les contraintes liées à la configuration de la boucle fermée, fixée au préalable par le concepteur en fonction des performances (signaux désirés). Rien ne garantit a priori qu'avec le choix initial des objectifs de performance, des actionneurs et des capteurs une solution existe. Dans le but d'étudier l'existence, de faire la synthèse ou la retouche d'une loi de commande, le cahier des charges est traduit sous la forme de critères mathématiques.

**Etape 3. Synthèse ou retouche de la loi de commande :** Dans cette étape le but est de rechercher la loi de commande qui satisfait les critères traduisant le cahier des charges pour le modèle représentant le système réel. Cette recherche est généralement réalisée via une technique de commande choisie et exécutée par la suite par un ordinateur (simulation). L'objectif majeur étant de valider le cahier des charges pour le système réel et pas simplement pour le modèle manipulé, le concepteur doit alors assurer une bonne conduite du correcteur en essayant d'obtenir le plus de garanties a priori sur le bon fonctionnement du correcteur appliqué sur le système réel (robustesse).

**Etape 4. Implémentation et validation de la commande :** Après avoir synthétisé la loi de commande, le concepteur doit garantir sa mise en pratique sur le système réel. Une première étape de validation est tout d'abord effectuée en simulation afin de limiter les éventuels échecs. Sur les systèmes complexes, la validation met souvent à jour des imperfections et nécessite des retouches du processus de conception. Les différentes étapes précédentes sont alors revues et affinées.

- correction du modèle (paramètres mal identifiés, dynamiques négligées, point de fonctionnement)
- modification du cahier des charges
- correction ou "affinage" de la loi de commande

**Utilisation de la retouche de correcteur :**

Cette démarche peut être utilisée suite à une modification dans une des étapes ci-dessus. Dans le cas où les conditions de la synthèse sont modifiées, elle permet d'éviter le recours à de nouvelles synthèses qui peuvent être très coûteuses en reformulation et calculs, voir impossibles à mener par des méthodes classiques suite à une complexification du cahier des charges. Actuellement, c'est dans ce contexte que les outils de retouche de correcteurs sont en cours de développement. Ils sont d'une importance croissante pour certains systèmes, comme les avions, dont la conception même est de plus en plus imbriquée avec le développement des lois de commande qui les équiperont. Différentes circonstances peuvent motiver une retouche :

- modification du modèle de synthèse (par exemple amélioration de la connaissance du système suite à une campagne d'identification) ;
- propagation du réglage d'une loi faite autour d'un point de fonctionnement donné, à un autre point de fonctionnement ;
- introduction de nouveaux objectifs dans un cahier des charges. La retouche de correcteurs peut alors être considérée comme une étape dans une synthèse multicritère ;
- adaptation du correcteur à une évolution de sa structure (structure complexe ne pouvant être prise en compte par des approches classiques).

Lorsqu'on raisonne à structure de correcteur fixée, la retouche peut être vue comme une simple optimisation de gains. Elle est en fait beaucoup plus et peut être un outil puissant pour prendre progressivement en compte un cahier des charges complexe et permettre la mise en œuvre des structures de commande originales. Dans l'idéal, les procédures de retouche doivent travailler dans un voisinage du correcteur où le maintien de la stabilité est garanti, l'estimation de directions de modification du correcteur assurant le maintien de caractéristiques acquises (performance, robustesse) ou leur amélioration.

Les étapes ci-dessus, bien qu'elles soient toutes décisives pour une bonne conception de lois de commande, n'ont pas connu le même essor. La majorité des travaux de recherche en Automatique s'intéresse aux problèmes de modélisation et identification des systèmes ainsi qu'à la synthèse des lois de commande. Ceci est certainement dû à la difficulté de formuler "la conception d'un cahier des charges + loi de commande" d'une manière précise et générique.

Comme exemple de ce type de difficulté, nous pouvons citer le problème du choix de la structure de commande : c'est une des problématiques les plus difficiles lors de l'élaboration d'un cahier des charges qui peut avoir des conséquences directes sur les performances du système bouclé. En particulier, l'amélioration des performances observée en modifiant la structure de commande peut être beaucoup plus importante que celle observée en modifiant les réglages d'une loi de commande particulière.

## 1.4. Le problème fondamental de la commande

Le problème fondamental de la théorie de la commande peut être énoncé comme suite [Boy91] :

*Etant donné un système à commander, une configuration de commande et un cahier des charges donné, trouver une loi de commande qui répond aux spécifications ou prouver son inexistence.*

L'aspect décisif de ce problème fondamental ressort surtout de l'existence ou l'inexistence d'une loi de commande et donc, de manière équivalente, de la faisabilité ou non du cahier des charges avec une configuration de commande donnée.

## 1.5. Les spécifications d'un cahier des charges

Les objectifs d'un problème de commande sont souvent formulés sous forme d'un cahier des charges contenant toutes les spécifications désirées. En général, il est difficile de définir un cahier des charges pour un problème de commande, du fait d'une part de la méconnaissance de performances que peut atteindre le système bouclé, et d'autre part de la nature contradictoire de certains objectifs tels que les critères de performances et de robustesse. Pour un cas donné, le dimensionnement des spécifications est initialement issu d'une connaissance qualitative du système étudié. Le cahier des charges est ensuite retouché (durci ou relâché) à l'issue de quelques essais de synthèse. Pour ce faire, il faut formuler ses spécifications sous forme de critères mathématiques permettant de définir l'ensemble des signaux de sortie désirés, l'erreur tolérée entre les signaux réels de sortie et les signaux désirés, l'ensemble des signaux de commande admissibles, l'ensemble des signaux de perturbation, etc.

En général, on distingue trois types de spécifications : les spécifications en performances, les spécifications en robustesse et les spécifications de loi de commande. Afin d'introduire ces différents types de spécifications, la boucle fermée de la figure 1.4 sera caricaturée par un schéma fonctionnel plus simple qui récapitule avec suffisamment d'exhaustivité les différentes situations courantes (cf. figure 1.5). Sauf mention contraire, les notations de ce schéma bloc seront adoptées dans la suite de ce chapitre et tout au long de ce manuscrit.

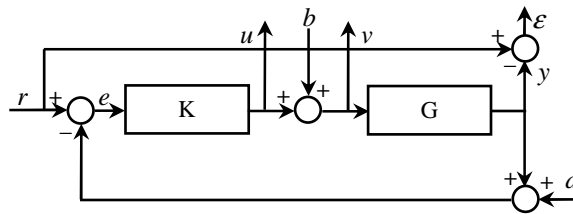


Fig. 1.5: Système et signaux en boucle fermée

Dans ce schéma fonctionnel, les blocs  $G$  et  $K$  désignent respectivement le système à commander et le correcteur. Dans le cas général, ils sont définis au sens des opérateurs. Dans le cas particulier des systèmes linéaires invariants,  $G$  et  $K$  sont explicitement définis par leurs matrices de fonctions de transfert et seront notés  $G(p)$  et  $K(p)$  où  $p$  désigne la variable de Laplace. Ces transferts sont définis respectivement par :  $G(p) = C(pI_n - A)^{-1}B + D$  et  $K(p) = C_k(pI_{n_k} - A_k)^{-1}B_k + D_k$  où les quadruples  $(A, B, C, D)$  et  $(A_k, B_k, C_k, D_k)$  définissent leurs formes d'état respectives :

$$G = \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \Leftrightarrow \underbrace{G = \begin{pmatrix} A & B \\ C & D \end{pmatrix}}_{\text{notation}} \quad (1-6)$$

$$K = \begin{cases} \dot{x}_k = A_k x_k + B_k e \\ u = C_k x_k + D_k e \end{cases} \Leftrightarrow \underbrace{K = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix}}_{\text{notation}} \quad (1-7)$$

On désigne par :

$x \in R^n$  le vecteur d'état du système à commander

$x_k \in R^{n_k}$  le vecteur d'état du correcteur

$u \in R^m$  la commande issue du correcteur  $K(p)$   
 $v \in R^m$  la commande effective  
 $y \in R^p$  le vecteur des signaux de sortie  
 $r \in R^p$  le signal de référence que doit suivre la sortie  $y$   
 $b \in R^m$  les perturbations affectant la commande  $u$   
 $d \in R^p$  les bruits de mesure

De manière générale, il s'agit de faire suivre, dans la mesure du possible, la sortie du système  $y(t)$  à un signal de référence  $r(t)$  appartenant à un ensemble bien défini. L'erreur de suivi de trajectoire est désignée par  $\varepsilon(t) = r(t) - y(t)$ . La façon dont la sortie du système suit la trajectoire de référence  $r(t)$  peut être exprimée par le fait que  $\varepsilon(t)$  doit appartenir à un ensemble bien déterminé.

Cette spécification doit être réalisée malgré la présence de perturbations  $b(t)$  qui agissent sur le système (ramenées en entrée de celui-ci) et des bruits  $d(t)$  sur la mesure de la sortie  $y(t)$ . Même si les perturbations et/ou les bruits ne sont pas mesurés, on sait a priori que ces signaux appartiennent à des ensembles déterminés.

De plus, la commande  $u(t)$  appliquée doit être raisonnable par rapport à l'application considérée (elle ne doit pas solliciter de façon trop importante les actionneurs). Ici encore, cela revient à dire que l'on a défini pour  $u(t)$  un ensemble admissible : la loi de commande doit assurer que  $u(t)$  appartient bien à cet ensemble. Cela constitue les objectifs de loi de commande.

Une propriété nécessaire (mais loin d'être suffisante) est la stabilité de la boucle fermée : pour des signaux d'entrée  $r(t)$ ,  $b(t)$  et  $d(t)$  d'amplitude finie, les signaux du système bouclé sont aussi d'amplitude finie.

Enfin, la loi de commande est construite à partir d'un modèle qui est une représentation idéalisée du système réel : les mesures des paramètres physiques sont toujours entachées d'incertitudes, les dynamiques rapides sont difficilement modélisables... Malgré toutes ces imperfections, la loi de commande doit fonctionner correctement sur le système réel c'est-à-dire assurer la stabilité et les performances recherchées (robustesse).

Dans ce qui suit, nous présentons les principaux outils d'analyse qui permettent de traduire les spécifications de commande en critères mathématiques. Par la suite, l'ingénieur automaticien aura la tâche de rechercher la meilleure formulation de ces critères et de synthétiser un algorithme de commande approprié à la formulation choisie.

### 1.5.1. La stabilité

La notion de stabilité est capitale en automatique ; elle représente une exigence critique dans la conception de système de commande. De nombreuses notions de stabilité existent, toutes ne sont pas calculables en temps fini (par exemple la définition générale de la stabilité au sens de Lyapunov n'est pas constructive) et certaines ne peuvent être utilisées de façon exactes lors d'une optimisation. Voyons quelques définitions courantes.

### 1.5.1.1. Stabilité entrée/sortie (stabilité externe)

Une première forme de stabilité pour les signaux et les systèmes est la stabilité externe qui consiste à évaluer le comportement des trajectoires suite à une perturbation externe (entrée non nulle). On parle, alors, de la stabilité EBSB (Entrée Bornée Sortie Bornée). D'une façon générale, un système E/S est dit stable de manière externe, si pour toute entrée bornée  $u$ , sa sortie  $y$  est bornée. En d'autres termes :

$$\|u\|_{\infty} = \sup_t |u(t)| < \infty \Rightarrow \|y\|_{\infty} = \sup_t |y(t)| < \infty \quad (1-8)$$

Dans ce cas,  $K(u) = \frac{\sup_t |y(t)|}{\sup_t |u(t)|}$  est défini pour toute entrée bornée, et si  $\sup_u K(u)$  est fini pour  $U = \{u : \sup_t |u(t)| < \infty\}$ , cette grandeur est appelée gain du système.

Cette définition ne peut évidemment pas être prise en compte directement par des contraintes sur des signaux types. Cette définition peut être approchée par l'appartenance de certains signaux à des gabarits. Pour que la contrainte par gabarit soit représentative il faut qu'elle porte sur des plages de temps suffisamment longues (par rapport aux temps de réponses du système) et que le système soit observable via les sorties considérées.

### 1.5.1.2. Stabilité externe des systèmes linéaires

**Définition 1.1 (temporelle)** Un système linéaire invariant à temps continu est EBSB (ou BIBO-stable) si et seulement si sa réponse impulsionnelle est absolument intégrable, i.e. si sa norme  $\ell^1$  existe (finie)

$$\int_0^{+\infty} |h(t)| dt = \|h(t)\|_1 < \infty \quad (1-9)$$

Cette condition dans le domaine temporel impose que la transformée de Laplace de la réponse impulsionnelle soit définie en 0. Par conséquent, 0 est dans le domaine de convergence de  $H(p)$ .

En ce qui concerne son utilisation pour l'optimisation : même type de remarque que pour la stabilité E/S.

**Définition 1.2 (fréquentielle)** Soit le système linéaire invariant à temps continu défini par la matrice de fonctions de transfert rationnelles irréductibles  $G(p)$ . Le système est EBSB-stable si et seulement si tous les pôles de  $G(p)$  sont à partie réelle négative :  $\text{Re}(p_i) < 0 \quad \forall i$ .

Ce critère peut être estimé en temps fini et de façon très efficace. Il peut être utilisé en optimisation non linéaire. Malheureusement il ne concerne que les systèmes linéaires.

### 1.5.1.3. La stabilité au sens de Lyapunov (stabilité interne)

Contrairement à la notion de stabilité EBSB qui s'intéresse seulement au comportement externe du système et exige que l'énergie des signaux en sortie  $y$  soit bornée dès que l'énergie fournie en entrée  $u$  est bornée, la stabilité interne exige que tous les signaux du système  $y$  compris les états internes soient stables. Cette seconde notion est donc plus restrictive et implique la stabilité externe.

Soit le système autonome continu de dimension finie décrit par l'équation différentielle non-linéaire :

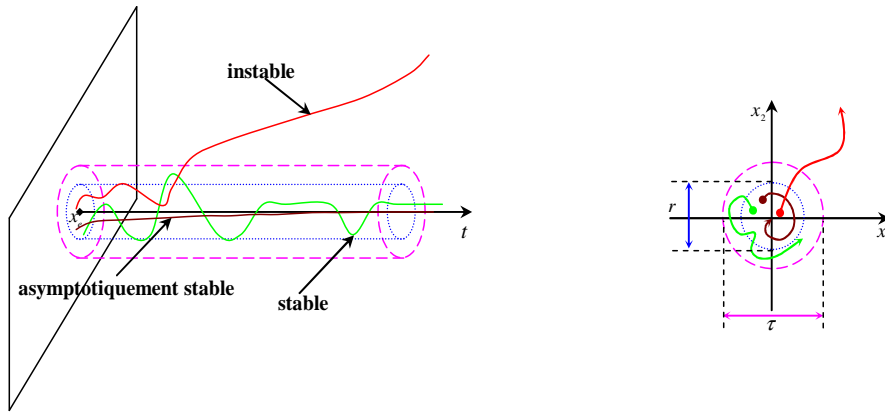


$$\dot{x} = f(x) \quad x \in \mathfrak{R}^n \quad (1-10)$$

Un vecteur  $x_e \in \mathfrak{R}^n$  est dit point ou état d'équilibre du système (1-10) si  $f(x_e) = 0$ . Ce point d'équilibre peut être ramené à l'origine par un simple changement de variable  $x \leftarrow x - x_e$ . Donc, sans perte de généralité, les définitions et les théorèmes qui suivent seront établis en considérant  $x_e = 0$ . Une question importante est de connaître le comportement des trajectoires initialement voisines de l'équilibre; pour cela, on définit la notion de stabilité interne d'un point d'équilibre au sens de Lyapunov [Kha02].

**Définition 1.3 (stabilité locale)** L'état d'équilibre  $x_e = 0$  du système (1-10) est localement:

- stable, si pour tout  $\tau > 0$ , il existe  $r = r(\tau)$ , tel que :  $\|x(t=0)\| < r \Rightarrow \|x(t)\| < \tau \quad \forall t > 0$ .
- asymptotiquement stable, s'il est stable et si  $r$  peut être choisi tel que :  $\|x(t=0)\| < r \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0$ .
- instable si non stable.



**Fig. 1.6: Stabilité interne au sens de Lyapunov**

Conceptuellement, la stabilité au sens de Lyapunov garantit que la trajectoire  $x(t)$  dans l'espace d'état restera à l'intérieur de la boule  $B(x_e, \tau)$  si son point de départ appartient à une boule  $B(x_e, r)$ . La stabilité asymptotique inclut cette propriété, mais spécifie, de plus, que toute trajectoire initialisée dans la boule  $B(x_e, r)$  converge vers  $x_e$  (cf. figure 1.6).

**Définition 1.4 (stabilité globale)** Si le point d'équilibre  $x_e$  est asymptotiquement stable quel que soit le vecteur d'état initial  $x(t=0)$  alors ce point d'équilibre est globalement asymptotiquement stable.

La méthode dite directe de Lyapunov consiste donc à générer pour un système donné une fonction scalaire de type énergétique qui admet une dérivée temporelle négative. Deux théorèmes de stabilité sont alors énoncés.

**Théorème 1.1 (stabilité locale)** Un état d'équilibre  $x_e$  est localement stable s'il existe une fonction continument dérivable  $V(x)$  appelée fonction de Lyapunov telle que :

- (1)  $V(0) = 0$
- (2)  $V(x) > 0 \quad \forall x \neq 0, x \in \Omega$  (définie positive)
- (3)  $\dot{V}(x) \leq 0 \quad \forall x \neq 0, x \in \Omega$  (défini semi négative)

De la même manière nous annonçons le théorème de stabilité globale au sens de Lyapunov :

**Théorème 1.2 (stabilité globale)** Un état d'équilibre  $x_e$  est globalement asymptotiquement stable s'il existe une fonction de Lyapunov  $V(x)$  telle que :

- (1)  $V(x) = 0$
- (2)  $V(x) > 0 \quad \forall x \neq 0$
- (3)  $\dot{V}(x) < 0 \quad \forall x \neq 0$
- (4)  $\dot{V} \rightarrow -\infty$  lorsque  $\|x\| \rightarrow \infty$

Bien qu'il annonce une condition suffisante de stabilité, ce théorème ne représente pas un résultat très constructif parce qu'il ne donne aucune indication quant à la construction d'une fonction de Lyapunov. On ne peut donc conclure sur l'inexistence d'une telle fonction. L'application générale n'est pas possible pour une approche par optimisation. Par contre, l'utilisation indirecte est envisageable : par exemple les méthodes de type Backstepping permettent de mettre au point des corrections garantissant la stabilité. Le problème d'optimisation porte alors sur une optimisation paramétrique pouvant améliorer la performance et la robustesse.

#### 1.5.1.4. La passivité

Ce paragraphe consiste à étendre le concept d'énergie à une plus large classe de systèmes. Soit

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (1-11)$$

La présence, à la fois d'une entrée et d'une sortie de dimension compatible, complique les interprétations. Pour ce type de système, l'interprétation de type Lyapunov ne s'applique plus. En effet, bien que l'énergie puisse avoir une tendance à décroître au cours du temps avec une entrée nulle, il serait tout à fait possible, en utilisant la nouvelle entrée à disposition, de l'augmenter de manière arbitraire. Dans d'autres cas, lors d'une connexion entre systèmes, l'énergie peut passer d'un système à un autre suivant une sorte de résonance, bien que chaque système pris isolément fasse décroître son énergie propre lorsque la connexion est rompue.

Toutefois, il existe une classe de système, les systèmes passifs, pour lesquels un certain type de comportement se trouve maintenu quelles que soient les connexions entre ces systèmes. Cette propriété est remarquable.

Le système (1-11) possède à la fois une entrée  $u$  et une sortie  $y$ . Un système est passif si, lorsque de la puissance est soutirée, le soutirage se fait au détriment du stock interne d'énergie. Ce stock est en quelque sorte l'analogue de la fonction de Lyapunov et vérifie les mêmes propriétés. On la notera  $V$ .

**Définition 1.5 (Passivité)** Soit le système (1-11). S'il existe  $\gamma > 0$ , et une fonction de stockage  $V > \gamma$ , tels que :

$$\dot{V} = u^T y - g \quad (1-12)$$

avec  $g \geq 0$ , alors le système est passif.

L'immense avantage de la notion de systèmes passifs est sa souplesse vis-à-vis des connexions [Kha02]. Les connexions en série, en parallèle et même en rétroaction conservent la passivité globale du système. Ce dernier cas est important lors de l'association de sous-systèmes passifs en retour de

sortie par exemple. Il est également possible de donner une définition équivalente de la passivité qui ne fait pas intervenir de notion différentielle.

**Définition 1.6 (Passivité)** Si  $\exists \alpha \in \mathbb{R}, \alpha > -\infty$  ;

$$\int_0^{\infty} u(\tau)y(\tau)d\tau > \alpha \quad (1-13)$$

alors le système est passif.

L'intérêt de la notion de passivité sera abordé avec plus de détails dans le cadre des systèmes linéaires et plus particulièrement pour la robustesse en stabilité de ces systèmes.

### 1.5.1.5. Stabilité interne des systèmes linéaires continus invariants

On rappelle qu'en temps continu,  $\dot{x} = Ax, x(0) = x_0 \Leftrightarrow x(t) = e^{(tA)}x_0$ . Ce système admet un ou une infinité de points d'équilibre selon que  $A$  soit inversible ou non. Dans la suite de ce document, on s'intéressera à la stabilité du seul point d'équilibre  $x_e = 0$ . Par conséquent, par abus de langage, on parlera de la stabilité du système au lieu de parler de la stabilité du point d'équilibre.

Afin de déduire les conditions de stabilité des systèmes linéaires, on introduit la forme diagonale de Jordan qui permet, via un changement de base, de représenter la matrice d'état  $A$  sous une forme dite similaire, plus facile à analyser.

Le théorème qui suit est énoncé pour les systèmes linéaires invariants autonomes, il peut être étendu pour les systèmes linéaires invariants avec entrées exogènes sans aucune modification.

**Théorème 1.3** La stabilité du système dynamique  $\dot{x} = Ax$  est donnée par :

- si  $\exists i, \text{Re}(\lambda_i) > 0$  alors le système est instable.
- sinon,  $\forall i, \text{Re}(\lambda_i) \leq 0$ 
  - si  $\forall i, \text{Re}(\lambda_i) < 0$ , alors le système est asymptotiquement stable.
  - si  $\forall j, \text{Re}(\lambda_j) = 0 \Rightarrow v_j = 1$ , alors le système est stable, sans être asymptotiquement stable.
  - si  $\exists j, \text{Re}(\lambda_j) = 0$  et  $v_j > 1$ , alors
    - si les blocs de Jordan associés à  $\lambda_j$  sont scalaires, alors le système est stable.
    - si un des blocs de Jordan associés à  $\lambda_j$  est non scalaire, alors le système est instable.

Selon le théorème 1.2, nous pouvons énoncer le théorème de stabilité de Lyapunov pour les systèmes linéaires continus stationnaires, les conditions du théorème 1.2 deviennent alors nécessaires et suffisantes et les propositions suivantes sont équivalentes :

- $\dot{x} = Ax$  est asymptotiquement stable (les valeurs propres de  $A$  sont à partie réelles négatives).
- $\exists P = P^T > 0$  telle que  $A^T P + PA < 0$ .
- $\forall Q = Q^T > 0, \exists P = P^T > 0$  telle que  $A^T P + PA + Q = 0$  ( $P$  est alors unique).

Notons que dans le cas présent les différents critères de stabilité interne des systèmes linéaires continus invariants se calculent très bien et sont utilisables sans problème en optimisation non linéaire.

### 1.5.1.6. Critères de stabilité des systèmes linéaires

Dans ce paragraphe, nous rappelons les principaux critères de la stabilité externe (EBSB) des systèmes linéaires à temps invariants. La richesse et la variété des critères pour cette catégorie de système peut et doit être exploitée. Par exemple, pour la retouche de correcteurs, elle peut être utilisée sur des linéarisés du modèle non linéaire.

#### Critère de Routh-Hurwitz

Le critère de Routh-Hurwitz est un critère algébrique qui permet de déterminer l'existence de pôles à partie réelle positive à partir de l'étude des coefficients du dénominateur de la matrice de transfert (donc du déterminant de  $pI - A$ ), et ceci sans expliciter ses pôles. Considérons le polynôme caractéristique de degré  $n$  et à coefficients constants décrit par :

$$P_A(p) = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0 \quad (1-14)$$

On forme le tableau suivant :

$$\begin{array}{l|llll}
 p^n & a_n & a_{n-2} & a_{n-4} & \dots & 0 \\
 p^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \dots & 0 \\
 p^{n-2} & x_1 & x_2 & x_3 & \dots & 0 \\
 p^{n-3} & y_1 & y_2 & y_3 & \dots & 0 \\
 \dots & \dots & \dots & \dots & \dots & 0 \\
 p^1 & \dots & \dots & 0 & & \\
 p^0 & \dots & 0 & & & 
 \end{array} \quad (1-15)$$

$$\text{avec : } x_1 = \frac{-1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}, x_2 = \frac{-1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}, \dots, y_1 = \frac{-1}{x_1} \begin{vmatrix} a_{n-1} & a_{n-3} \\ x_1 & x_2 \end{vmatrix}, y_2 = \frac{-1}{x_1} \begin{vmatrix} a_{n-1} & a_{n-5} \\ x_1 & x_3 \end{vmatrix} \dots \text{etc.}$$

Pour que le système soit stable, il suffit que tous les éléments de la première colonne de la table de Routh soient de même signe. Dans le cas contraire, le nombre de racines instables est égale au nombre de changements de signe dans la première colonne de la table de Routh.

Cette approche (et ses nombreuses variantes) permet de transformer une demande de stabilité en un ensemble de contrainte sur les coefficients. Les contraintes sont alors des inégalités entre des polynômes à plusieurs indéterminés, facilement manipulables et différentiable. Ce type de critère est donc pleinement exploitable pour les approches par optimisation générales.

#### Critère de Nyquist

Il s'agit d'un critère graphique dans le domaine fréquentiel qui permet de conclure à la stabilité externe d'un système linéaire en boucle fermée par la seule connaissance de sa boucle ouverte. Il est d'une très grande importance pratique car il utilise une représentation graphique de la réponse fréquentielle qui peut être déduite de l'expérience.

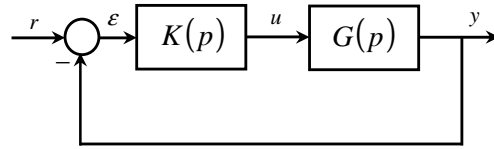


Fig. 1.7: Système de commande à retour unitaire

Considérons la boucle de commande multivariable représentée par la figure 1.7. La matrice des transferts en boucle ouverte en sortie est donnée par  $L(p) = G(p)K(p)$ , et celle en boucle fermée en sortie est donnée par  $H_{r \rightarrow y}(p) = (I_p + L(p))^{-1}L(p)$ . Selon la définition 1.2, le système bouclé est stable si et seulement si les racines du polynôme caractéristique de la boucle fermée sont strictement négatives. D'une manière équivalente, les zéros du dénominateur  $\det(I_p + L(p))$  de la boucle fermée doivent être dans le demi-plan gauche du plan complexe. Le critère de stabilité de Nyquist permet d'établir le nombre de pôles instables du système bouclé directement du lieu de Nyquist de  $\det(L(p))$ . Rappelons que le lieu de Nyquist de  $\det(I_p + L(p))$  correspond à l'image de  $\det(I_p + L(p))$  quand  $p$  parcourt le contour de Nyquist (figure 1.8) dans le sens anti-trigonométrique.

Le critère de Nyquist est basé sur le théorème de Cauchy, il peut être énoncé dans le cas continu comme suit :

**Théorème 1.4 (Critère de Nyquist)** Dans le cas où la matrice de fonctions de transfert  $L(p)$  ne possède pas de pôles imaginaires purs<sup>1</sup>, le système bouclé décrit par  $H_{r \rightarrow y}(p)$  est stable au sens EBSB si et seulement si l'image du contour de Nyquist (figure 1.8) par la fonction complexe  $\det(I_p + L(p))$  ne passe pas par l'origine et l'entoure dans le sens trigonométrique un nombre de fois égal au nombre de pôles instables en boucle ouverte.

Le critère de Nyquist n'est pas utilisable directement lors des optimisations. Ce sont les notions de marges qui sont le plus souvent mises en pratique.

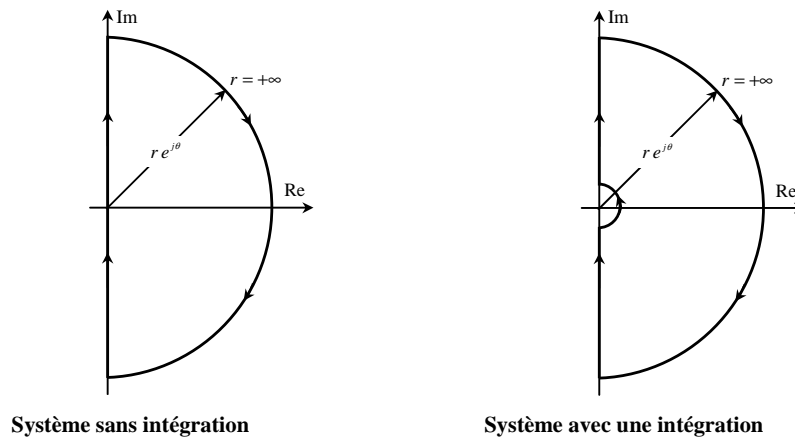


Fig. 1.8: Contour de Nyquist

<sup>1</sup> Dans le cas où le transfert possède des pôles imaginaires purs, le critère précédent s'applique après une modification technique qui ne sera pas développée ici. Pour plus de détails, voir par exemple [Fra86].

***Théorème du petit gain***

Un autre résultat important pour la stabilité interne des systèmes linéaires est celui basé sur le théorème du petit gain (Small Gain Theorem). Une version simplifiée de ce théorème dans le cas des systèmes linéaires continus et invariants peut être formulée comme suit :

***Théorème 1.5 (Théorème du petit gain)*** *Etant donné un système de commande à retour unitaire EBSB stable en boucle ouverte (figure 1.7), le système en boucle fermée est stable intérieurement si :*

$$\|L(j\omega)\|_{\infty} = \sup_{\omega \geq 0} \bar{\sigma}(L(j\omega)) < 1 \quad \forall \omega \tag{1-16}$$

où  $\bar{\sigma}$  est la valeur singulière maximale de la boucle ouverte  $L(j\omega)$ .

Il fournit à la stabilité interne une condition suffisante qui est très simple mais aussi souvent très pessimiste. Sa condition est plus forte que celle du théorème de Nyquist car elle n'inclut aucune information concernant la phase.

***Critère de passivité***

Les définitions de la passivité (définitions 1.5 et 1.6), s'appliquent aussi bien aux systèmes linéaires que non-linéaires. Pour les systèmes linéaires, cette propriété peut se caractériser en fonction de la caractéristique fréquentielle.

***Théorème 1.6 (Positivité des systèmes linéaires)*** *Soit  $G(p)$  un système linéaire stationnaire multivariable et stable. Ce système est dit positif si et seulement si l'une des conditions équivalentes suivantes est vérifiée :*

$$(i) \quad \forall T > 0, \int_0^T y^T(t)u(t)dt \geq 0 \tag{1-17}$$

$$(ii) \quad G(p)+G^*(p) \geq 0 \text{ pour tout } p \text{ tel que } \text{Re}(p) \geq 0 \tag{1-18}$$

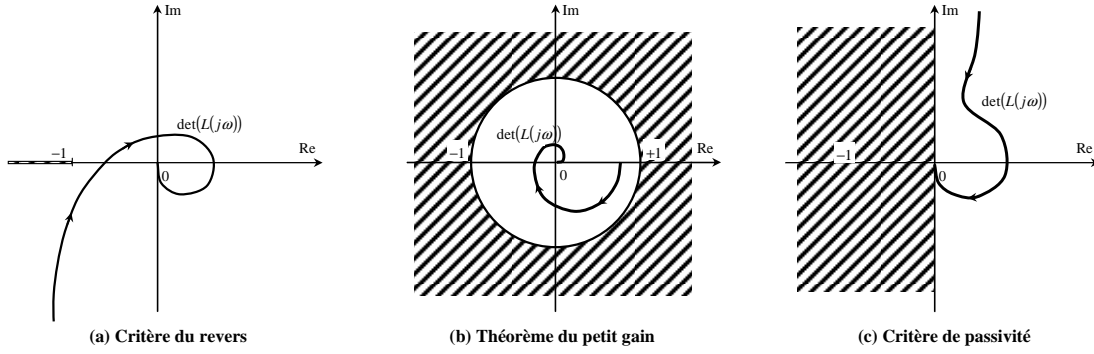
$$(iii) \quad \|(I+G)^{-1}(I-G)\|_{\infty} \leq 1 \tag{1-19}$$

La clé de la démonstration de ce théorème repose sur la définition 1.6 et le théorème de Parseval qui stipule que le bilan de la puissance sur tout l'horizon temporel est égal au bilan sur tout le spectre.

Notons que dans le cas des systèmes monovariables, la condition (ii) se traduit par  $\text{Re}(G(p)) \geq 0$ . Cette condition nécessaire et suffisante et la propriété d'association des systèmes passifs<sup>2</sup> permettent de définir une condition forte de stabilité en boucle fermée en contraignant la boucle ouverte  $L(j\omega)$  à être seulement dans le demi-plan droit du plan complexe (figure 1.9).

Ainsi, la positivité peut être employée comme un puissant critère de stabilité en boucle fermée. Certes, il est moins pessimiste que le Théorème du petit gain mais il reste toujours plus restrictif que le critère de Nyquist (figure 1.9). Ainsi, malgré son caractère conservatif, la positivité est largement utilisée dans le cadre des systèmes incertains et devient particulièrement efficace en présence de modes souples à amortissement incertain dans le système (cf. théorème 1.10).

<sup>2</sup> Le système issu d'une mise en série, en parallèle ou en rétroaction de deux systèmes passifs reste passif [Sch00].



**Fig. 1.9: Critères de stabilité graphiques.**

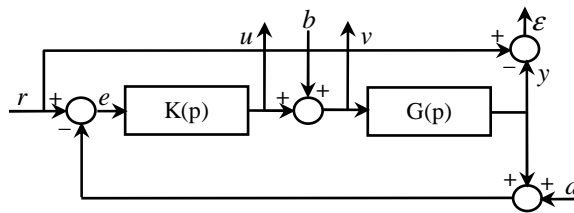
Pour certaines applications, des demandes de positivité peuvent être faites sur des zones spectrales en utilisant des contraintes pondérées du type :

$$\|W_1(I+G)^{-1}(I-G)W_2\|_{\infty} \leq \gamma \quad \text{ou} \quad \max_{\omega_1 < \omega < \omega_2} \bar{\sigma}(W_1(j\omega)(I+G(j\omega))^{-1}(I-G(j\omega))W_2(j\omega)) \leq \gamma$$

### 1.5.2. Les spécifications en performances

Les spécifications en performances décrivent qualitativement le fonctionnement souhaité du système bouclé sous l'effet du contrôleur. Elles représentent l'objectif majeur du problème de commande ainsi qu'un élément moteur pour l'évolution du cahier des charges. Nous montrerons ici que les performances peuvent s'exprimer sous forme de gabarits temporel et/ou fréquentiel qui peuvent être utilisés pour formuler les problèmes de retouche de correcteur.

Afin de pouvoir introduire ces spécifications pour les systèmes linéaires à temps continu, nous considérons le schéma bloc 1.10.



**Fig. 1.10: Schéma fonctionnel d'une boucle de régulation**

Ici, on différencie  $u$  la commande fournie à la sortie du correcteur et  $v$  la commande effectivement reçue à l'entrée du système à commander  $G$ . En notant  $T_{\alpha \rightarrow \beta}$  la fonction de transfert du signal d'entrée  $\alpha$  vers le signal de sortie  $\beta$ , on a les relations suivantes entre les sorties et les entrées du système :

$$\begin{cases} \varepsilon(p) = T_{r \rightarrow \varepsilon}(p) r(p) + T_{b \rightarrow \varepsilon}(p) b(p) + T_{d \rightarrow \varepsilon}(p) d(p) \\ u(p) = T_{r \rightarrow u}(p) r(p) + T_{b \rightarrow u}(p) b(p) + T_{d \rightarrow u}(p) d(p) \\ v(p) = T_{r \rightarrow v}(p) r(p) + T_{b \rightarrow v}(p) b(p) + T_{d \rightarrow v}(p) d(p) \\ y(p) = T_{r \rightarrow y}(p) r(p) + T_{b \rightarrow y}(p) b(p) + T_{d \rightarrow y}(p) d(p) \end{cases} \quad (1-20)$$

Nous définissons dans le tableau suivant les notations pour les différents transferts de la boucle de régulation.

Entrée \ Sortie	Référence $r$	Perturbation $b$	Bruit de mesure $d$
Erreur $\varepsilon$	$T_{r \rightarrow \varepsilon} = S = (I_p + GK)^{-1}$	$T_{b \rightarrow \varepsilon} = -GS_e$	$T_{d \rightarrow \varepsilon} = T$
Commande calculée $u$	$T_{r \rightarrow u} = KS$	$T_{b \rightarrow u} = -T_e = -KG(I_m + KG)^{-1}$	$T_{d \rightarrow u} = -KS$
Commande effective $v$	$T_{r \rightarrow v} = KS$	$T_{b \rightarrow v} = S_e = (I_m + KG)^{-1}$	$T_{d \rightarrow v} = -KS$
Sortie du système $y$	$T_{r \rightarrow y} = T = GK(I_p + GK)^{-1}$	$T_{b \rightarrow y} = GS_e$	$T_{d \rightarrow y} = -T$

**Tab. 1.1: Les transferts de la boucle fermée**

$S$  et  $T$  (respectivement  $S_e$  et  $T_e$ ) sont la sensibilité et la sensibilité complémentaire en sortie (respectivement en entrée). Nous constatons que les transferts de la boucle peuvent être définis par seulement six transferts dans le cas multivariable ;  $S, S_e, T, T_e, KS, GS_e$  et seulement quatre pour le cas monovariable ;  $S, T, KS, GS$  (les opérateurs commutent).

Dans un premier temps, la formulation des performances dans le cas des systèmes linéaires monovariés est présentée. Le cas des systèmes multivariables sera discuté à la fin de ce paragraphe

### 1.5.2.1. Suivi de trajectoire de référence (consigne)

Il s'agit d'évaluer la précision du système en boucle fermée en étudiant l'influence du signal de référence  $r(t)$  sur le signal d'erreur  $\varepsilon(t)$ . Le signal d'erreur  $\varepsilon(t)$  est caractérisé par son régime permanent et par son régime transitoire. Pour cela, on va considérer les classes de signaux usuels : échelon, rampe, sinusoïde ainsi que les signaux définis par leurs modules dans le domaine fréquentiel.

#### En régime permanent

Théoriquement, le régime permanent correspond à l'état des différents signaux du système quand le temps est infiniment grand. Dans le cas d'un suivi de référence, le but est de ramener l'erreur de suivi à zéro, c'est à dire :  $\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{t \rightarrow \infty} (y(t) - r(t)) = 0$ . Dans le cas des systèmes linéaires continus, l'étude de la précision se base essentiellement sur le théorème de la valeur finale ( $\lim_{t \rightarrow +\infty} \varepsilon(t) = \lim_{p \rightarrow 0} p\varepsilon(p)$ ). En pratique, le régime permanent correspond au comportement du système après un temps fini  $T_e$  appelé temps d'établissement :

$$T_e = \inf_{T > 0} \left\{ T \mid \forall t > T : |\varepsilon(t)| \leq \alpha |r(t)| \right\} \quad (1-21)$$

La stationnarité dans cette définition est chiffrée en terme d'une erreur relative maximale  $\alpha$  souvent choisie entre 0.01 et 0.05 selon la précision requise.

Dans le cas des systèmes linéaires continus, l'étude de la précision se base essentiellement sur le théorème de la valeur finale que nous rappelons ci-dessous :

**Théorème 1.7 (Théorème de la valeur finale)** Si tous les pôles de  $p\varepsilon(p)$  sont dans le demi plan gauche (axe imaginaire exclu) alors :

$$\lim_{t \rightarrow +\infty} \varepsilon(t) = \lim_{p \rightarrow 0} p\varepsilon(p) \quad (1-22)$$



Pour un signal de référence de type échelon et afin d'assurer une erreur de suivi nulle, le transfert en boucle fermée  $T_{r \rightarrow \varepsilon}(p) = S(p)$  doit contenir au moins un zéro en zéro. Plus particulièrement, si le système à commander ne possède pas un pôle à l'origine, il faut introduire un pôle intégrateur dans le transfert  $K(p)$ . Autrement,  $\lim_{t \rightarrow \infty} \varepsilon(t) = |T_{r \rightarrow \varepsilon}(0)| = \varepsilon_{stat} \neq 0$ , cette erreur  $\varepsilon_{stat}$  est appelée erreur statique.

De même, pour des entrées en forme de rampes, le suivi est assuré dans le cas où  $T_{r \rightarrow \varepsilon}(p)$  contient au moins deux zéros en zéro. Si  $T_{r \rightarrow \varepsilon}(p)$  ne contient qu'un seul zéro alors  $\lim_{t \rightarrow +\infty} \varepsilon(t) = \varepsilon_{trainage}$ , cette erreur  $\varepsilon_{trainage}$  est appelée erreur de trainage.

D'une manière générale, pour les signaux dont la transformée de Laplace s'écrit  $r(p) = A/p^k$  et  $k \geq 3$ , le système en boucle fermée sera capable de suivre ces signaux s'il est déjà capable de suivre les signaux d'ordre  $k-1$  et l'erreur de suivi tendra vers zéro si  $T_{r \rightarrow \varepsilon}(p)$  contient au moins  $k$  zéros en zéro.

Dans le cas d'un signal de référence sinusoïdal  $r(t) = A \sin(\omega_0 t)$ , le théorème de la valeur finale ne s'applique pas. Cependant, on sait que la réponse harmonique d'un système dynamique stable à un signal sinusoïdal tend vers le régime permanent suivant :  $\varepsilon(t) = A |T_{r \rightarrow \varepsilon}(j\omega_0)| \sin(\omega_0 t + \arg(T_{r \rightarrow \varepsilon}(j\omega_0)))$ . Par suite, pour assurer un suivi de référence parfait de ce signal ( $\lim_{t \rightarrow +\infty} \varepsilon(t) = 0$ ), il est nécessaire d'avoir  $|T_{r \rightarrow \varepsilon}(j\omega_0)| = 0$ , ce qui revient à dire que  $T_{r \rightarrow \varepsilon}(p)$  possède deux zéros complexes conjugués en  $\pm j\omega_0$ . Sinon, on peut toujours essayer de rechercher une erreur faible d'amplitude maximale  $\varepsilon_{sin}$  en imposant  $|T_{r \rightarrow \varepsilon}(j\omega_0)| \leq \varepsilon_{sin}$ .

Le suivi de trajectoire peut aussi être analysé dans le domaine fréquentiel pour les signaux définis par le module de leur transformée de Fourier. En effet, chaque signal temporel de la boucle de commande peut être vu comme une somme infinie pondérée de signaux sinusoïdaux : pour chaque pulsation  $\omega$ ,  $|\varepsilon(j\omega)|$  apparaît comme le poids du signal sinusoïdal de pulsation  $\omega$  dans le signal  $\varepsilon(t)$ .

$$\varepsilon(t) = \int_{-\infty}^{+\infty} |\varepsilon(j\omega)| e^{j(\omega t + \arg(\varepsilon(j\omega)))} d\omega \quad (1-23)$$

Le signal d'erreur  $\varepsilon$  est associé via le transfert  $T_{r \rightarrow \varepsilon}(j\omega)$ , au signal de référence  $r$  par la relation :  $|\varepsilon(j\omega)| = |T_{r \rightarrow \varepsilon}(j\omega)| |r(j\omega)|$ . Ainsi, pour modifier  $\varepsilon(t)$ , il est possible de choisir le module  $|T_{r \rightarrow \varepsilon}(j\omega)|$ , à travers le choix du correcteur  $K(j\omega)$ , de manière à façonner  $|\varepsilon(j\omega)|$  connaissant  $|r(j\omega)|$ .

### En régime transitoire

Le suivi est d'autant mieux assuré que l'erreur  $\varepsilon(t)$  est proche de zéro pour les signaux de référence qui nous intéressent. On peut définir un signal d'erreur  $\varepsilon(t)$  faible comme appartenant à un ensemble défini par :

$$E = \{ \varepsilon(j\omega) \text{ tel que } |\varepsilon(j\omega)| \leq \varepsilon_{sup} A \} \quad (1-24)$$

où  $A$  caractérise la taille du signal d'entrée (par exemple, l'amplitude de l'échelon) et  $\varepsilon_{sup}$  un petit scalaire positif. Il s'agit ici de normaliser l'erreur  $\varepsilon$  par une grandeur caractéristique du signal d'entrée  $r$ . Par exemple, dans le cas où le signal de référence est un signal en forme d'échelon :

$$\forall \omega, |\varepsilon(j\omega)| \leq \varepsilon_{sup} A \Leftrightarrow \forall \omega, \left| T_{r \rightarrow \varepsilon}(j\omega) \frac{A}{j\omega} \right| \leq \varepsilon_{sup} A \Leftrightarrow \forall \omega, |T_{r \rightarrow \varepsilon}(j\omega)| \leq \varepsilon_{sup} |j\omega| \quad (1-25)$$

On peut donc exprimer que l'on recherche un signal d'erreur  $\varepsilon(t)$  appartenant à un certain ensemble pour un signal de référence donné  $r(t)$  par une contrainte sur le module de la fonction  $T_{r \rightarrow \varepsilon}(j\omega)$ .

Cela suggère donc de généraliser la démarche précédente. Dans les méthodes fréquentielles, la caractérisation d'un ensemble de signaux  $s(t)$  se fait via leur spectre fréquentiel. Pour cela, on introduit une fonction de transfert  $W_s(p)$  dite de pondération. On peut alors définir un ensemble de signaux comme

$$\{s(j\omega) \text{ tel que } |s(j\omega)| \leq |W_s(j\omega)|\}, \text{ ou encore comme } \left\{s(j\omega) \text{ tel que } |s(j\omega)| \leq \frac{1}{|W_s(j\omega)|}\right\} \quad (1-26)$$

Par commodité, la première définition est utilisée pour définir les ensembles de signaux d'entrée (en l'occurrence, ici,  $r(t)$ ) et la seconde pour définir les ensembles de signaux de sortie du système bouclé (en l'occurrence, ici,  $\varepsilon(t)$ ) [Sco03].

Dans notre cas particulier, on a le signal d'entrée  $r$  qui peut être défini par l'introduction d'une fonction de transfert  $W_r$  telle que, pour les signaux de référence  $r$  qui nous intéressent, nous avons :

$$R = \{r(j\omega) \text{ tel que } |r(j\omega)| \leq |W_r(j\omega)|\} \quad (1-27)$$

avec  $W_r(j\omega) = A/j\omega$ . Pour le signal d'erreur  $\varepsilon$ , on désire qu'il appartienne à un ensemble défini par :

$$E = \left\{\varepsilon(j\omega) \text{ tel que } |\varepsilon(j\omega)| \leq \frac{1}{|W_\varepsilon(j\omega)|}\right\}, \text{ avec } W_\varepsilon(j\omega) = \frac{1}{\varepsilon_{\text{sup}} A} \quad (1-28)$$

Par la suite, la sortie vérifiera la spécification désirée si, pour tout signal de référence  $r \in R$ , on a  $\varepsilon(j\omega) = T_{r \rightarrow \varepsilon}(j\omega)r(j\omega) \in E$ , c'est-à-dire si :

$$\begin{aligned} \forall r(j\omega) \in R, \forall \omega, |\varepsilon(j\omega)| \leq \frac{1}{|W_\varepsilon(j\omega)|} &\Leftrightarrow \forall r(j\omega) \in R, \forall \omega, |T_{r \rightarrow \varepsilon}(j\omega)r(j\omega)| \leq \frac{1}{|W_\varepsilon(j\omega)|} \\ &\Leftrightarrow \forall \omega, |T_{r \rightarrow \varepsilon}(j\omega)W_r(j\omega)| \leq \frac{1}{|W_\varepsilon(j\omega)|} \\ &\Leftrightarrow \forall \omega, |W_\varepsilon(j\omega)T_{r \rightarrow \varepsilon}(j\omega)W_r(j\omega)| \leq 1 \end{aligned} \quad (1-29)$$

Dans le cas où les pondérations  $W_r(p)$  et  $W_\varepsilon(p)$  sont stables, ceci se note :

$$\|W_\varepsilon T_{r \rightarrow \varepsilon} W_r\|_\infty \leq 1 \quad (1-30)$$

La contrainte faisant intervenir la norme  $H_\infty$  s'interprète donc comme une contrainte sur la forme du module de la fonction de transfert  $T_{r \rightarrow \varepsilon}$ . Pour que la spécification de suivi de trajectoire soit satisfaite (c'est-à-dire que  $r \in R \Rightarrow \varepsilon \in E$ ), la loi de commande  $K$  est telle que le module de la fonction de transfert  $T_{r \rightarrow \varepsilon}$  est inférieur au module de la fonction de transfert  $1/W_r W_\varepsilon$ . Par la suite, on posera  $W_1(j\omega) = W_r W_\varepsilon$ . Dans le cas d'un suivi de référence en échelon,  $W_1(j\omega) = 1/\varepsilon_{\text{sup}} j\omega$ .

Dans le cas considéré de signaux de référence en échelon, une difficulté se présente car la fonction de transfert  $W_1(p) = 1/\varepsilon_{\text{sup}} p$  n'est pas stable. Dans ce cas, on peut prendre :

$$W_1(p) = \frac{1}{\varepsilon_{\text{sup}}(p + \bar{\varepsilon})} \quad (1-31)$$

avec  $\bar{\varepsilon}$  faible et positif. Dans ce cas-là, on obtient une contrainte sur le module de la fonction de transfert  $T_{r \rightarrow \varepsilon}$ . Dans ce qui précède, cette contrainte sur le module est appelée “gabarit fréquentiel” :

$$\Omega(\omega) = \frac{1}{|W_1(j\omega)|} = \varepsilon_{\text{sup}} \sqrt{\omega^2 + \bar{\varepsilon}^2} \quad (1-32)$$

La forme du module de la fonction de transfert  $T_{r \rightarrow \varepsilon}$  représentée sur la figure 1.11 est relativement typique pour le suivi de référence en échelon. La pulsation  $\omega_c^w$ , telle que  $|W_1(j\omega_c^w)| = 1$ , est décisive car  $\omega_c^w \approx 1/\varepsilon_{\text{sup}}$ . Plus sa valeur est importante, et plus on peut garantir que l'erreur  $|\varepsilon(j\omega)|$  est faible. La relation de complémentarité entre les fonctions de sensibilité permet de déduire une contrainte en basses fréquences pour le transfert de  $T_{r \rightarrow y}$  :  $|T(j\omega)| \geq 1 - \Omega(\omega)$ .

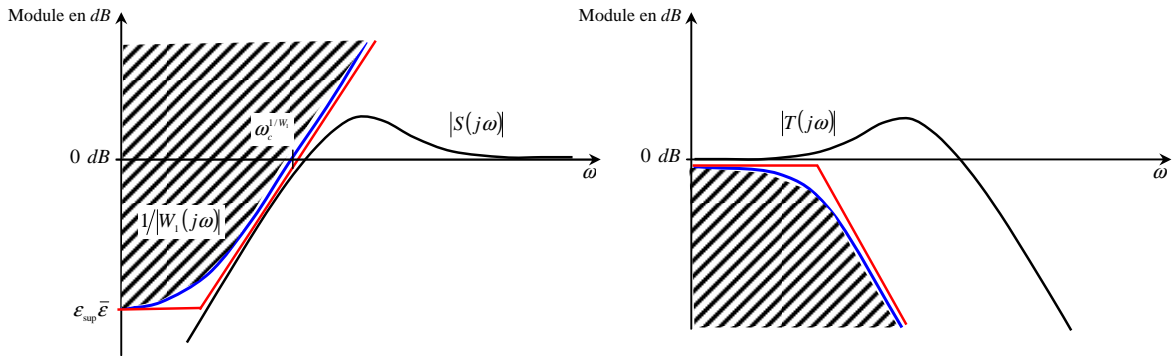


Fig. 1.11: Contraintes sur les modules des fonctions de sensibilité  $S$  et  $T$

**Résumé**

Le tableau suivant résume l'étude du suivi de trajectoire pour les 3 classes de signaux de références.

	$W_1(p)$	Erreur $\varepsilon(t)$ en régime permanent
Echelon d'amplitude $A$	$\frac{1}{\varepsilon_{\text{sup}}(p + \bar{\varepsilon})}$	$ \varepsilon(t)  \leq \bar{\varepsilon} \varepsilon_{\text{sup}} A$
Rampe de pente $a$	$\frac{1}{\varepsilon_{\text{sup}} p(p + \bar{\varepsilon})}$	$ \varepsilon(t)  \leq \bar{\varepsilon} \varepsilon_{\text{sup}} a$
Sinusoïde $(A, \omega_0)$	$\frac{k \omega_0}{p^2 + \bar{\varepsilon} p + \omega_0^2}$	$ \varepsilon(t)  \leq \bar{\varepsilon}/k A  \sin(\omega_0 t + \phi) $

Tab. 1.2: Suivi de référence de différentes classes de signaux

Le régime transitoire de la réponse temporelle  $y(t)$  d'un système à une consigne  $r(t)$  en échelon est également caractérisé par un ou plusieurs indicateurs de rapidité et d'amortissement (cf. figure 1.12) :

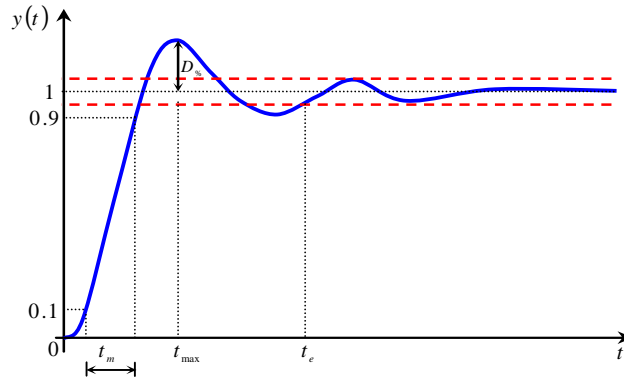
**Caractéristiques temporelles**

**Le temps de montée  $T_m$**  : temps pour que la sortie  $y(t)$  passe de 10% à 90% de la valeur finale ;

**Le temps du premier maximum  $T_{\max}$**  : dans le cas où la réponse à un dépassement non nul, temps pour lequel se produit le premier dépassement ;

**Le temps d'établissement  $T_e$**  : défini ici comme étant le temps à partir duquel la sortie  $y(t)$  reste inférieure à  $\pm\alpha\%$  de la valeur finale (cf. équation (1-21));

**Le dépassement en %  $D_{\%}$**  : exprimé en pourcentage, il représente le rapport entre le dépassement maximale de la sortie  $y(t)$  et sa valeur asymptotique  $y(\infty)$ .



**Fig. 1.12: Caractéristiques temporelles du régime transitoire de la réponse à un échelon**

La rapidité peut être définie par le temps de réponse qui, en fonction du problème considéré, peut être définie par l'un des trois temps précédemment introduits. Dans la suite, par convention et sauf mention contraire, le temps de réponse sera défini par le temps d'établissement.

La précision statique est souvent caractérisée par l'erreur statique relative (différence entre la valeur de l'échelon de référence  $r(t)$  et la valeur finale du signal de sortie  $y(t)$  ramenée sur la valeur finale de l'échelon  $r(t)$ ) (cf. relation (1-22)). Quant à la précision dynamique, elle peut être évaluée en termes de critères temporels sur le signal d'erreur. Les deux principales classes de critères à minimiser sont :

- Les intégrales de moyennes quadratiques :  $J = \int_{t=0}^{\infty} \varepsilon(t)^2 dt$
- Les intégrales de moyennes absolues :  $J = \int_{t=0}^{\infty} |\varepsilon(t)| dt$

Plusieurs variantes de ces critères peuvent être formulées, soit en utilisant des erreurs relatives, soit par l'ajout de fonctions de pondérations dépendantes ou indépendantes du temps.

**1.5.2.2. Rejet/atténuation de signaux de perturbation.**

Après avoir étudié le suivi de signaux de référence  $r(t)$ , on considère le problème du rejet de la perturbation  $b(t)$ . On suppose que le signal de référence est nul (problème de régulation) et on souhaite rejeter l'effet de la perturbation sur la sortie  $y(t)$  ou de façon équivalente sur l'erreur de suivi de trajectoire  $\varepsilon(t)$ .

Pour cela, on utilise la même approche que précédemment. La seule différence est qu'au lieu de considérer  $T_{r \rightarrow \varepsilon}$ , on considère la fonction de transfert  $T_{b \rightarrow \varepsilon} = -SG = -GS_e$  qui relie le signal de

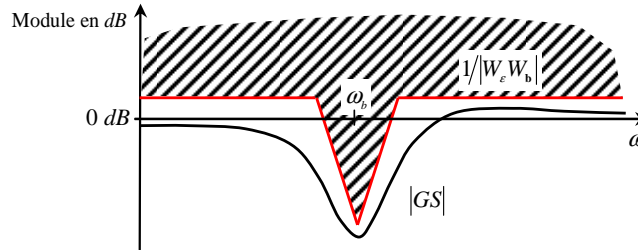
perturbation  $b(t)$  à l'erreur de suivi de trajectoire  $\varepsilon(t)$ . En répétant strictement le même raisonnement qu'avant, on arrive à la conclusion que la perturbation sera atténuée si

$$\|W_\varepsilon T_{b \rightarrow \varepsilon} W_b\|_\infty \leq 1 \quad (1-33)$$

où la fonction de transfert  $W_\varepsilon$  définit l'erreur de suivi de trajectoire et la fonction de transfert  $W_b$  définit l'ensemble  $B$  des signaux de perturbations:

$$B = \{b(j\omega) \text{ tel que } |b(j\omega)| \leq |W_b(j\omega)|\} \quad (1-34)$$

De la même façon que précédemment, on peut faire le raisonnement suivant. Sur les gammes de pulsations où le module de la transformée de Fourier du signal  $b(t)$  est important, pour assurer une bonne atténuation, voir un rejet des perturbations, le module de la fonction de transfert  $T_{b \rightarrow \varepsilon} = -GS$  doit être faible. Par exemple, une forme typique pour le module de la fonction de transfert  $-GS$  pour assurer l'atténuation de l'effet d'une perturbation sinusoïdale de pulsation propre  $\omega_b$  est représentée figure 1.13. Dans ce cas-là, le tableau 1.2 reste valable en remplaçant  $W_1(p)$  par  $W_\varepsilon(p)W_b(p)$ .



**Fig. 1.13: Gabarit fréquentiel du transfert  $T_{b \rightarrow \varepsilon}$  pour une perturbation sinusoïdale de pulsation  $\omega_b$**

### 1.5.2.3. Atténuation des bruits de mesure.

Il s'agit d'étudier l'influence des signaux de bruit  $d(t)$  sur le signal de commande  $u(t)$  et le signal de sortie  $y(t)$ . La densité spectrale de puissance du bruit de mesure  $d$  est importante dans les hautes pulsations. Comme précédemment, si on assimile le bruit à un signal déterministe, on peut définir un ensemble de signaux de bruits  $D$  par l'introduction d'une fonction de transfert  $W_d$  telle que :

$$D = \{d(j\omega) \text{ tel que } |d(j\omega)| \leq |W_d(j\omega)|\} \quad (1-35)$$

où  $W_d$  est un filtre passe-haut. Pour assurer l'atténuation de l'effet des bruits sur la commande et sur la sortie du système, il faut donc que  $T_{d \rightarrow u}$ , et  $T_{d \rightarrow y}$  soient faibles en module pour la gamme des hautes pulsations. La forme générale des fonctions de transfert  $T_{d \rightarrow u}$  et  $T_{d \rightarrow y}$  est donc celle de fonctions de transfert passe-bas. Cependant, les transferts  $T_{d \rightarrow \varepsilon}$  et  $T_{d \rightarrow y}$  sont complémentaires. Il faut donc faire attention à la cohérence des contraintes qui sont imposées sur les différents transferts : comme  $T_{d \rightarrow \varepsilon} + T_{d \rightarrow y} = 1$ , on doit avoir pour les hautes pulsations  $|T_{d \rightarrow \varepsilon}(j\omega)| = |S(j\omega)| \approx 1$ . Un compromis est alors nécessaire (cf. figures 1.11 et 1.14).

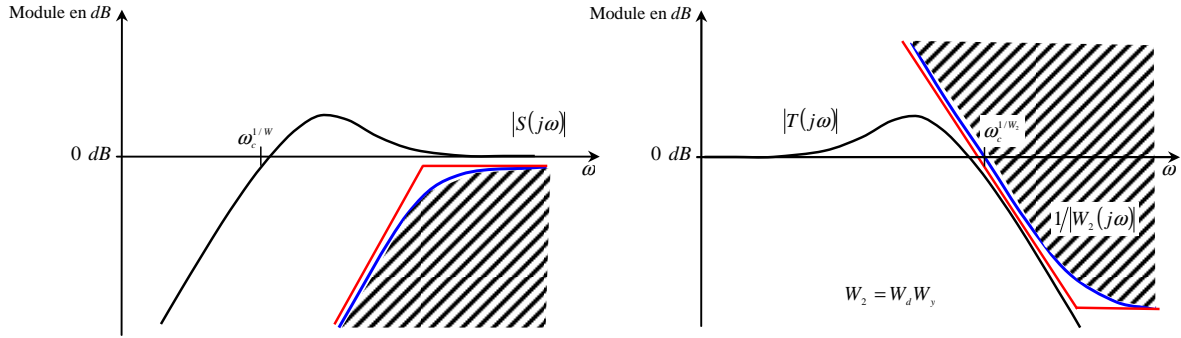


Fig. 1.14: Contraintes sur les modules des fonctions de sensibilité  $S$  et  $T$

#### 1.5.2.4. Commande modérée.

Un autre point important mentionné précédemment est que la commande ne doit pas être trop forte afin de ne pas saturer les actionneurs. Pour définir cela, on peut introduire une pondération  $W_u$  telle que les signaux de commande  $u$  désirables appartiennent à l'ensemble  $U$  avec

$$U = \left\{ u(j\omega) \text{ tel que } |u(j\omega)| \leq \frac{1}{|W_u(j\omega)|} \right\} \quad (1-36)$$

Or,  $u(p) = T_{r \rightarrow u}(p)r(p) + T_{b \rightarrow u}(p)b(p) + T_{d \rightarrow u}(p)d(p)$ . Ainsi, pour éviter, par exemple, une influence trop grande du bruit de mesure  $w(t)$  sur la commande, les gains de la fonction de transfert  $T_{d \rightarrow u} = KS$  doivent être limités dans les hautes pulsations, donc la fonction de transfert  $KS$  doit avoir la forme d'un transfert passe-bas. Cela peut se formaliser par :

$$\|W_u T_{d \rightarrow u} W_d\|_{\infty} \leq 1 \quad (1-37)$$

De plus, par rapport aux signaux de référence, les commandes ne doivent pas être trop fortes. On a donc intérêt à limiter au maximum  $|T_{r \rightarrow u}(j\omega)|$  ce qui peut se traduire par :

$$\|W_u T_{r \rightarrow u} W_r\|_{\infty} \leq 1 \quad (1-38)$$

Notons que dans le cas de la boucle fermée décrite par la figure 1.10, on a :  $T_{d \rightarrow u} = -T_{r \rightarrow u} = KS$ . Les contraintes (1-37) et (1-38) sont vérifiées si en introduisant une pondération  $W_2$  telle que  $|W_2(j\omega)| \geq \max(|W_u(j\omega)W_d(j\omega)|, |W_u(j\omega)W_r(j\omega)|)$ , on a :  $\|W_2 KS\|_{\infty} \leq 1$ . D'après la discussion précédente, l'inverse de la fonction de transfert  $W_2$  est une fonction passe-bas.

#### 1.5.2.5. Dépendance des spécifications et limite de performances

Dans une approche purement fréquentielle, la vérification d'un cahier des charges revient à tester, si pour les pondérations définissant les signaux de référence  $W_r$ , de perturbation  $W_b$  et de bruit  $W_d$  et les pondérations définissant les signaux d'erreur de suivi de trajectoires  $W_\epsilon$  et de commande  $W_u$ , les conditions suivantes sont satisfaites :

- Suivi de trajectoires de référence  $\|W_\epsilon T_{r \rightarrow \epsilon} W_r\|_{\infty} \leq 1$  ;
- Rejet de perturbations  $\|W_\epsilon T_{b \rightarrow \epsilon} W_b\|_{\infty} \leq 1$  ;

- Atténuation des bruits  $\|W_u T_{d \rightarrow u} W_d\|_{\infty} \leq 1$  ;
- Commandes modérées  $\|W_u T_{r \rightarrow u} W_r\|_{\infty} \leq 1$  ;

Dans le cas où l'on recherche une loi de commande  $K$  telle que le système en boucle fermée vérifie les contraintes ci-dessus (c'est-à-dire le cahier des charges), rien ne garantit qu'elle existe. Avant la mise au point de la loi de commande, il est donc important de se demander si les spécifications du cahier des charges sont réalistes pour le système considéré, c'est-à-dire, essayer de s'assurer a priori qu'il existe une loi de commande qui permet de satisfaire le cahier des charges.

Tout d'abord, il existe des relations entre les différentes fonctions de transfert en boucle fermée, ce qui implique que les différentes spécifications du cahier des charges ne sont pas indépendantes les unes des autres. Il est donc nécessaire de faire des compromis entre les différentes spécifications. Ces contraintes sont de plusieurs natures.

### *Contraintes algébriques entre les différentes fonctions de transfert*

Les différentes fonctions de transfert en boucle fermée sont reliées par des relations algébriques. Citons, la relation de complémentarité entre la fonction de sensibilité et la fonction de sensibilité complémentaire :  $T_{r \rightarrow \varepsilon} + T_{r \rightarrow y} = 1$ , la dépendance de certains transferts par exemple :  $T_{b \rightarrow \varepsilon} = G T_{r \rightarrow \varepsilon}$  (la fonction de transfert  $T_{b \rightarrow \varepsilon}$  est le produit de la fonction de transfert  $T_{r \rightarrow \varepsilon}$  par la fonction de transfert  $G$  qui est donnée a priori car c'est le modèle du système à commander et donc on ne peut pas, à travers le choix de  $K$ , modéliser  $T_{r \rightarrow \varepsilon}$  et  $T_{b \rightarrow \varepsilon}$  de façon indépendante),  $T_{r \rightarrow y} = G T_{r \rightarrow u}$  etc..

### *Contraintes sur la forme du module des fonctions de transfert en boucle fermée*

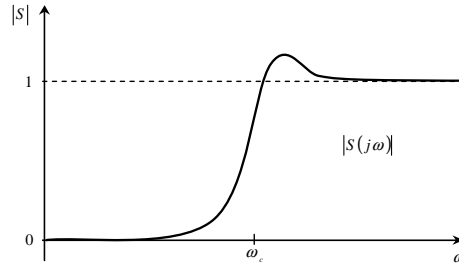
Sur ce sujet, les résultats sont assez nombreux mais malheureusement assez techniques et spécifiques. On va donc juste en présenter quelques uns afin d'en donner un aperçu. Dans ce qui suit, la fonction de transfert de gain de boucle est supposée rationnelle et strictement propre.

**Intégrale de sensibilité de Bode :** Le premier résultat concernant cette question est celui du théorème de Bode appelé aussi la formule de l'aire [Bod45]. Ce théorème stipule que si le transfert en boucle ouverte  $L = GK$  de la figure 1.10 possède au moins deux pôles de plus que de zéros, la fonction de sensibilité doit satisfaire :

$$\int_{-\infty}^{+\infty} \log|S(j\omega)| d\omega = \pi \sum_{i=1}^k \operatorname{Re}(p_i) \quad (1-39)$$

où  $p_i$  sont les pôles instables de la fonction de transfert en boucle ouverte  $L(j\omega)$ . Par la suite, pour un système stable en boucle ouverte, l'intégrale sur toutes les pulsations de la fonction  $\log|S(j\omega)|$  sera nulle. Cela signifie que la fonction  $\log|S(j\omega)|$  prend à la fois des valeurs positives et négatives ou encore que  $|S(j\omega)|$  prend à la fois des valeurs inférieures et supérieures à 1.

Selon ce qui a été énoncé dans le paragraphe 1.1, pour que la commande à contre-réaction soit efficace et utile,  $|S(j\omega)|$  doit être inférieure à 1 pour une bande de basses pulsations. L'intégrale de Bode implique que si cette condition doit être réalisée, ce sera au prix d'une amplification (au lieu d'une atténuation) des perturbations en hautes pulsations, comme illustré dans la figure ci-dessous. Ceci est également valable si le système en boucle ouverte a des pôles à partie réelle dans le demi-plan droit.



**Fig. 1.15: Le dilemme atténuation/amplification des perturbations en basses/hautes pulsations**

La formule de l'aire de Bode ne reflète pas suffisamment le compromis nécessaire entre atténuation et amplification des perturbations, car l'amplification pour de hautes pulsations peut rester relativement faible tant qu'elle s'étend sur une bande de pulsations suffisamment large. La présence dans le demi-plan droit de zéros du système impose des limitations bien plus sévères.

**Égalité de Zames-Francis :** Dans le cas de zéros et de pôles à parties réelles positives dans le système, on peut utiliser l'égalité de Zames-Francis [Zam83, Fra84], fondée sur la formule intégrale de Poisson tirée de la théorie des fonctions complexes. Nous supposons que le système en boucle fermée de la figure 1.10 est stable et que le transfert en boucle ouverte  $L=GK$  possède un zéro  $z = x + iy$  dans le demi-plan-droit ( $x > 0$ ). L'égalité de Zames-Francis est donnée par :

$$\int_{-\infty}^{+\infty} \log(|S(j\omega)|) \frac{x}{x^2 + (y - \omega)^2} d\omega = \pi \log |B_{pôles}^{-1}(z)| \quad (1-40)$$

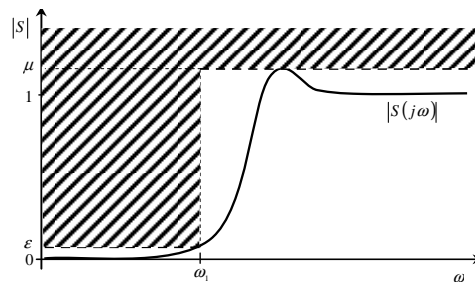
$B_{pôles}$  est le produit de Blaschke formé à partir des pôles  $p_i$  instables du transfert en boucle ouverte :

$$B_{pôles}(z) = \prod_i \frac{p_i - z}{\bar{p}_i + z} \quad (1-41)$$

$\bar{p}_i$  représente le complexe conjugué de  $p_i$ .

Nous voulons que la fonction de sensibilité  $|S(j\omega)|$  soit inférieure à un  $\varepsilon$ , dans la bande de pulsations  $[0, \omega_1]$ , avec  $\omega_1$  une pulsation donnée. Nous analyserons dans ce qui suit la valeur du pic  $\mu$  de  $|S(j\omega)|$  dans la gamme de pulsations complémentaire. La figure 1.16 montre les limites  $\varepsilon$  et  $\mu$  ainsi que le comportement souhaité de la fonction de sensibilité  $|S(j\omega)|$ . Le gabarit fréquentiel est défini par :

$$b(\omega) = \begin{cases} \varepsilon & \text{pour } |\omega| \leq \omega_1 \\ \mu & \text{pour } |\omega| > \omega_1 \end{cases} \quad (1-42)$$



**Fig. 1.16: Limites sur la fonction de sensibilité  $S(j\omega)$**



L'égalité de Zames-Francis permet de montrer [Fre88] que :

$$\mu \geq \left( \frac{1}{\varepsilon} \right)^{\frac{W_z(\omega_i)}{\pi - W_z(\omega_i)}} \left| B_{\text{pôles}}^{-1}(z) \right|^{\frac{\pi}{\pi - W_z(\omega_i)}} \quad (1-43)$$

Par conséquent, pour une valeur de  $\varepsilon < 1$ , la valeur du pic  $\mu$  est supérieure à 1. De plus, plus  $\varepsilon$  est petit, plus la valeur de pic est grande. Ainsi, une sensibilité faible en basses pulsations est associée à une grande valeur de pic en hautes pulsations.

L'égalité de Zames-Francis est valable pour tout zéro du demi-plan droit, et en particulier celui qui possède le module le plus faible. Dès lors, on montre que :

- La limite supérieure de la bande pour laquelle une atténuation efficace des perturbations est possible, est bornée par le module du plus petit des zéros du demi-plan droit. En pratique, la bande passante réalisable est toujours inférieure à ce module.
- Si le procédé a des pôles instables, l'atténuation réalisable de perturbations s'en trouve encore diminuée. Cet effet est particulièrement prononcé lorsque plusieurs paires de pôles-zéros du demi-plan droit sont proches.
- Si le système ne possède pas de zéros instables, la bande passante maximale qui peut être obtenue n'est limitée que par les possibilités du système.

A l'image des résultats pour la fonction de sensibilité  $S(j\omega)$ , des compromis bien définis sont valables pour la fonction de sensibilité complémentaire  $T(j\omega)$ . Dans ce cas, le rôle des zéros instables sera joué par les pôles instables de la boucle ouverte et vice-versa. Ceci est simplement vérifié par :

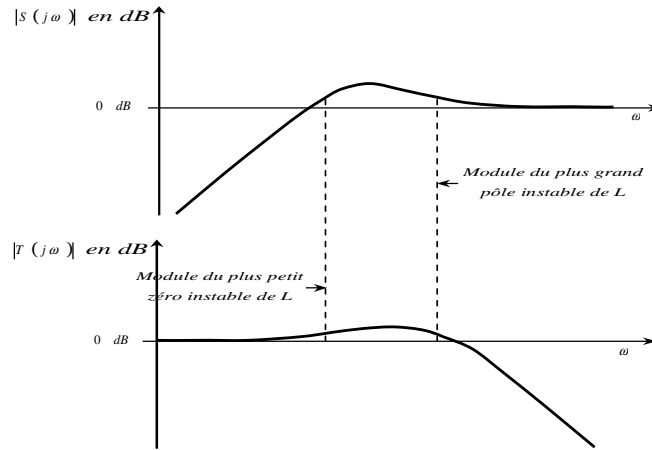
$$T = L(1 + L)^{-1} = (1 + L^{-1})^{-1} \quad (1-44)$$

Suite à un raisonnement dual à celui utilisé pour la fonction de sensibilité  $S(j\omega)$ , si l'on veut éviter des pics excessifs de la fonction de sensibilité complémentaire  $T(j\omega)$  en pulsations basses et intermédiaires. Le module  $|T(j\omega)|$  ne peut être petit que pour des pulsations supérieures au module du pôle instable de la boucle ouverte  $L(j\omega)$  dont le module est le plus grand. Encore une fois, la présence de paires rapprochées dans le demi-plan aggrave la situation.

Les effets qualitatifs des pôles instables du système sur l'allure de la fonction de sensibilité complémentaire  $T(j\omega)$  sont résumés par les remarques suivantes :

- La limite inférieure de la plage pour laquelle la fonction de sensibilité complémentaire peut être faible est bornée par le module du plus grand pôle instable en boucle ouverte. En pratique, la bande passante réalisable est toujours supérieure à ce module.
- Si le système a des zéros instables, la réduction de  $T(j\omega)$  est diminuée. Cet effet est surtout affirmé quand une ou plusieurs paires de pôles-zéros instables sont proches.

La figure 1.17 résume les contraintes créées par les zéros et les pôles instables du système à commander



**Fig. 1.17: Contraintes imposées par les zéros et les pôles instables sur les fonctions de sensibilité**

La fonction de sensibilité  $S(j\omega)$  ne peut être faible que pour des pulsations inférieures au module du plus petit des zéros instables. La fonction de sensibilité complémentaire  $T(j\omega)$  ne peut commencer à décroître vers zéro qu'à des pulsations supérieures au module du plus grand des pôles instables. La zone de transition où  $S(j\omega)$  et  $T(j\omega)$  prennent leur valeur maximale est située dans la gamme de pulsations intermédiaires.

**Liens entre le transfert en boucle ouverte et les transferts de la boucle fermée**

Les liens entre la fonction de transfert en boucle ouverte  $L = GK$  et les fonctions de transfert en boucle fermée ont été mis en avant très tôt en Automatique fréquentielle classique. En effet, la fonction de transfert en boucle ouverte dépend linéairement du correcteur  $K$  alors que les fonctions de transfert en boucle fermée sont des fonctions non-linéaires de  $K$ . Dans la perspective d'ajuster un correcteur  $K$ , il est intéressant d'essayer de traduire les spécifications du cahier des charges comme des contraintes sur la fonction de transfert en boucle ouverte même si elles ne s'expriment pas naturellement comme cela. Nous rappelons que pour le transfert en boucle ouverte, les hautes et basses pulsations sont définies par rapport à la pulsation de coupure  $\omega_c$  définie par  $|G(j\omega_c)K(j\omega_c)| = 1$ .

Dans de nombreux cas, afin de remplir les spécifications en performance déjà détaillées, nous devons satisfaire  $|G(j\omega)K(j\omega)| \gg 1$  pour les basses pulsations ( $\omega \ll \omega_c$ ) et  $|G(j\omega)K(j\omega)| \ll 1$  pour les hautes pulsations ( $\omega \gg \omega_c$ ). Le tableau suivant résume les dépendances entre le transfert en boucle ouverte et les différents transferts en boucle fermée dans le cas monovarié.

Transfert	Basses pulsations	Hautes pulsations
$ K(j\omega)G(j\omega) $	$\gg 1$	$\ll 1$
$ S(j\omega) $	$\approx  G(j\omega)K(j\omega) ^{-1}$	$\approx 1$
$ T(j\omega) $	$\approx 1$	$ G(j\omega)K(j\omega) $
$ K(j\omega)S(j\omega) $	$\approx  G(j\omega) ^{-1}$	$\approx  K(j\omega) $

**Tab. 1.3: Liens entre le transfert en boucle ouverte et les transferts en boucle fermée**

### 1.5.2.6. Performances des systèmes linéaires multivariables

La définition des performances précédemment introduites repose essentiellement sur la notion de gain d'une fonction de transfert à une pulsation  $\omega$ . Dans le cas d'un système à une entrée et une sortie, cette notion correspond au module de la fonction de transfert. Elle se généralise dans le cas de systèmes multivariables via la notion de valeurs singulières.

#### Définition des objectifs de performances

**Poursuite des signaux de référence :** En se basant sur le même principe que dans le cas monovarié, la poursuite idéale d'un signal de référence  $r$  par la sortie  $y$  entraîne que  $T = I - S \approx I$ , soit  $S \approx 0$ . Pour que ceci soit réalisé, il suffit que la valeur singulière maximale de la fonction de sensibilité en sortie  $\bar{\sigma}(S)$  vérifie  $\bar{\sigma}(S) \ll 1$ . Par la suite, nous pouvons écrire :

$$\bar{\sigma}(S) = \frac{1}{\underline{\sigma}(I_p + GK)} \leq \frac{1}{\underline{\sigma}(GK) - 1} \quad (1-45)$$

où  $\underline{\sigma}(GK)$  désigne la plus petite valeur singulière du transfert en boucle ouverte en sortie.

Si, de plus, cette valeur singulière vérifie  $\underline{\sigma}(GK) \gg 1$ , nous en déduisons l'équivalence suivante  $\bar{\sigma}(S) \approx 1/\underline{\sigma}(GK)$ .

Nous déduisons ainsi que pour assurer un bon suivi de trajectoire de référence, il faut rendre  $\bar{\sigma}(S)$  le plus faible possible dans le domaine fréquentiel auquel appartiennent les signaux de consigne, ou également, maximiser  $\underline{\sigma}(GK)$ . Comme la poursuite du signal de référence prend son sens aux basses fréquences, nous retrouvons la notion classique de précision associée au grand gain en boucle ouverte.

**Rejet/atténuation des perturbations :** Une seconde performance qu'un système de commande doit assurer est le bon rejet de la perturbation  $b$ . Autrement dit, on souhaite que  $b$  ait une faible influence sur la sortie  $y$ . Cela sera vérifié si la valeur singulière maximale de la matrice de transfert liant  $b$  à  $y$  est aussi faible que possible aux fréquences où la perturbation  $b$  est importante.

Compte tenu de l'équation (1-20) et du tableau 1.1 cela s'écrit :

$$\bar{\sigma}((I_p + GK)^{-1}G) = \bar{\sigma}(SG) \ll 1 \quad (1-46)$$

Vu que  $G$  décrit un système généralement passe-bas, le rejet des perturbations en basses fréquences est équivalent à :

$$\bar{\sigma}(S) \ll 1 \quad (1-47)$$

Donc, pour le rejet des perturbations en basses fréquences, cette condition rejoint celle de la poursuite du signal de référence par la sortie. On rejoint ici l'idée d'un gain de boucle élevé dans ce domaine fréquentiel pour assurer de bonnes performances.

**Rejet de bruits de mesures :** En se référant toujours aux équations (1-20) et au tableau 1.1, nous remarquons que pour réduire l'ampleur de la propagation du bruit de mesure dans la chaîne, il est nécessaire que  $\bar{\sigma}(T) = \bar{\sigma}(I_p - S)$  soit le plus petit possible dans le domaine fréquentiel du bruit de mesure (souvent les hautes fréquences).

En utilisant l'égalité ci-dessous, le rejet des bruits de mesure peut être formulé sur le gain de boucle :

$$\bar{\sigma}(I_p - S) = 1/\underline{\sigma}(I_p + (GK)^{-1}) \quad (1-48)$$

Ceci se traduit dans le domaine fréquentiel des hautes fréquences par  $\bar{\sigma}(GK) \ll 1$ .

**Limitation de l'énergie de commande :** Les commandes reçues par le système seront réduites si la valeur singulière  $\bar{\sigma}(KS)$  est faible. Toutefois, on peut rappeler que  $\bar{\sigma}(T) > \bar{\sigma}(G)\bar{\sigma}(KS)$ . Ce qui entraîne :

$$\bar{\sigma}(KS) < \bar{\sigma}(T)/\bar{\sigma}(G) \quad (1-49)$$

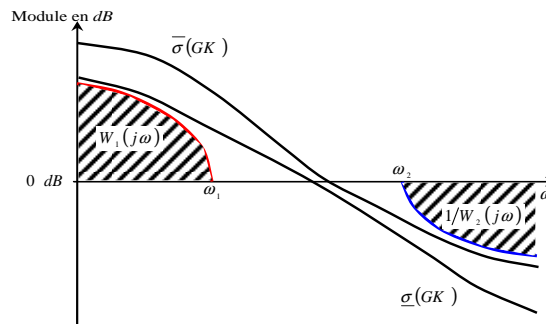
Par la suite, l'énergie de commande sera d'autant plus faible que  $\bar{\sigma}(T)$  sera faible sauf dans les régions fréquentielles où  $\bar{\sigma}(G)$  est lui-même élevé.

**Concept du Loop-Shaping**

Nous pouvons maintenant dresser le bilan des conditions obtenues sur la matrice de transfert  $GK$  :

- On obtient un bon rejet des perturbations sur la sortie et un bon suivi du signal de référence en rendant  $\underline{\sigma}(GK)$  le plus grand possible. Or, par nature, tout processus est globalement passe-bas et  $\|GK\|$  est donc faible en hautes fréquences. L'obtention de ces performances nominales qui se traduit par une augmentation de  $\underline{\sigma}(GK)$  sera donc recherchée aux basses fréquences. Cet objectif se formule par une inégalité telle que :  $\underline{\sigma}(G(j\omega)K(j\omega)) > |W_1(j\omega)| > 1$  pour  $\omega < \omega_1$ .
- De plus, il est important de réduire le plus possible le gain de la boucle aux hautes fréquences afin d'assurer d'une part la réjection des bruits de mesure, et d'autre part la stabilité du système aux incertitudes fréquentielles (retard, dynamiques hautes fréquences négligées). Cet objectif se traduit par l'inégalité suivante :  $\underline{\sigma}(G(j\omega)K(j\omega)) < \frac{1}{|W_2(j\omega)|}$  avec  $|W_1(j\omega)| > 1$  pour  $\omega < \omega_2$ .

L'ensemble de ces conditions permet donc de modéliser le transfert en boucle ouverte afin que les valeurs singulières maximale et minimale en boucle fermée obéissent aux contraintes de performance. L'allure des gains de boucle est représentée par la figure ci-dessus.



**Fig. 1.18: Contraintes de Loop-Shaping**

Les liens entre le transfert en boucle ouverte et le transfert en boucle fermée résultants sont décrits dans la figure 1.19.

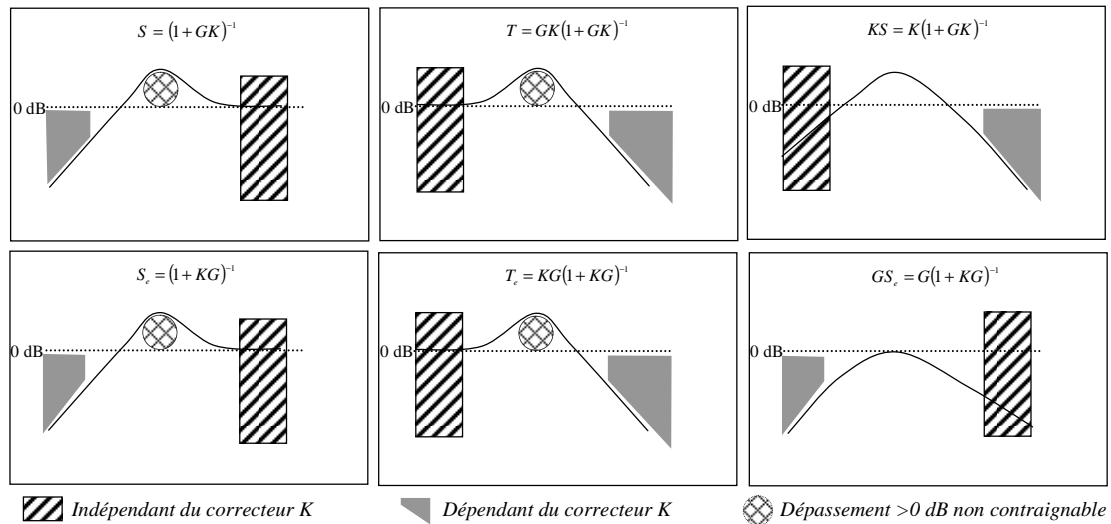


Fig. 1.19: Comportement fréquentiel des transferts en boucle fermée

### 1.5.3. Les spécifications en robustesse

Elles décrivent les marges de sécurité qu'un système bouclé doit vérifier pour que les performances et la stabilité soient conservées si le système à commander ou une de ses parties a été modifiée ou perturbée. Les perturbations, en question, peuvent avoir plusieurs origines, par exemple :

- le système à commander a été mal modélisé ou identifié ;
- un des opérateurs de la boucle fermée (système ou correcteur) change physiquement à cause de la tolérance de ses composants ou d'un coefficient de température par exemple ;
- les non linéarités négligées lors de la synthèse de la commande et qui peuvent être significatives dans le système bouclé réel ;
- l'évolution du point de fonctionnement d'un système non linéaire ce qui implique une erreur significative entre le système et son linéarisé tangent ;
- les défauts qui peuvent être observés pour des composants comme les capteurs et les actionneurs.

Pratiquement, le correcteur  $K$  est souvent synthétisé à partir d'un modèle  $G_{\text{mod}}$  du système physique réel  $G_{\text{réel}}$ . L'étude de la robustesse de la loi de commande  $K$  consiste à obtenir le maximum de garanties en stabilité d'abord, en essayant de la garantir pour toute une classe de systèmes centrée autour du modèle nominal  $G_{\text{mod}}$ , et en performances, afin de répondre aux exigences désirées. Ce modèle de référence qui peut provenir des équations de la physique ou d'un processus d'identification ne peut être qu'une approximation de la réalité. Ses carences peuvent être multiples : incertitudes paramétriques, dynamiques et non linéarités négligées, paramètres mal identifiés, changement du point de fonctionnement, etc. Il est donc insuffisant d'optimiser l'asservissement par rapport au modèle nominal, mais il faut aussi se prémunir contre l'incertitude de modélisation et les aléas externes. Bien que ces facteurs soient par essence mal connus, on dispose généralement d'informations sur leur amplitude maximale, plage de variation, nature statistique. C'est à partir de cette connaissance sommaire qu'on va tenter de robustifier la commande. On distingue deux grandes catégories d'incertitudes :

**Les incertitudes non structurées :** Elles rassemblent les dynamiques négligées dans le modèle. On ne dispose, en général, que d'une borne supérieure sur l'amplitude de ces dynamiques. Par conséquent, on doit se prémunir contre le pire des cas dans la limite de cette borne.

**Les incertitudes structurées :** Elles sont liées aux variations ou erreurs d'estimation sur certains paramètres physiques du système, ou à des incertitudes de nature dynamique, mais entrant dans la boucle en différents points. Les incertitudes paramétriques interviennent principalement lorsque le modèle est obtenu à partir des équations de la physique. La manière dont les paramètres influent sur le comportement du système détermine la structure de l'incertitude.

Ces deux catégories d'incertitudes peuvent englober des phénomènes très divers : linéaires ou non-linéaires, stationnaires ou à temps-variant, frottements, hystérésis, etc.

En fonction des données sur les imperfections du modèle, les incertitudes peuvent être quantifiées sous plusieurs formes (cf. figure 1.20) : incertitude additive, incertitude multiplicative à l'entrée, incertitude multiplicative à la sortie, incertitude additive inverse, incertitude multiplicative inverse à l'entrée ou incertitude multiplicative inverse en sortie. Pour toutes ces configurations, l'incertitude  $\Delta$  doit être EBSB stable.

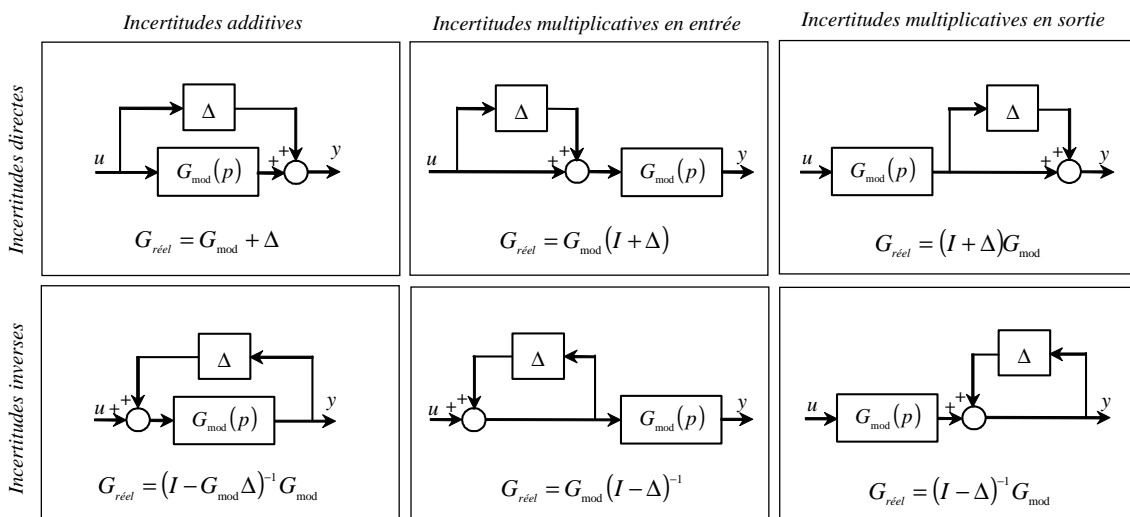


Fig. 1.20: Les classes d'incertitudes non structurées

**L'effet de robustesse d'une boucle de contre-réaction :** Comme nous l'avons déjà énoncé au paragraphe 1.2, l'une des propriétés fondamentales de la structure de commande en boucle fermée est d'améliorer, sous certaines conditions, la robustesse en performance vis-à-vis des erreurs de modélisation du système à commander. Afin d'établir cette propriété, nous comparons la boucle fermée avec sa boucle ouverte équivalente.

En choisissant  $K_{eq}$  tel que :  $K_{eq} = K(1 + G_{\text{mod}}K)^{-1}$  et pour un système à commander  $G_{\text{mod}}$ , connu parfaitement et soumis à aucune perturbation extérieure, les deux configurations sont équivalentes au sens E/S :  $T_{r \rightarrow y_{BF}} = T_{r \rightarrow y_{BO}} = G_{\text{mod}}K(1 + G_{\text{mod}}K)^{-1}$ . En revanche, nous constatons rapidement que la structure en boucle ouverte équivalente, contrairement à la structure en boucle fermée, ne permet pas la stabilisation de systèmes instables et n'apporte pas d'amélioration dans la réponse du système aux conditions initiales.

Afin de mettre en évidence l'apport de la structure de contre-réaction à la robustesse aux incertitudes du système  $G_{\text{mod}}$ , nous comparons l'erreur absolue due à la présence d'une perturbation additive directe  $\Delta$  dans le processus (cf. figure 1.21)

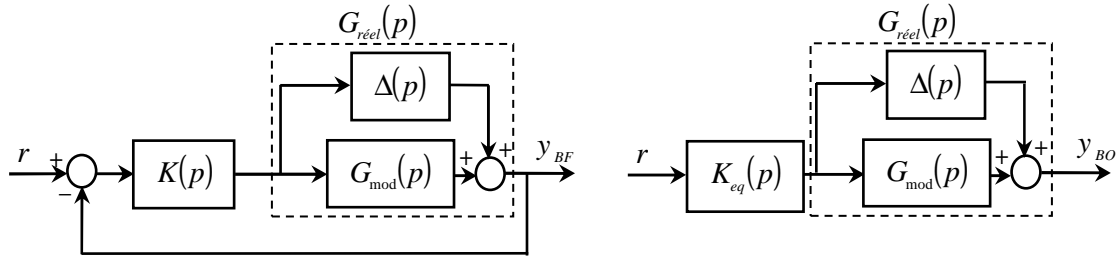


Fig. 1.21: Système incertain en boucle fermée et sa boucle ouverte équivalente

Nous écrivons alors :

$$\Delta y_{BO}(p) = \Delta(p) K_{eq}(p) r(p) = \Delta(p) \frac{K(p)}{1 + G_{\text{mod}}(p) K(p)} r(p) \quad (1-50)$$

$$\Delta y_{BF}(p) = \Delta(p) \frac{K(p)}{(1 + G_{\text{mod}}(p) K(p)) \cdot (1 + G_{\text{réel}}(p) K(p))} r(p) \quad (1-51)$$

Soit finalement,

$$\Delta y_{BF}(p) \approx S(p) \Delta y_{BO}(p) \quad (1-52)$$

Cette dernière équation montre qu'une réduction de l'amplitude de la fonction de sensibilité permet une réduction des erreurs dues aux incertitudes de modélisation et améliore donc la robustesse aux performances. Ainsi, une meilleure insensibilité de la boucle fermée sur la boucle ouverte équivalente, et ce sur l'ensemble du domaine fréquentiel, exige un correcteur  $K(p)$  satisfaisant :

$$|1 + G_{\text{mod}}(j\omega) K(j\omega)|^{-1} \leq 1 \quad \forall \omega \quad (1-53)$$

La propriété (1-52) constitue en soi une justification fondamentale de la commande en boucle fermée. Toutefois, nous avons montré, lors de l'étude des spécifications en performance, que cette contrainte de la fonction de sensibilité ne peut être effective que sur une plage de fréquences (basses fréquences). Dans le cas où la robustesse est perdue, un ajustement de la loi de commande est nécessaire. Ceci se fait à travers la définition des conditions (contraintes) de robustesse que le système bouclé doit vérifier. Ces conditions sont souvent formulées dans le domaine fréquentiel et peuvent prendre plusieurs formes où la tendance est, essentiellement, de limiter les dégradations et les variations du comportement du système en boucle fermée, dus à certain ensemble de perturbations. On peut grouper les spécifications en robustesse en trois grandes catégories :

### 1.5.3.1. La robustesse en stabilité

Dans le cas linéaire, l'utilisation du critère de Nyquist permet de ramener l'étude de la stabilité d'un système en boucle fermée à l'étude de certaines caractéristiques de la réponse fréquentielle de sa fonction de transfert en boucle ouverte. Des notions de robustesse peuvent alors être déterminées sur

la boucle ouverte du système. Bien que limité au cas linéaire monovarié, la puissance et la facilité de calcul de ces critères en font un outil de choix pour spécifier, même imparfaitement, des problèmes complexes.

**Les marges de stabilité classiques**

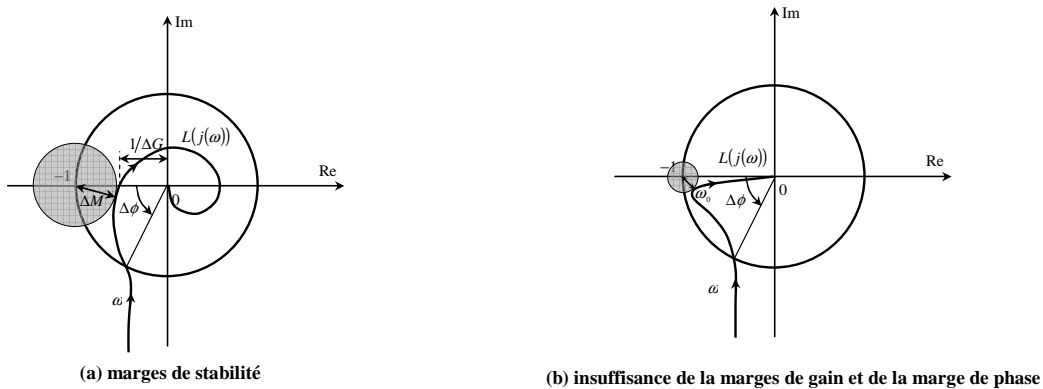
La figure 1.22(a) présente les marges de stabilité classiques. La marge de gain  $\Delta G$  est le scalaire positif par lequel le lieu de Nyquist doit être multiplié afin de passer par le point  $-1$  :

$$\Delta G = \frac{1}{|L(j\omega_{-\pi})|} \tag{1-54}$$

où  $\omega_{-\pi}$  est la pulsation pour laquelle le lieu de Nyquist coupe l'axe des réels négatifs au point le plus éloigné de l'origine (cf. figure 1.21(a)).

La marge de phase  $\Delta\phi$  représente la plus petite phase supplémentaire qui doit être ajoutée afin de déstabiliser le système bouclé :

$$\Delta\phi = \inf_i \{ \arg(L(j\omega_c^i)) + \pi \}, \text{ avec } |L(j\omega_c^i)| = 1 \tag{1-55}$$



**Fig. 1.22: Insuffisance des notions de marge de gain et de marge de phase**

Une autre notion, qui découle directement de la définition de la marge de phase, est la *marge de retard*  $\Delta R$ . Elle est définie comme étant le rapport entre la marge de phase et la pulsation de coupure  $\omega_c$  pour laquelle elle a été calculée :  $\Delta R = \Delta\phi / \omega_c$ . Bien qu'il y ait une relation assez étroite entre la marge de phase et la marge de retard, cette dernière reste très utilisée en industrie, étant donnée son interprétation aisée. En effet, La marge de retard  $\Delta R$  correspond à la plus petite valeur de la constante de retard  $\tau$  telle que le système rebouclé  $e^{-j\omega\tau} L(j\omega)$  soit instable.

Les spécifications en robustesse sous forme de marge de phase et de gain restent qualitatives et deviennent caduques pour les systèmes multivariés où ces notions sont difficiles à généraliser. A titre d'exemple, les notions de marge de phase et de gain présentent des limites même pour le cas monovarié. Le lieu de Nyquist de la figure 1.22(b) montre, malgré une excellente marge de phase supérieure à  $60^\circ$  et une marge de gain infinie, une faible robustesse aux incertitudes paramétriques ou structurelles autour de la pulsation  $\omega_0$ <sup>3</sup>. Dans ce cas, une petite incertitude sur le système peut se traduire, dans le plan de Nyquist, par une variation de gain et/ou un retard de phase qui déstabilise la

<sup>3</sup> La pulsation correspondant à la plus petite distance entre le lieu de Nyquist et le point (-1,0)



boucle de commande. D'autre part, si le tracé de  $L_{\text{mod}}(j\omega)$  est suffisamment éloigné du point  $(-1,0)$ , il y a peu de chance que le tracé de  $L_{\text{réel}}(j\omega)$  le recouvre ou l'encercle. Ainsi, il sera judicieux de considérer la distance du lieu de Nyquist  $L_{\text{mod}}(j\omega)$  au point  $(-1,0)$  afin de mesurer la robustesse en stabilité. Cette marge dite **marge de module** apparaît comme un critère pertinent pour caractériser la robustesse à la stabilité. Elle est définie par l'une des relations suivantes :

$$\Delta M = \inf_{\omega \geq 0} \{ |1 + L_{\text{mod}}(j\omega)| \} = \sup_{\omega \geq 0} \{ |1 + L_{\text{mod}}(j\omega)|^{-1} \} = \|S(j\omega)\|_{\infty}^{-1} \quad (1-56)$$

La marge de gain correspond à une incertitude sur le gain du système, indépendamment de la pulsation  $\omega$ , donc à un type très spécifique d'incertitude. La marge de phase s'interprète difficilement, si ce n'est par la définition de la marge de retard (soit un autre type d'incertitude très spécifique). À l'opposé, la marge de module mesure correctement la distance entre le point critique et le tracé de la fonction de transfert en boucle ouverte, mais elle n'a pas a priori d'interprétation en termes d'incertitude d'erreur entre le modèle et le système réel. Une interprétation sera néanmoins présentée dans la suite de ce chapitre.

Pour les applications de type optimisation il faut noter que les différentes définitions des marges de stabilité sont de type implicite. En effet, ils dépendent du calcul des pulsations correspondantes ( $\omega_{-\pi}$ ,  $\omega_c$  et  $\omega_0$ ) qui sont les résultats de problèmes implicites (recherche de racines de polynômes).

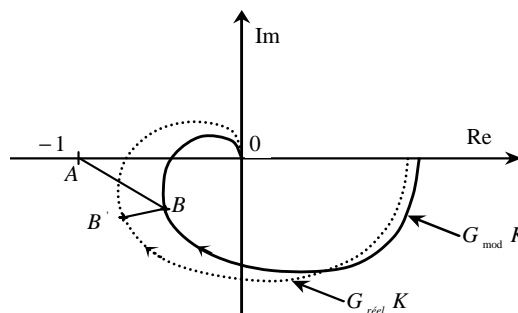
**Formulation de la robustesse en stabilité à partir de la fonction de sensibilité complémentaire  $T$**

Les marges classiques de stabilité ne permettent pas de rendre compte directement et clairement des deux types d'incertitudes : structurelles, telles que les dynamiques hautes fréquences négligées et paramétriques telles que les erreurs d'estimation sur les paramètres du système. Soient les tracés de Nyquist représentés sur la figure 1.23, avec d'une part le lieu de Nyquist du système nominal  $G_{\text{mod}}K$  et d'autre part le lieu de Nyquist du système perturbé  $G_{\text{réel}}K$  par des incertitudes de nature soit fréquentielle soit paramétrique. Une condition suffisante pour que le système perturbé reste stable est que l'on ait  $|BB'| < |AB|$  quel que soit la pulsation  $\omega$ . Cette contrainte se traduit par les inégalités suivantes :

$$|G_{\text{réel}}(j\omega)K(j\omega) - G_{\text{mod}}(j\omega)K(j\omega)| < |1 + G_{\text{mod}}(j\omega)K(j\omega)|$$

soit :

$$\frac{|G_{\text{réel}}(j\omega)K(j\omega) - G_{\text{mod}}(j\omega)K(j\omega)|}{|G_{\text{mod}}(j\omega)K(j\omega)|} |T(j\omega)| < 1 \quad (1-57)$$

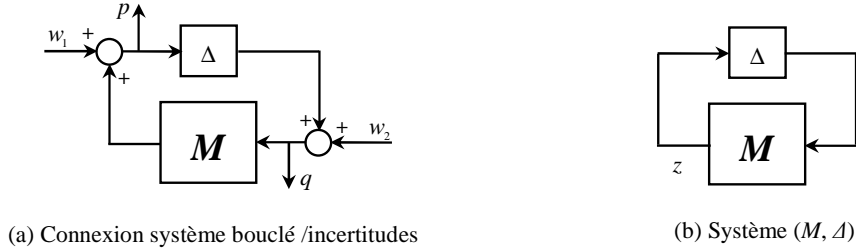


**Fig. 1.23: Lieu de Nyquist du système nominal et perturbé**

Cette relation montre que plus l'erreur relative du modèle sera élevée, plus le module de la fonction de sensibilité complémentaire  $T(j\omega)$  devra être faible. Cette robustesse en stabilité exige donc une fonction de sensibilité complémentaire  $T(j\omega)$  de module très faible aux hautes fréquences ce qui implique un gain de boucle  $|G_{\text{mod}}(j\omega)K(j\omega)|$  petit. Nous constatons que cette exigence est en parfaite concordance avec les exigences en performances étudiées au paragraphe 1.5.2.

**Analyse de la robustesse en stabilité par le théorème du petit gain**

Nous considérons le système incertain représenté par le schéma bloc 1.23(a) :



**Fig. 1.24: Représentation de la connexion modèle/incertitudes**

L'opérateur  $M$  comporte toutes les parties nominales (sans incertitudes) du système bouclé. La stabilité interne<sup>4</sup> de ce système est définie selon le théorème suivant :

**Théorème 1.8 (Stabilité interne d'un système incertain)** *Le système bouclé représenté par la figure 1.24(a) est stable de façon interne si la matrice de fonctions de transfert définie par :*

$$\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} [I - \Delta(p)M(p)]^{-1}\Delta(p) & [I - \Delta(p)M(p)]^{-1} \\ [I - M(p)\Delta(p)]^{-1} & [I - M(p)\Delta(p)]^{-1}M(p) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{1-58}$$

*est stable. De plus, on dit qu'une famille de systèmes est stable si chacun de ses membres est stable.*

Un des résultats incontournables pour l'étude de la robustesse en stabilité des systèmes est le théorème du petit gain. Déjà énoncé auparavant, ce théorème est généralisé maintenant pour les systèmes incertains en se basant sur le critère de Nyquist multivariable (cf. théorèmes 1.4 et 1.5) :

**Théorème 1.9 (Théorème du petit gain généralisé)** *La famille des systèmes  $(M, \Delta)$ , représentée par la figure 1.24(b), est stable pour toutes les matrices de fonctions de transfert  $\Delta$  telles que  $\|\Delta\|_{\infty} \leq \beta$  (respectivement  $\|\Delta\|_{\infty} < \beta$ ) si et seulement si  $\|M\|_{\infty} < 1/\beta$  (respectivement  $\|M\|_{\infty} \leq 1/\beta$ ).*

La plus grande valeur du scalaire  $\|M\|_{\infty}^{-1}$  peut être considérée comme une marge de stabilité : il s'agit d'un indicateur qui permet d'évaluer la quantité maximale d'incertitude sur le système pour laquelle on peut garantir que la loi de commande va stabiliser le système en boucle fermée.

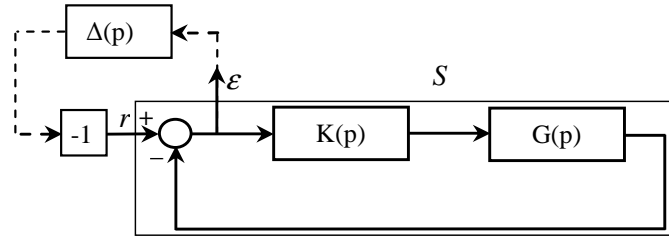
Grâce au théorème du petit gain, il est possible de donner une interprétation de la marge de module définie au début de ce paragraphe comme étant l'inverse de la norme  $H_{\infty}$  de la fonction de sensibilité  $S$  (cf. équation (1-56)). Or d'après le théorème, si  $\|S\|_{\infty} \leq 1/\beta$  alors le système  $(S, -\Delta)$ , représenté par la

<sup>4</sup> Voir la définition donnée dans le paragraphe 1.5.1.3

figure 1.25, est stable pour toute incertitude  $\Delta$  telle que  $\|\Delta\|_\infty < \beta$ . En choisissant  $\beta = \Delta M$  où  $\Delta M$  représente la marge de module, on a donc ces deux inégalités qui sont vérifiées.

D'après le théorème du petit gain, si  $\|\Delta\|_\infty < \Delta M$  alors le système bouclé représenté ci-dessus est stable. Cela correspond à la famille de systèmes de la forme :

$$G(p) = \frac{G_{\text{mod}}(p)}{1 + \Delta(p)} \Leftrightarrow \Delta(p) = \frac{G(p)^{-1} - G_{\text{mod}}(p)^{-1}}{G_{\text{mod}}(p)^{-1}}, \quad (1-59)$$



**Fig. 1.25: Interprétation de la marge de module**

ce qui correspond à une incertitude multiplicative inverse. La marge de module mesure donc la quantité maximale d'erreur relative sur la fonction de transfert inverse du système pour laquelle le système bouclé représenté par la figure 1.25 reste stable. L'intérêt de ce type d'incertitudes est d'obtenir des familles de fonctions de transfert contenant des fonctions de transfert instables même si le système  $G_{\text{mod}}$  est stable.

L'interprétation de la marge de module permet de la définir proprement dans le cas des systèmes multivariables, via les fonctions de sensibilité  $S$  et  $S_e$ . Elles permettent d'étudier la stabilité de la boucle fermée en présence d'une incertitude multiplicative inverse en sortie :  $G_{\text{réel}} = (I + \Delta)^{-1} G_{\text{mod}}$  et en présence d'une incertitude multiplicative inverse en entrée :  $G_{\text{réel}} = G_{\text{mod}} (I + \Delta)^{-1}$ . La première peut modéliser des incertitudes sur les capteurs, la seconde sur les actionneurs. On peut donc définir une marge de module en entrée du système comme l'inverse de  $\|S_e\|_\infty$  et une marge de module en sortie du système comme l'inverse de  $\|S\|_\infty$ . Dans le cas des systèmes monovariables, elles sont égales : assurer la robustesse par rapport à des incertitudes en entrée du système revient à assurer la robustesse par rapport à des incertitudes en sortie du système. Dans le cas des systèmes multivariables, ce n'est pas toujours vrai. En conclusion, pour l'étude de la robustesse des systèmes multivariables, il est impératif d'étudier à la fois les marges en entrée et en sortie du système.

Type d'incertitude	$M(p)$	Borne sur l'incertitude	Stabilité robuste ssi
$G_{\text{mod}} + \Delta$	$KS = S_e K$	$\ \Delta(j\omega)\ _\infty \leq \ W_a(j\omega)\ _\infty$	$\ W_a(j\omega)K(j\omega)S(j\omega)\ _\infty < 1$
$G_{\text{mod}}(I + \Delta)$	$T_e$	$\ \Delta(j\omega)\ _\infty \leq \ W_{me}(j\omega)\ _\infty$	$\ W_{me}(j\omega)T_e(j\omega)\ _\infty < 1$
$(I + \Delta)G_{\text{mod}}$	$T$	$\ \Delta(j\omega)\ _\infty \leq \ W_{ms}(j\omega)\ _\infty$	$\ W_{ms}(j\omega)T(j\omega)\ _\infty < 1$
$(I - G_{\text{mod}}\Delta)^{-1}G_{\text{mod}}$	$GS_e = SG$	$\ \Delta(j\omega)\ _\infty \leq \ W_{ai}(j\omega)\ _\infty$	$\ W_{ai}(j\omega)G(j\omega)S_e(j\omega)\ _\infty < 1$
$G_{\text{mod}}(I - \Delta)^{-1}$	$S_e$	$\ \Delta(j\omega)\ _\infty \leq \ W_{mei}(j\omega)\ _\infty$	$\ W_{mei}(j\omega)S_e(j\omega)\ _\infty < 1$
$(I - \Delta)^{-1}G_{\text{mod}}$	$S$	$\ \Delta(j\omega)\ _\infty \leq \ W_{msi}(j\omega)\ _\infty$	$\ W_{msi}(j\omega)S(j\omega)\ _\infty < 1$

**Tab. 1.4: Conditions de stabilité robuste**

Dans le tableau 1.4, pour les différentes classes d'incertitudes introduites dans la figure 1.20, il est indiqué la matrice de fonctions de transfert correspondante pour l'application du théorème du petit gain ainsi que la condition de stabilité robuste.

**Analyse de la robustesse en stabilité par le critère de passivité :**

Comme nous l'avons déjà mentionné auparavant, l'intérêt de la passivité découle du résultat de bouclage des systèmes passifs. Ce résultat peut être énoncé par théorème suivant :

**Théorème 1.10 (Bouclage d'un système passif)** *Considérons le système bouclé de la figure 1.24(b). Si  $M(p)$  est un système linéaire stable passif, alors la boucle fermée est stable pour tout opérateur passif  $\Delta$  de  $\ell^2$ , c'est-à-dire pour tout opérateur  $\Delta(\cdot)$  satisfaisant :*

$$\forall T > 0, \int_0^T z^T(t)\Delta(z(t))dt \geq 0 \tag{1-60}$$

Une conséquence immédiate de ce théorème est que la boucle de suivi de la figure 1.10 est stable dès que la boucle ouverte  $L(p) = G(p)K(p)$  est passive (il suffit de choisir  $\Delta = 1$ ). On peut donc rechercher la passivité comme un moyen d'imposer la stabilité. Cette démarche est particulièrement pertinente en présence des modes souple d'amortissement incertain [Ala99]. La boucle ouverte reste passive et la stabilité est maintenue quelles que soient les variations d'amortissement. On notera qu'il s'agit ici d'incertitudes paramétriques pour laquelle le théorème du petit gain est typiquement très conservatif. Avec l'approche par passivité, on évite ce conservatisme et on obtient souvent une bonne robustesse sans dégradation importante des performances.

D'après la condition (iii) du Théorème 1.7, la passivité de la boucle ouverte peut s'exprimer comme une contrainte sur la norme  $H_\infty$  de  $S - T = (I + GK)^{-1}(I - GK)$ . En pratique, on n'exige souvent la passivité que dans des bandes limitées de fréquence (par exemple, au voisinage des modes souples), et on utilisera donc une fonction de pondération pour sélectionner les zones de fréquences désirées.

**1.5.3.2. La robustesse en performance :**

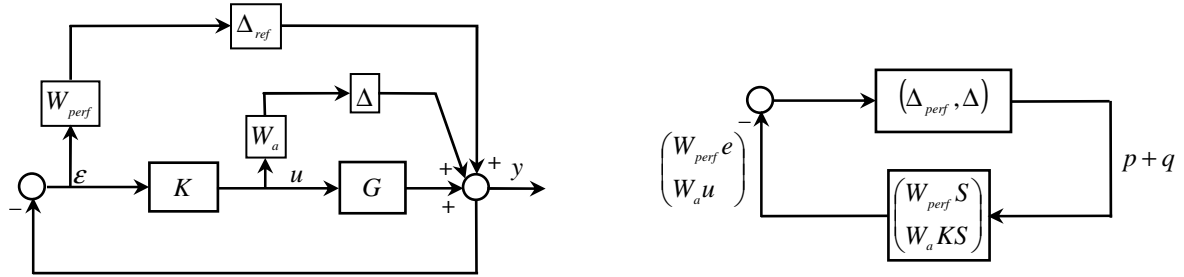
Il s'agit de forcer le système bouclé à conserver un certain degré de performance vis-à-vis aux différentes incertitudes susmentionnées. On a vu dans les paragraphes 1.5.2.1 que les performances de suivi étaient liées au transfert consigne/sortie  $T_{r \rightarrow y}(p) = T(p)$ . Le suivi est bon dans la bande de fréquence où  $T(j\omega) \approx I$ , ou d'une façon équivalente là où le gain minimal en boucle ouverte  $\underline{\sigma}(GK(j\omega))$  est élevé de sorte que  $S(j\omega) \approx 0$ . On peut donc spécifier les performances par un gabarit sur  $S(j\omega)$  de la forme :

$$\forall \omega, W_{perf}(\omega) \overline{\sigma}(S(j\omega)) < 1 \tag{1-61}$$

où  $W_{perf}(\omega)$  est une fonction de pondération rationnelle stable dont le module est grand aux fréquences où des performance de suivi élevées sont requises. Comme pour la stabilité robuste, la condition (1-61) est exprimée pour le modèle nominal  $G(p)$  du système réel. Pour que les performances soient robustes, il faut en fait garantir (1-61) pour toute la famille de systèmes définie par le niveau d'incertitude sur ce modèle. On va donc chercher des conditions sur les transferts en boucle fermée analogues à celles de la table 1.4 et qui expriment la robustesse des performances.

Considérons à nouveau le cas d'une incertitude additive non structurée sur le modèle  $G(p)$  du système réel. Tout d'abord, il est utile d'observer que la contrainte de performance (1-61) peut toujours s'exprimer comme une contrainte de stabilité robuste. En effet, cette condition est équivalente à la stabilité robuste de la boucle en figure 1.24(b) avec  $M = W_{perf}S$ . On appellera  $\Delta_{perf}(p)$  le bloc d'incertitude fictif associé à cette reformulation.

Fort de cette remarque, le problème de robustesse des performances est équivalent à un problème de stabilité robuste pour la boucle suivante :



**Fig. 1.26: Problème de stabilité robuste équivalent**

Les pondérations  $W_a$  et  $W_{perf}$  reflètent la dépendance fréquentielle du niveau d'incertitude. Les incertitudes  $\Delta(p)$  et  $\Delta_{perf}(p)$  sont respectivement contraintes par  $\|\Delta(p)\|_\infty < 1$  et  $\|\Delta_{perf}(p)\|_\infty < 1$ .

Il est possible de réécrire le schéma bloc par un schéma équivalent avec un vecteur d'incertitudes  $\tilde{\Delta} := (\Delta_{perf}, \Delta)$  qui vérifie  $\|\tilde{\Delta}\|_\infty < \sqrt{2}$ .

En application du théorème du petit gain, une condition suffisante pour la robustesse des performances est donc :

$$\left\| \begin{array}{c} W_{perf} S \\ W_a K S \end{array} \right\|_\infty < \frac{1}{\sqrt{2}} \quad (1-62)$$

Des conditions analogues peuvent être obtenues pour les autres types d'incertitudes non structurées. Ces résultats ont les implications suivantes :

- Dans le cas où l'incertitude est faible, on peut utiliser des grands gains pour obtenir des performances robustes élevées.
- Dans le cas où l'incertitude est importante, par contre, les grands gains deviennent inopérants. Ils donnent toujours des performances élevées pour le modèle nominal, mais les performances se dégradent rapidement lorsqu'on s'écarte de ce modèle.

En résumé, à une fréquence donnée, le niveau d'incertitude dynamique limite les gains utilisables et donc les performances robustes atteignables à cette fréquence. Il y a par conséquent conflit entre exigences de performance et de robustesse.

### 1.5.4. Les spécifications de la loi de commande

Ces spécifications décrivent les propriétés de la loi de commande elle-même ; elles peuvent porter sur plusieurs aspects. A titre d'exemple :

**Les spécifications liées aux limites sur le signal de commande :** elles sont généralement dues au choix des actionneurs qui définit l'ensemble de signaux de commande admissibles. Nous citons par exemple, les saturations et les variations limitées.

**Les spécifications liées à la structure de la commande :** elles consistent à imposer une structure donnée à la loi de commande, soit par souci de simplicité soit en raison d'une contrainte d'implémentation qui peut être calculatoire ou physique. Dans ce sens, on peut citer les contraintes d'ordre qui sont généralement imposées dans le domaine industriel et plus particulièrement dans le domaine aérospatial et qui traduisent une exigence d'embarquabilité de la loi de commande. Lors d'un échec de la synthèse d'une loi de commande à structure fixe, toute la configuration de la boucle fermée peut être remise en question.

**La stabilité des contrôleurs en boucle ouverte :** elle représente aussi une exigence majeure, car elle traduit souvent indirectement une robustesse vis-à-vis des éventuelles variations paramétriques qui peuvent déstabiliser le système bouclé. Pour d'autres cas, une exigence supplémentaire peut être demandée, elle consiste à assurer aussi une minimalité de phase du contrôleur afin d'éviter des dégradations en performance. C'est le cas pour certains systèmes linéaires à structure flexible (avec des modes souples) où une simplification pôle-zéro peut avoir des conséquences fâcheuses, tant sur les performances que sur la stabilité du système.

D'autres spécifications peuvent être imposées lors de la phase d'implémentation ; elles sont en grande partie liées à la technologie du système de commande utilisé. Elles sont prises en compte a priori lors de la synthèse ou bien a posteriori par un ajustement de la loi de commande.

## 1.6. Classification des méthodes de synthèse

Tout au long du développement de la théorie de la commande, le problème de la commande (cf. section 1.4) a été traité par plusieurs approches de résolution. Dans cette section, nous rappellerons les grandes classes de ces méthodes, en donnant leurs avantages et inconvénients.

### 1.6.1. Méthodes de synthèse synthétiques

Pour les méthodes synthétiques, le correcteur est, initialement, bâti autour d'un correcteur simple. Dans le but d'atteindre les objectifs de commande, la structure de ce correcteur est augmentée par l'ajout de plusieurs parties selon les objectifs requis (filtre réjecteur pour annuler l'effet d'une résonance, filtre passe-bas pour réduire le bruit de mesures, filtre avance de phase pour augmenter la phase de la boucle ouverte...). Le correcteur peut ainsi être structuré en plusieurs sous-systèmes ce qui est l'un des avantages majeurs des méthodes synthétiques. Le correcteur est alors synthétisé étape par étape, avec la possibilité d'utiliser plusieurs outils et techniques aux différentes étapes [Nyq32, Bla34,

Bod45, Hor63, Mac77, Mac79, Hor82] (tracé de Bode, lieu de Nyquist, abaque de Black, lieu des pôles/zéros...).

Les méthodes de synthèse synthétiques trouvent leur application, principalement, pour les systèmes qui acceptent plusieurs correcteurs relativement simples vérifiant les objectifs de commande. Les principaux avantages de ces méthodes sont : une complexité réduite des correcteurs, une structure de correcteurs modulaire, une efficacité surtout avec des objectifs de commande simples, une économie en terme de temps. Néanmoins, elles présentent également quelques inconvénients : impossibilité de chiffrer la limite des performances, nécessité d'une bonne connaissance du système à commander, difficulté rencontrée pour les systèmes multivariables et d'ordres élevés. Pour des systèmes plus compliqués, les avancées dans le domaine de l'optimisation et les méthodes de calcul numérique permettent l'utilisation de méthodes d'optimisation paramétrique très sophistiquées pour calculer ou affiner le réglage de ces correcteurs.

### 1.6.2. Méthodes de synthèse modernes (LQ/LQG/LTR)

Les méthodes de synthèse analytiques sont des méthodes basées sur une solution analytique d'un problème de commande optimale, par exemple, la commande Linéaire Quadratique Gaussienne (LQG) qui est basée sur une résolution d'une équation de Riccati algébrique. Les solutions analytiques ne sont valables que pour des problèmes de synthèse très spécifiques, l'ingénieur doit donc essayer de formuler son problème de commande sous la forme d'un problème analytiquement solvable tout en considérant, aussi finement que possible, les spécifications du cahier des charges [Kal61, Ath66, Kwa72, Bry75, And90]. Les techniques existantes pour la synthèse de ces commandes optimales supposent la sélection des matrices de pondérations [Bry75] (comme dans le cas LQG), l'ajout de bruits fictifs ou encore de dynamiques fictives dans le modèle du système [Kwa72, Doy81].

L'un des avantages significatifs des méthodes analytiques est que, si le problème de commande optimale est raisonnablement bien posé, le contrôleur optimal résultant tend à garantir, au pire des cas, ou à améliorer, les objectifs originaux de la synthèse. En particulier, les correcteurs analytiquement synthétisés garantissent la stabilité du système à commander, ce qui représente un atout capital dans le cas de systèmes multivariables instables où l'ingénieur automatique aura du mal à utiliser facilement son expertise comme dans le cas des méthodes synthétiques.

Le principal défaut des méthodes analytiques est la difficulté de bien formuler le problème de commande original sous la forme d'un problème de commande analytiquement solvable. Lors de la conception du problème de commande optimale, l'ingénieur automatique sera ramené à reformuler le problème original dans le cadre de la méthode analytique. Cette tâche est compliquée par le fait qu'il peut être plus difficile d'employer son intuition et ses compétences pratiques afin d'ajuster des matrices de pondération et des dynamiques fictives. Les correcteurs synthétisés par des méthodes analytiques sont souvent complexes : ordres élevés, interconnexions dans le cas de systèmes multivariables. Cette complexité obscurcit l'éventualité d'une équivalence à un simple correcteur qu'on pouvait synthétiser par les méthodes synthétiques. Ces méthodes sont souvent suivies par une phase de réduction du correcteur afin de déterminer le correcteur le moins complexe qui remplit les mêmes performances que le correcteur synthétisé analytiquement.

### 1.6.3. Méthodes de synthèse par optimisation

Au cours du développement de l'Automatique linéaire, l'optimisation a joué un rôle prépondérant dans l'évolution des techniques de commande. Plusieurs méthodes de la commande linéaire sont établies à partir de la théorie de l'optimisation et ses différents algorithmes.

Une première catégorie de techniques de commande à base d'optimisation est celle de la commande optimale (LQ et LQG). Ces méthodes traitent en particulier les problèmes de commande multivariables avec des spécifications temporelles et se formulent comme un problème sans contraintes de minimisation convexe avec un critère quadratique pondéré rassemblant les spécifications de poursuite et les limitations sur les énergies de commandes. Sous certaines conditions, la résolution analytique de ce type de problème est effectuée via une résolution d'une équation algébrique de Riccati.

Dans la même catégorie, la synthèse  $H_\infty$  présente des solutions aux problèmes de commande robuste via une formulation fréquentielle du cahier des charges. Elle se base sur l'utilisation des concepts fondamentaux de l'Automatique fréquentielle classique : le cahier des charges est traduit comme des gabarits sur les modules des fonctions de transfert en boucle fermée (voir les paragraphes 1.5.2 et 1.5.3). Son formalisme mathématique est basé sur la norme  $H_\infty$  pondérée. Le problème d'optimisation formulé est convexe, il assure une optimalité globale du correcteur synthétisé. Ce type de problèmes peut être résolu par deux méthodes : soit analytiquement, sous certaines conditions, via une résolution d'une équation de Riccati (Problème  $H_\infty$  standard) [Glo88, Doy89a], soit à travers la formulation d'un problème d'optimisation faisant intervenir des inégalités matricielles linéaires<sup>5</sup> (problème LMI) [Gah94, Iwa94, Zho95]. Certains cahiers des charges mixtes, avec des spécifications temporelles et fréquentielles, peuvent être formulés sous forme de problèmes LMI, on cite les techniques  $H_2/H_\infty$  [Doy89b, Sch95, Hin98].

Une deuxième formulation du problème de commande sous forme d'un problème d'optimisation est possible. Elle consiste à formuler un problème d'optimisation paramétrique afin de réaliser des spécifications plus génériques et qui ne se formulent pas sous forme d'un problème d'optimisation analytiquement solvable (convexe). L'idée de ce type de formulation remonte à l'ère de l'Automatique fréquentielle classique. En effet, les premières utilisations de l'optimisation paramétrique sont nées suite au problème de structure de correcteurs qui est présent dans la plupart des approches classiques et modernes. Ce problème était particulièrement motivé par les compromis entre les différentes spécifications. Les approches développées dans ce sens, se basent initialement sur une structure paramétrique du correcteur. Par exemple : la structure d'un correcteur proportionnel intégral  $K_p + K_i/p$  où les paramètres à ajuster sont  $K_p$  et  $K_i$ , ou également la structure d'un correcteur d'ordre fixe dans l'espace d'état ; les paramètres sont alors tous les éléments des matrices de la forme d'état mises sous une forme canonique donnée.

Une troisième approche consiste à définir un critère pour l'analyse ou l'optimisation générale des qualités du système commandé. Une première façon de faire consiste à choisir ce critère à partir d'un problème solvable analytiquement comme dans le cas de la commande LQG. Ceci aura l'avantage de pouvoir comparer le critère optimal, associé au correcteur optimal, à tout autre minimum global réalisé par n'importe quel contrôleur analytiquement calculable. Une deuxième façon de faire consiste à

<sup>5</sup> La solution par équations de Riccati est antérieure à la solution par optimisation LMI. Cependant, la démarche adoptée pour l'obtenir est beaucoup moins générale.



définir la fonction critère comme une somme pondérée ou un maximum de plusieurs indices de performance (l'intégrale quadratique de l'erreur d'une réponse à un échelon, l'intégrale du module d'un transfert sur une bande de pulsations où le bruit est concentré, l'amplitude maximale du signal de commande,...). Finalement, il est possible d'ajouter des contraintes explicites comme des plages de variation sur les paramètres du correcteur ou des limites pour le placement des pôles de la boucle fermée ou encore des gabarits sur le module du transfert en boucle ouverte...etc. Une fois que la structure du correcteur, le critère et les éventuelles contraintes sont spécifiés, l'ingénieur aura à résoudre un problème d'optimisation non linéaire et surtout non convexe [Boy90].

Dans la littérature, plusieurs techniques d'optimisation numérique ont été développées pour résoudre le problème de commande. Certaines de ces techniques ne sont que des algorithmes heuristiques simples tels que les méthodes de descente. Quelques-unes se basent sur des algorithmes plus spécialisés pour des catégories de problèmes bien spécifiques, d'autres sont des boîtes à outil sophistiquées pour traiter de très larges classes de problèmes. À titre d'exemples, nous citons les travaux suivants : une analyse étendue sur la synthèse itérative des correcteurs LQG dans [Doy81], "SANDY" une technique à base de gradients pour la recherche de correcteurs optimaux d'ordre fixe [Lyu83], un recueil de travaux sur le problème de recherche de correcteurs optimaux à retour d'état statique et d'ordre fixe à partir d'un critère LQG [Mak87], une technique d'optimisation paramétrique appliquée aux problèmes de commande en avionique [Gan86], une boîte à outil interactive de simulation et d'optimisation pour des problèmes de commande avec contraintes paramétriques [Pol84, Pol85, Fan89]. Pour une vision plus globale concernant le choix des algorithmes d'optimisation non linéaire et leurs implémentations, le lecteur peut se reporter à [Gil81].

Plus récemment, on peut citer les méthodes de synthèse à base d'algorithmes métaheuristiques [Tan98, Lem02], les travaux de retouche de correcteurs à base de l'identification Bayésienne [Mou02, Del05], les travaux de K.J. Åström sur la synthèse de correcteurs PID [Ast05], les travaux de D. Henrion concernant le problème de retour de sortie statique [Hen05, Hen06] ainsi que les travaux de J.V. Burke et P. Apkarian sur la synthèse de correcteurs de retour de sortie par optimisation non différentiable [Bur06, Apk05, Apk06].

### **1.6.3.1. Classification (convexité et dimension du problème)**

Dans le cadre de la résolution du problème fondamental de commande, les approches d'optimisation peuvent être divisées en deux grandes catégories : les approches convexes et les approches non convexes.

Dans le but de répondre à la faisabilité du cahier des charges étudié, les approches convexes établissent le problème de commande sous la forme d'un problème d'optimisation avec un optimum global (le correcteur optimal obtenu est global). Dans le cas où cette formulation est possible, la résolution du problème formulé nécessite, par la suite, des algorithmes efficaces.

Le meilleur exemple d'une telle classe de problème est la commande via la résolution de LMI. Cette technique permet de répondre à une formulation fréquentielle des spécifications de commande. Historiquement, sa première formulation était à base des techniques fréquentielles classiques ce qui a mené à un problème d'optimisation convexe dit problème  $H_\infty$  standard. Malgré la convexité du problème formulé, sa résolution n'était pas facile, vu que sa dimension était infinie (les contraintes sur les modules des transferts dépendent de la pulsation  $\omega$ ). Afin de surmonter cette difficulté, une

première formulation équivalente à ce problème sous la forme d'une équation de Riccati a été proposée par [Doy89a]. A partir de cette formulation et pour une classe moins générale de systèmes, une solution au problème de synthèse  $H_\infty$  a été proposée [Glo88, Doy89a]. Dans la même optique d'obtenir une formulation de dimension finie du problème, une démarche était de passer en représentation d'état. D'une part, pour un ordre donné, le correcteur est alors paramétré par les matrices de sa représentation d'état (soit un nombre fini de variables de décision). D'autre part, un résultat fondamental, le lemme borné réel (cas particulier du lemme de Kalman-Yakubovitch-Popov [Zam66]) des années 60 permet de transformer la vérification des contraintes infinies (qui doit être vraie pour toute pulsation  $\omega$ ) en la résolution d'un problème d'optimisation de dimension finie [Saf80].

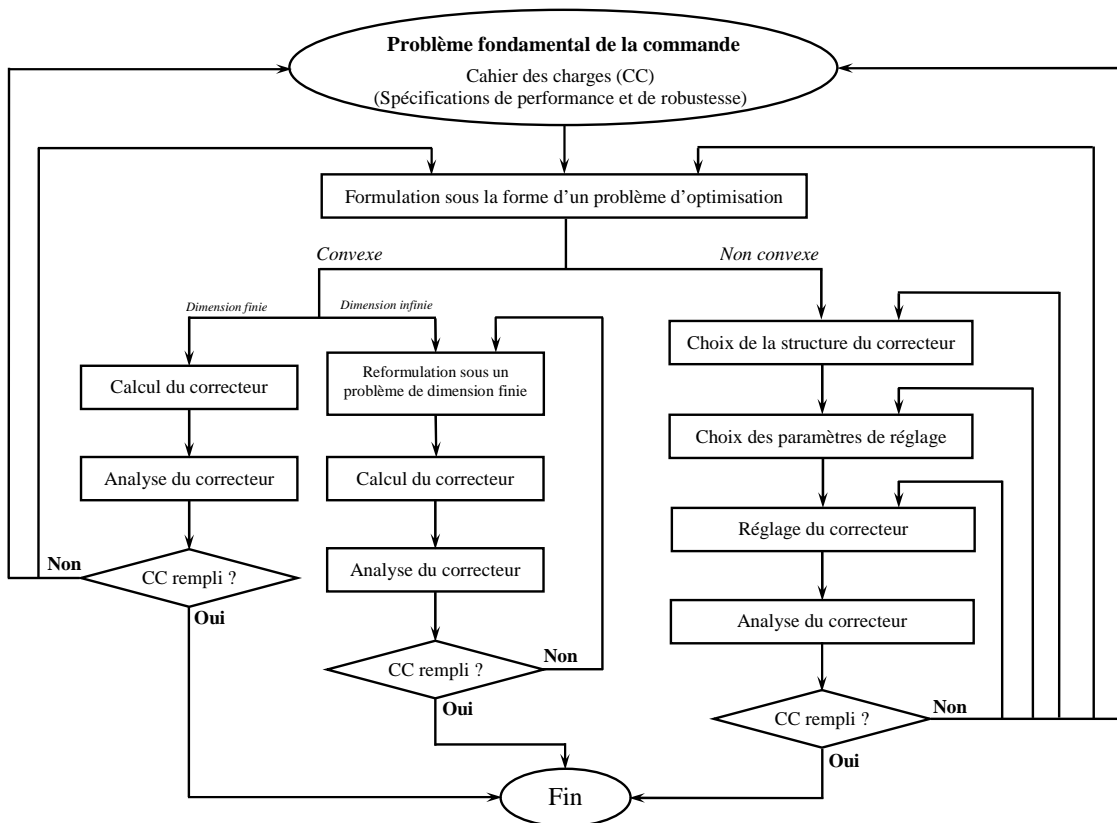


Fig. 1.27: Structure générale d'une synthèse par optimisation

En revanche, les problèmes d'optimisation à structure fixe du correcteur sont généralement non convexes. Une fois que la structure est choisie, les paramètres du correcteur sont ajustés de façon à ce que le cahier des charges soit rempli (étape "Réglage des paramètres du correcteur" sur la figure 1.27). Ceci est fait à l'aide d'une stratégie d'optimisation choisie. A l'issue de cette étape, il est impératif de vérifier si le correcteur obtenu remplit bien le cahier des charges initial (étape "Analyse du correcteur" sur la figure 1.27). S'il n'est pas vérifié alors soit il n'existe pas de correcteur remplissant le cahier des charges, soit un mauvais choix a été fait à l'une des étapes du processus de conception :

- mauvaise traduction des spécifications du cahier des charges (choix du critère par exemple)
- mauvais choix de structure pour le correcteur ;
- mauvais choix de paramètres pour le correcteur.

Malgré l'abondance des possibilités, la synthèse par optimisation représente un outil très puissant qui, associé au savoir-faire de l'ingénieur, permet d'identifier clairement ; pourquoi le correcteur obtenu ne remplit pas le cahier des charges, où le processus de conception doit être repris, et enfin comment il doit être modifié.

## 1.7. Vers une approche d'optimisation non linéaire

Les applications industrielles sont plus que jamais au cœur des enjeux actuels de l'Automatique. On observe une augmentation constante des exigences de qualité et de performance des systèmes asservis et l'exigence de plus en plus forte du "meilleur compromis". Les cahiers des charges d'asservissements industriels sont de plus en plus complexes et serrés.

Quand elles sont applicables, les méthodes de synthèse convexes permettent de garantir une réponse définitive au problème de commande ; dans le cas affirmatif le cahier des charges est approuvé et le correcteur est explicitement donné, dans le cas contraire, c'est la formulation du cahier des charges qui est souvent mise en question. Même si ces méthodes de synthèse paraissent plus adaptées et directes pour répondre au problème de commande, le processus de conception de la loi de commande reste par nature itératif du fait de la formulation approchée des spécifications du cahier des charges comme dans le passage assez approximatif des spécifications temporelles aux contraintes fréquentielles sur le module des fonctions de transfert en boucle fermée (synthèse  $H_\infty$ ).

En terme de complexité calculatoire, les reformulations des problèmes convexes de dimension infinie en problèmes de dimension finie (cf. figure 1.27) se font souvent par certaines approximations (une base finie de pulsations ou de temps). Toutes ces approximations impliquent que le correcteur optimal obtenu peut ne pas satisfaire les spécifications du cahier des charges initial.

Dans bien des cas, il n'est pas possible d'obtenir une formulation exacte ou même suffisamment approchée du problème de commande sous la forme d'un problème d'optimisation convexe ou d'une méthode analytique. Même dans le cas où cela est partiellement faisable, une formulation comme un problème d'optimisation générique est souvent nécessaires pour satisfaire aux exigences fine du cahier des charges lors d'une second phase de retouche de correcteur (contrainte sur la structure de commande par exemple). Cette approche générique permet également de prendre en compte dans le processus de synthèse l'évolution ou l'ajustement permanent du cahier des charges par des retouches successives.

## 1.8. Conclusion

Un bilan des différentes spécifications d'un cahier des charges régissant le problème fondamental de la commande linéaire a été dressé. Ces spécifications sont formulées selon leur nature dans le domaine temporel et/ou fréquentiel via des critères algébriques et graphiques tirés en majorité de la théorie des fonctions complexes.

Un bref état de l'art des méthodes de synthèses nous montre que pour la plupart, elles consistent à exprimer des cahiers des charges d'un type donné sous forme d'optimisation convexe ( $H_\infty$ ,  $H_2$ , LQG...). Les algorithmes correspondants sont efficaces en terme de résolution mais ils ne traitent que de cahiers des charges particuliers adaptés à chaque méthode considérée.

Dans le but de répondre aux exigences accrues des industriels et pour une meilleure prise en compte d'un cahier des charges, il est possible de ne pas partir d'un critère particulier mais de formuler le problème de synthèse comme un problème d'optimisation général regroupant l'ensemble de toutes les spécifications (même structurelles). Dans ce cas, les demandes sont formulées plus précisément mais les problèmes d'optimisation obtenus sont non convexes et très complexes. Il est alors nécessaire d'analyser les propriétés de ces problèmes afin de développer des algorithmes de calcul de correcteurs les plus simples et les plus efficaces possibles.

L'objectif des chapitres suivants est de montrer que, sous la forme de logiciels de conception assistée par ordinateurs, des outils basés principalement sur l'optimisation générale, fusionnant les critères mathématiques et graphiques cités auparavant, peuvent être développés. Leur ambition est d'aborder les enjeux industriels actuels et de répondre au problème fondamental de commande avec des cahiers des charges génériques.



# Chapitre 2

## Formulation et analyse de cahier des charges par optimisation

Dès la fin des années 80, de nombreux développements ont porté sur les possibilités d'utiliser l'optimisation pour des besoins de l'ingénierie et en particulier sur les difficultés liées à la complexité calculatoire de ces problèmes d'optimisation une fois dressés. Dans le domaine de l'Automatique, ces développements concernaient les méthodes d'optimisation convexe qui s'appliquent, en général, pour des techniques de synthèses de correcteurs.

Par ailleurs, la méconnaissance des limites de performances atteignables par un système bouclé ainsi que l'aspect antagoniste des spécifications de commande ont fait du dimensionnement et de la validation du cahier des charges des vrais challenges. Aujourd'hui, la faisabilité d'un cahier des charges pour un système est un nouveau thème de recherche auquel s'attachent fortement les industriels, qui sont souvent confrontés à des contraintes en performance complexes qui sont exprimées, simultanément, dans différents domaines (contraintes de normes, d'encadrement, temporelles, fréquentielles,...).

Dans cette perspective, des études ont permis de formuler une partie des besoins de l'ingénieur automatique sous formes de problèmes d'optimisation convexes de dimension infinie [Boy91]. Ces derniers traitent la question d'existence et d'unicité d'un correcteur optimal qui respecte les exigences des cahiers des charges. Malheureusement, dans le cas général, les critères formulés ne peuvent être résolus qu'en dimension finie. Cette approximation ainsi que les reformulations du cahier des charges d'origine sous forme convexe produisent des problèmes solvables en temps fini, mais approchés (dans le cas présent c'est bien le problème qui est une approximation). Bien que ces techniques soient efficaces, leur application ne peut être étendue pour tous types de demandes.

Dans notre approche, l'accent est mis sur les méthodes constructives, qui peuvent être développées en dehors du contexte convexe des problèmes d'optimisation en Automatique. Elle peut prendre en compte à la fois des contraintes sur des trajectoires dans le domaine temporel et des contraintes sur des transferts en boucle ouverte ou du système de commande. L'approche proposée permet donc une "vraie" formulation des exigences au détriment d'une analyse exacte de leurs faisabilités.

Une des applications possibles de cette approche, qui semble également d'un grand intérêt pour les industriels, est la retouche de correcteurs. Elle consiste à modifier les paramètres du correcteur déjà synthétisé afin de valider le cahier des charges d'origine. Cette optique devient très intéressante lorsqu'il s'agit d'éviter de parcourir toutes les étapes d'un processus de synthèse long et compliqué. En outre, elle est très utile dans le cas d'une modification du modèle de synthèse ou d'une évolution

des spécifications en termes de performances et/ou de robustesses. De nos jours, la retouche de correcteurs présente de grands domaines d'application tels que : l'aéronautique, l'aérospatiale, l'énergie.

Dans ce chapitre, nous montrons les particularités de cette approche globale qui résulte d'une part de la nature des variables utilisées en automatique (temporelle, fréquentielle, implicite, explicite...) et d'autre part de la non différentiabilité de certaines fonctions souvent utilisées lors de la formulation directe des problèmes d'optimisation. Une analyse algorithmique de cette spécificité montre le besoin accru d'une méthode d'optimisation adéquate pour la résolution des problèmes d'optimisation résultants.

Le chapitre débute par la description d'un formalisme général qui permet de traduire explicitement les spécifications génériques du cahier des charges en un problème d'optimisation de type problème de faisabilité. Les principes des différentes méthodologies existantes pour le traitement de tels problèmes sont détaillés et comparés entre eux afin de proposer une démarche efficace en terme de résolution. En se basant, essentiellement, sur une traduction directe des spécifications en des contraintes implicites par rapport aux paramètres de commande, notre approche peut être qualifiée d'intuitive et générique. Elle permet de regrouper toutes les contraintes en un problème d'optimisation paramétrique non linéaire sans contraintes.

Le problème formulé est difficile car il est non convexe, multi-objectif par construction. Son estimation doit mêler des stratégies de nature différente puisqu'il fait intervenir des estimations intermédiaires de type explicite, implicite, fréquentiel et temporel. Ainsi, la complexité mathématique du problème d'optimisation résultant est ensuite étudiée afin d'identifier les techniques numériques susceptibles de le résoudre avec efficacité. Cette étude concerne la complexité calculatoire des critères établis ainsi que leurs variations par rapport aux paramètres du correcteur. Elle dévoile la nécessité d'une résolution adaptée au problème d'optimisation formulé qui présente la propriété d'être presque partout différentiable.

## 2.1. Concepts fondamentaux pour une approche générique

La formulation des cahiers des charges découle de l'application directe du formalisme décrit dans le premier chapitre. Elle correspond à une tentative de réponse aux deux principales questions constituant notre problématique :

- Dans quel cadre mathématique précis devons-nous formuler le problème de commande pour avoir une chance d'obtenir des critères génériques, pertinents et calculables ?
- Quelle approche unifiée devons-nous adopter pour une résolution efficace de ces critères génériques ?

L'approche d'analyse d'un cahier des charges consiste à proposer un critère mathématique permettant de garantir le bon comportement du correcteur pour la validation des différentes spécifications. Pour qu'elle soit viable, cette approche doit traiter une traduction adéquate des exigences du système bouclé en un temps raisonnable.

La motivation principale est d’offrir un outil d’aide à la conception utilisable par l’ingénieur pour traiter les problèmes industriels de commande. Cela correspond pratiquement à trois exigences :

*Généralité* : un même cadre mathématique doit permettre d’aborder des systèmes avec des spécifications de natures différentes : linéarité, non linéarité, performance fréquentielle, performance temporelle, etc. En outre, un problème de commande doit pouvoir être traité de façon modulaire.

*Accessibilité* : un même formalisme doit permettre de traiter de façon transparente un problème d’Automatique sans connaissances approfondies et exhaustives des résultats théoriques ; l’ingénieur automatique doit pouvoir mettre en œuvre des outils proposés sans maîtrise des concepts avancés ou d’un vocabulaire spécial.

*Calculabilité* : un test de faisabilité d’un cahier des charges n’est pertinent que dans la mesure où il est calculable lors de sa mise en œuvre sur un problème industriel : il doit être applicable avec un temps de calcul raisonnable.

Un des objectifs de ce travail est de démontrer qu’il est actuellement possible de conjuguer les résultats obtenus en théorie de l’Automatique, les nouvelles méthodes d’optimisation non convexe et la puissance de calcul grandissante des ordinateurs pour un coût de plus en plus faible, pour lancer les bases d’un outil industriel efficace faisant des compromis intéressants entre les trois exigences précédentes.

## 2.2. Cahiers des charges et critères mathématiques

Un point clé de l’analyse de la faisabilité d’un cahier des charges est le problème de traduction mathématique des spécifications de commande. Ce problème est très complexe. En effet, généralement deux types de spécifications peuvent être distingués :

- i. Les spécifications d’ordre *qualitatif* : par exemple, “la suspension automobile doit assurer le confort des passagers”;
- ii. Les spécifications chiffrées ou d’ordre *quantitatif* : par exemple, “la suspension automobile doit rejeter les perturbations en tant de secondes”.

Le cadre mathématique qui nous intéresse ne permet de traiter que les problèmes quantitatifs. Le problème est donc de ramener des spécifications du domaine qualitatif dans le domaine quantitatif. Par exemple, il semble difficile de proposer une mesure du confort des passagers. Par contre, il est possible de mettre au point un indicateur quantitatif représentant le confort du passager d’une voiture. Le problème reste ouvert et est laissé à la discrétion de l’ingénieur. Le deuxième type de spécifications est naturellement de nature quantitatif. Néanmoins, reste le problème de la pertinence de la spécification, même quantitative, par rapport aux critères mathématiques aisément calculables. Une pratique courante consiste à définir des indicateurs temporels tels que le temps de réponse, le dépassement ou l’erreur quadratique moyenne sur la réponse du système bouclé à des signaux de consigne typiques (comme des échelons) ou autrement, à imposer des gabarits temporels à ces réponses. Malheureusement, il n’existe pas à notre connaissance, de critères qui seraient une traduction directe de ces spécifications et qui aboutiraient à des méthodes simples et efficaces d’analyse et de commande. Cependant, pour les systèmes linéaires, il est souvent possible en pratique



de spécifier indirectement ces objectifs à l'aide par exemple de contraintes sur l'emplacement des pôles ou sur la réponse fréquentielle des fonctions de transfert du système. La spécification indirecte induit alors lors de la synthèse un processus essais/erreurs.

Voyons brièvement les principales approches du problème de traduction des spécifications.

Dans les approches de type commande optimale, la performance est mesurée par un compromis entre la quantité interprétée comme l'énergie dépensée pour la commande et une quantité interprétable comme l'énergie d'erreur de poursuite ou encore l'énergie de la sortie par rapport à un bruit en entrée. Pour la commande des systèmes linéaires stationnaires, Zames a proposé de mesurer la performance en regardant dans le domaine fréquentiel les propriétés d'atténuation ou d'amplification des fonctions de transfert du système [Zam81]. Mesurer la performance revient alors à vérifier que la norme  $H_\infty$  de certaines fonctions de transfert du système pondérées par un gabarit fréquentiel est inférieure à un certain seuil. La performance s'interprète donc en terme de désensibilisation. Malheureusement, la performance d'un système ne se réduit pas seulement à des contraintes de type fréquentiel. Il est connu que les contraintes dans le domaine temporel ne sont pas reliées simplement à celles en fréquentiel. Néanmoins, la commande  $H_\infty$  donne généralement en pratique des résultats satisfaisants [Fon95].

Les succès rencontrés par la commande  $H_\infty$  ont motivé son extension à la commande des systèmes non linéaires. Une première approche consiste à exploiter le fait que la norme  $H_\infty$  est une norme induite et que, de ce fait, son extension naturelle aux systèmes non linéaires est obtenue avec la norme induite deux (le gain  $\ell^2$  [Sch92, Iso95]) cela permet de traduire les spécifications en terme d'atténuation énergétique entre des signaux exogènes agissant sur le système et certains de ses signaux de sortie. En ces termes, peut être traduit le fait qu'il est désirable qu'une erreur de poursuite possède une énergie faible comparée aux énergies des signaux de perturbation et/ou de consigne [Fro95a].

Une autre généralisation possible de l'approche  $H_\infty$  est de formuler le problème de la performance d'un système comme la vérification d'une norme incrémentale pondérée par certains opérateurs entre des entrées et des sorties du système non linéaire [Fro95b]. Elle a été proposée par Fromion comme étant une extension de l'approche de Zames au cadre non linéaire. Là encore, la performance est exprimée en termes de désensibilisation. Elle peut s'interpréter comme une exigence de performance sur les linéarisations non stationnaires du système le long de ses trajectoires. De plus, elle repose sur le fait que la réponse d'un système à un signal d'entrée est obtenue par la somme de ses réponses à de petites variations en entrée [Fro95a].

Dans ce mémoire, pour des raisons de calculabilité, la performance sera en général estimée pour les systèmes non linéaires par des critères temporels sous forme d'encadrement (critère  $\ell^\infty$ ), des critères  $\ell^1$  ou  $\ell^2$  et d'une manière plus directe par des indicateurs de type temps de réponse.

### 2.3. Traduction du cahier des charges

La validation du cahier des charges d'un problème de commande revient à trouver les paramètres du correcteur qui permettent de satisfaire toutes ses spécifications. Ainsi, temporellement, à chaque correcteur solution correspondent des trajectoires de commande associés aux différents signaux de consigne qui, appliqués en entrées du système, lui confèrent un comportement satisfaisant en sortie.

De même, dans le domaine fréquentiel, à chaque correcteur solution correspondent des transferts en boucle ouverte et fermée respectant certains gabarits et certaines marges de stabilité et des pôles et des zéros respectant un certain lieu des racines. L'existence de tels trajectoires et transferts est une condition nécessaire pour l'existence du correcteur solution.

En se basant sur les différentes définitions des spécifications de commande introduites au premier chapitre, nous pouvons les grouper en deux grandes catégories selon leur appartenance au domaine temporel ou fréquentiel.

### 2.3.1. Expression des contraintes temporelles E/S du système

Dans ce principe de formulation en trajectoires, il s'agit d'analyser l'existence de trajectoires E/S d'un système bouclé vérifiant un certain nombre de contraintes qui dépendent implicitement du correcteur  $K$ . Précisément, à partir d'un problème de commande décrit sous forme de spécifications imposées au système bouclé, nous proposons un cadre permettant la formulation d'un nouveau problème dont les contraintes concernent les trajectoires E/S du système à corriger. La prise en compte de la structure de correcteur  $K$ , dans notre vision trajectoires, la transforme en une approche équivalente de synthèse.

La mise en commun des contraintes E/S formulées traduit le problème de faisabilité du cahier des charges en un problème d'optimisation qui permet d'étudier les limites de performances que peut atteindre le système et d'aider à la construction du cahier des charges.

Nous considérons, dans notre étude, deux approches différentes pour la formulation des contraintes temporelles :

#### 2.3.1.1. Formulation directe des spécifications temporelles

Dans cette formulation, les spécifications définissent les caractéristiques classiques concernant la dynamique de certains signaux de la boucle fermée. Dans le cas d'un problème de poursuite par exemple, il s'agit d'évaluer la précision du système en boucle fermée en étudiant l'influence du signal de référence  $r(t)$  sur le signal d'erreur  $\varepsilon(t)$ .

Lorsque le système bouclé est stable et pour un signal de référence de type échelon, le régime transitoire du signal d'erreur  $\varepsilon(t)$  est caractérisé par un ou plusieurs indicateurs de rapidité et d'amortissement :

- Pente initiale ;  $\delta_0 = -\dot{\varepsilon}(t=0)$
- Temps de montée ;  $T_{monté} = \{T_{10\%} - T_{90\%} \mid \varepsilon(T_{10\%}) = 0,1 \cdot r(t) \text{ et } \varepsilon(T_{90\%}) = 0,9 \cdot r(t)\}$
- Dépassement ; si  $\exists \tau > 0 \mid \varepsilon(\tau) < 0$ , alors  $D = -\min_t \{\varepsilon(t)/r(t)\}$
- Valeur maximale ( $\ell^\infty$ ) ;  $y_{\max} = \max_t (r(t) - \varepsilon(t))$
- Temps du premier maximum ; si  $\exists \tau > 0 \mid \varepsilon(\tau) < 0$ , alors  $T_{\max_1} = \inf_{T>0} \{T \mid \dot{\varepsilon}(t=T) = 0\}$
- Temps d'établissement à  $\alpha\%$  ;  $T_e = \inf_{T>0} \{T \mid \forall t > T : |\varepsilon(t)| \leq \alpha/100 \cdot r(t)\}$
- Valeur finale ;  $y_\infty = \lim_{t \rightarrow \infty} (\varepsilon(t) - r(t))$

La performance d'une loi de commande est aussi évaluée par sa précision dynamique. Cette dernière est souvent mesurée par l'un des critères intégraux suivants :

- L'intégrale de moyennes quadratiques (ISE) ;  $J_{ISE} = \int_{t=0}^{+\infty} \varepsilon(t)^2 dt$

- L'intégrale de moyennes absolues (IAE) ;  $J_{IAE} = \int_{t=0}^{+\infty} |\varepsilon(t)| dt$
- L'intégrale de moyennes quadratiques pondérées par le temps (ITSE) ;  $J_{ITSE} = \int_{t=0}^{+\infty} t \cdot \varepsilon(t)^2 dt$
- L'intégrale de moyennes absolues pondérées par le temps (ITAE) ;  $J_{ITAE} = \int_{t=0}^{+\infty} t \cdot |\varepsilon(t)| dt$

Ces critères de précision à minimiser ainsi que l'ensemble d'indices de performance, précités, peuvent bien être définis pour d'autres signaux de la boucle comme dans le cas d'un découplage pour les systèmes multivariables. D'une façon générale, les spécifications directes du cahier des charges sur ce type d'indicateurs se transcrivent dans le domaine temporel soit par un critère à minimiser soit par une contrainte de type minmax sous la forme suivante :

$$\alpha_i^{\min} \leq \alpha_i \leq \alpha_i^{\max} \quad (2-1)$$

où  $\alpha_i^{\min}$  et  $\alpha_i^{\max}$  sont, respectivement, la borne minimale et la borne maximale de l'indicateur de performance  $\alpha_i$ .

### 2.3.1.2. Formulation des spécifications temporelles sous forme de gabarits

Certaines contraintes temporelles peuvent se traduire sous forme d'enveloppes temporelles appliquées aux différents signaux appropriés [Boy91]. Cette formulation est plus large, elle permet de regrouper les contraintes concernant les différents signaux du système bouclé telles que les spécifications de performance d'un cahier des charges industriel ; les contraintes de poursuite, les contraintes de rejet des perturbations, les limitations d'amplitudes et de vitesses de variations ou encore les contraintes de découplage dans le cas des systèmes multivariables.

D'autre part, cette approche permet de traduire parfaitement les contraintes associées aux indicateurs de performance (temps de réponse, dépassement,...) et d'éviter ainsi le recours à des calculs complexes. Elle permet de formuler fidèlement le cahier des charges d'un problème de commande sous forme de contraintes imposées aux différents signaux du système bouclé.

Dans un cas général, une enveloppe temporelle à un signal  $s(t)$  de la boucle de commande consiste à le contraindre à varier entre deux gabarits, minimal noté  $s^{\min}(t)$  et maximal noté  $s^{\max}(t)$ . L'ensemble des signaux vérifiant cette contrainte s'écrit donc :

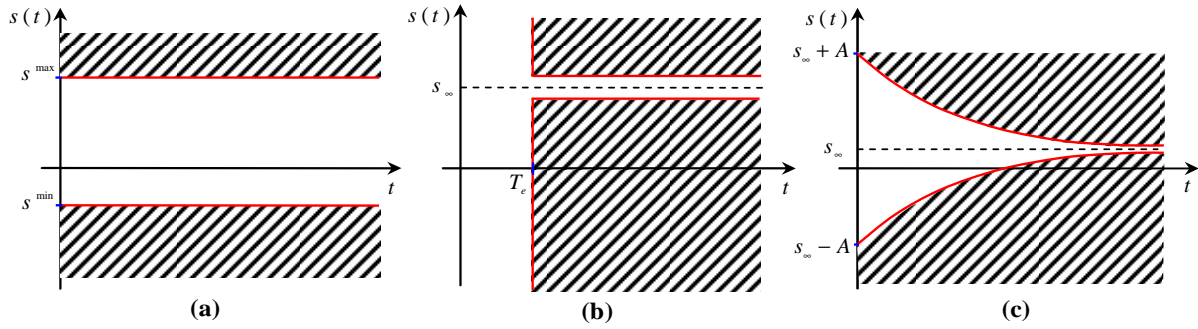
$$C_s = \{s(t) \mid s^{\min}(t) \leq s(t) \leq s^{\max}(t), \forall t\} \quad (2-2)$$

où  $s(t)$  peut désigner n'importe quel signal de la boucle fermée ou l'une de ses dérivées.

Toutes ces contraintes ou objectifs peuvent également être exprimés par un modèle de référence  $s_{ref}(t)$  par rapport auquel on cherchera à minimiser la différence  $\Delta s(t) = s(t) - s_{ref}(t)$ . Typiquement, on dénombre trois classes élémentaires d'enveloppes temporelles. La superposition de ces enveloppes permet de construire d'autres types d'enveloppes selon les exigences désirées.

#### *Enveloppe temporelle de type minmax*

Il s'agit de contraindre le signal  $s(t)$  à être borné ; c'est-à-dire à varier entre une valeur minimale  $s^{\min}$  et une valeur maximale  $s^{\max}$ . Les deux gabarits formant l'enveloppe sont donc des fonctions constantes (cf. figure 2.1(a)).



**Fig. 2.1 Envelopes temporelles élémentaires**

Elle peut aussi exprimer des spécifications classiques de dépassement en choisissant pour gabarits constants, minimal et maximal respectivement, les valeurs :  $s^{\min} = \min(s(t))/s_\infty$  et  $s^{\max} = \max(s(t))/s_\infty$ , où  $s_\infty$  désigne la valeur asymptotique imposée à la sortie considérée  $s(t)$ .

De plus, exprimer la variation bornée d'un signal  $s(t)$  se traduit par une contrainte sur la dérivée par rapport au temps de ce signal. Nous écrivons souvent :

$$|\dot{s}(t)| < M, \quad \forall t \text{ (avec } M > 0) \quad (2-3)$$

Cette contrainte est représentée par une enveloppe minmax imposée à la dérivée du signal  $s(t)$  avec pour borne supérieure  $+M$ , et borne inférieure  $-M$ .

### **Enveloppe temporelle de temps d'établissement**

Le temps d'établissement est un indice de performance du système. Il mesure le temps à partir duquel le régime permanent souhaité est atteint. Cet indice de performance est imposé en sortie du système ; il suppose que la valeur asymptotique  $s_\infty$  de la sortie considérée est connue. La définition du temps d'établissement a été déjà énoncée dans le paragraphe 2.3.3.1. Elle se traduit en terme de gabarit par une enveloppe temporelle sur le signal  $s(t)$  définie par :

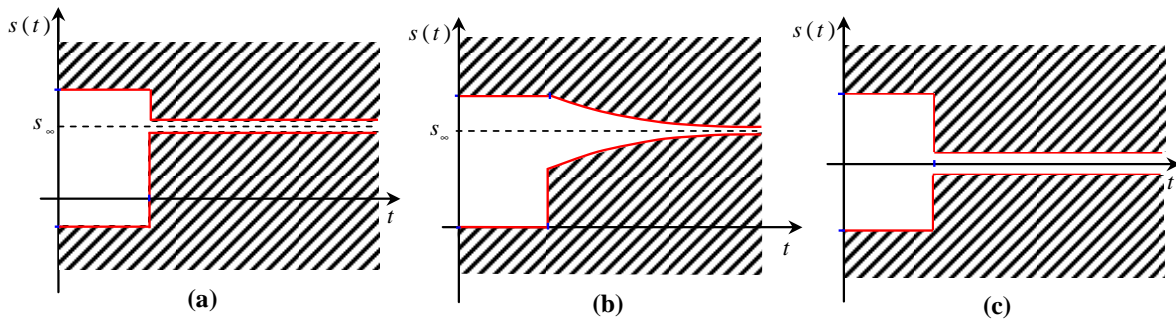
$$\begin{cases} s^{\min}(t) = \begin{cases} -\infty & \forall t \in ]-\infty, T_e[ \\ (1 - \alpha/100) \cdot s_\infty & \text{sinon} \end{cases} \\ s^{\max}(t) = \begin{cases} +\infty & \forall t \in ]-\infty, T_e[ \\ (1 + \alpha/100) \cdot s_\infty & \text{sinon} \end{cases} \end{cases} \quad (2-4)$$

La figure 2.1(b) représente ce type de gabarit. Dans le cas où le gain statique en boucle fermée est nul, cette contrainte peut exprimer une rejection de perturbation.

### **Envelopes temporelle de temps de réponse**

Les gabarits minimal et maximal imposés au signal  $s(t)$  peuvent être définis d'une façon plus générale. Par exemple, nous pouvons retrouver des gabarits de type temps de réponse définis à base de fonctions exponentielles (cf. figure 2.1(c)) :

$$\begin{cases} s^{\min}(t) = s_\infty - A e^{-t/\tau} \\ s^{\max}(t) = s_\infty + A e^{-t/\tau} \end{cases} \quad (2-5)$$



**Fig. 2.2 Enveloppes temporelles mixtes**

Ce type de contrainte peut à son tour traduire un rejet de perturbation. La dynamique de réjection sera donc fonction de la constante de temps  $\tau$ , et le degré d'atténuation fonction du paramètre  $A$ .

Quand plusieurs contraintes sont imposées à un même signal, l'enveloppe temporelle associée est obtenue en superposant les enveloppes élémentaires correspondant à chacune de ces contraintes. Dans le cas par exemple des contraintes simultanées de temps d'établissement et de dépassement, il s'agira donc de superposer les enveloppes de la figure 2.1(a) et 2.1(b). La contrainte obtenue est donnée par la figure 2.2(a). Dans le cas où la contrainte de temps d'établissement est remplacée par une contrainte de temps de réponse de type exponentielle, l'enveloppe temporelle résultante est donnée par la figure 2.2(b).

Un deuxième exemple de superposition de contraintes temporelles est la spécification représentée par la figure 2.2(c). La valeur du gain statique est imposée nulle. Cette spécification, imposée aux sorties, exprime le rejet d'un signal de perturbation sur une entrée du système. Dans le cas des systèmes multivariables, elle permet de traduire le découplage entre une entrée et une sortie donnée.

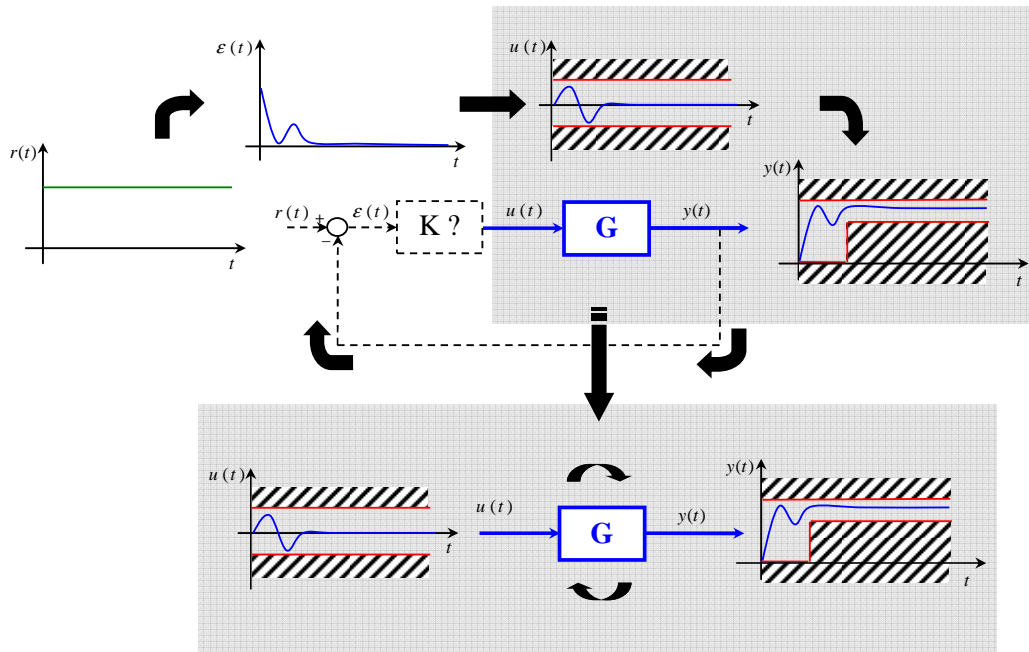
### 2.3.2. Faisabilité des cahiers des charges temporels

Considérons, par exemple, le cas du problème de commande représenté par la figure 2.3. Il s'agit de contraindre la sortie du système  $y(t)$  à suivre un échelon de consigne  $r(t)$  en respectant une enveloppe temporelle donnée ; le signal de commande  $u(t)$  étant aussi borné.

Une condition nécessaire à la faisabilité de ce problème est l'existence d'au moins une trajectoire de commande  $u(t)$  appartenant au gabarit de commande telle que, appliqué en entrée du système à commander  $G$ , elle permettrait l'obtention d'une sortie satisfaisante.

La trajectoire de sortie  $y(t)$  du système à commander  $G$  est l'image par ce système de la trajectoire de commande  $u(t)$ . Ainsi, il est clair qu'une contrainte au niveau de l'une, engendre une contrainte au niveau de l'autre. Par exemple, le temps de réponse minimum que peut atteindre le système en sortie peut varier en fonction des amplitudes maximales de commandes autorisées. Ce compromis, souvent difficile à chiffrer, présente l'une des difficultés rencontrées lors de la définition du cahier des charges.

Pour résoudre le problème d'expression des contraintes temporelles, nous distinguons deux approches : une approche d'analyse et une approche de synthèse.



**Fig. 2.3 Expression des contraintes E/S**

Par la suite, nous proposons d'étudier ces deux approches qui diffèrent conceptuellement par la considération ou non de la structure de la loi de commande et, par conséquent, par leurs méthodes de résolution.

### **2.3.2.1. Etude de la faisabilité des spécifications temporelles par une approche E/S**

Dans une approche purement d'analyse, le problème de faisabilité (ou de compatibilité des contraintes), en monovariante, revient à prouver l'existence ou la non existence d'un couple entrée/sortie  $(u(t), y(t))$  vérifiant les gabarits (cf. figure 2.3). La structure de correcteur n'étant pas prise en compte, ce problème est défini à base de contraintes concernant les trajectoires E/S du système à corriger seulement. Cette vision trajectoire se distingue de la synthèse du fait qu'elle ne fait pas intervenir de structure de correction mais traite seulement du comportement intrinsèque du système considéré.

Ce concept est similaire au problème de planification de trajectoires, souvent considéré dans des approches de commande pour les systèmes non linéaires (platitude, commande prédictive,...). Cette dernière consiste également à chercher des couples de signaux E/S vérifiant certaines conditions. Les trajectoires d'entrée calculées servent donc de consignes et sont utilisées dans une structure de suivi de trajectoire.

#### **Problème de faisabilité pour un système monovariante**

A chaque correcteur, solution du problème, correspondent des trajectoires E/S vérifiant le problème de faisabilité suivant :

Existe-t-il  $(u(t), y(t))$  tel que :

$$\begin{cases} u(t) \in C_u, & C_s = \{u(t) \mid u^{\min}(t) \leq u(t) \leq u^{\max}(t), \forall t\} \\ y(t) \in C_y, & C_y = \{y(t) \mid y^{\min}(t) \leq y(t) \leq y^{\max}(t), \forall t\} \\ y(t) = G(u(t)) \end{cases} \quad (2-6)$$

Nous rappelons que  $G$  correspond, dans le cas général, à l'opérateur temporel associé au système. Dans le cas linéaire, cet opérateur représente la convolution du signal par la réponse impulsionnelle du transfert  $G$ .

Le problème (2-6) peut être énoncé d'une façon plus compacte en le paramétrant uniquement en fonction de l'entrée  $u(t)$ . A chaque signal  $u(t)$ , nous associons le couple de trajectoire  $(u(t), G(u(t)))$  et le nouveau problème de faisabilité s'écrit :

Existe-t-il  $u(t)$  tel que :

$$\begin{cases} u(t) \in C_u \\ G(u(t)) \in C_y \end{cases} \quad (2-7)$$

Notons que ce problème peut aussi s'exprimer à base de considérations ensemblistes. En effet, ce problème est faisable si et seulement si l'intersection des signaux appartenant à  $C_y$  avec l'image par  $G$  des signaux d'entrée appartenant à  $C_u$  est non vide :  $G(C_u) \cap C_y \neq \emptyset$ .

Ce problème de faisabilité peut aussi s'écrire :

Existe-t-il  $(u(t), \gamma)$  tel que :

$$\begin{cases} u^{\min}(t) - \gamma \leq u(t) \leq u^{\max}(t) + \gamma \\ y^{\min}(t) - \gamma \leq G(u(t)) \leq y^{\max}(t) + \gamma \\ \gamma \leq 0 \end{cases} \quad (2-8)$$

où  $\gamma$  est une variable réelle additive supplémentaire. En choisissant comme variables de décision le vecteur  $(u(t), \gamma)$ , nous définissons le problème d'optimisation suivant :

$$\begin{aligned} & \min \gamma \\ & \begin{cases} u^{\min}(t) - \gamma \leq u(t) \leq u^{\max}(t) + \gamma \\ y^{\min}(t) - \gamma \leq G(u(t)) \leq y^{\max}(t) + \gamma \end{cases} \end{aligned} \quad (2-9)$$

Le problème de faisabilité (2-6) est alors équivalent à la condition suivante :

$$\gamma_{opt} \leq 0 \quad (2-10)$$

où  $\gamma_{opt}$  est le minimum global du problème d'optimisation (2-9).

Une autre mise en forme du problème de faisabilité (2-6) est possible en utilisant cette fois une variable  $\delta$  positive au lieu de la variable additive  $\gamma$  :

$$\begin{aligned} & \min \delta \\ & \left\{ \begin{array}{l} \delta(u^{\min}(t) - \bar{u}(t)) \leq u(t) - \bar{u}(t) \leq \delta(u^{\max}(t) - \bar{u}(t)) \\ \delta(y^{\min}(t) - \bar{y}(t)) \leq G(u(t)) - \bar{y}(t) \leq \delta(y^{\max}(t) - \bar{y}(t)) \\ \bar{u}(t) = (u^{\max}(t) + u^{\min}(t))/2, \quad \bar{y}(t) = (y^{\max}(t) + y^{\min}(t))/2 \\ \delta > 0 \end{array} \right. \end{aligned} \quad (2-11)$$

Dans ce cas, les variables d'optimisation sont  $(u(t), \delta)$  et le problème (2-6) est faisable si et seulement si :

$$0 < \delta_{opt} \leq 1 \quad (2-12)$$

Les deux formulations, énoncées ci-dessus, sont envisageables et leurs problèmes d'optimisation présentent les mêmes propriétés. Dans la suite de cette section, on se contentera d'exposer une des deux formulations.

Dans le cas où le cahier des charges comporte plusieurs contraintes sur les trajectoires E/S impliquant, par exemple, différents signaux de consigne (des spécifications pour une réponse à un échelon, à une rampe, à une sinusoïde...) chaque consigne pourra être associée à une condition nécessaire faisant intervenir un problème élémentaire de faisabilité de type (2-6). Le problème de faisabilité E/S global s'écrit sous la forme :

$$\begin{aligned} & \text{Existe-t-il } (u_i(t))_{i=1, \dots, q} \text{ tel que :} \\ & \left\{ \begin{array}{l} u_i(t) \in C_{u_i}, \quad C_{u_i} = \{u_i(t) \mid u_i^{\min}(t) \leq u_i(t) \leq u_i^{\max}(t), \forall t\} \\ G(u_i(t)) \in C_{y_i}, \quad C_{y_i} = \{y_i(t) \mid y_i^{\min}(t) \leq y_i(t) \leq y_i^{\max}(t), \forall t\} \end{array} \right\}_{i=1, \dots, q} \end{aligned} \quad (2-13)$$

Ici, il ne s'agit pas de trouver un même couple permettant de satisfaire toutes les contraintes élémentaires mais d'analyser pour chacune d'entre elles, s'il existe des couples de trajectoires  $(u_i(t), y_i(t))$  la vérifiant. Le problème de commande global consiste à chercher un correcteur unique permettant de générer pour chaque contrainte une trajectoire de commande  $u_i(t)$  correspondant à une sortie satisfaisante  $y_i(t)$ . Le problème d'existence d'un tel correcteur sera abordé dans la suite de ce paragraphe.

Comme dans le cas d'une contrainte E/S, le problème de faisabilité (2-13) est exprimé par le problème d'optimisation suivant :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} (u_i^{\min}(t) - \gamma) \leq u_i(t) \leq (u_i^{\max}(t) + \gamma) \\ (y_i^{\min}(t) - \gamma) \leq G(u_i(t)) \leq (y_i^{\max}(t) + \gamma) \end{array} \right\}_{i=1, \dots, q} \end{aligned} \quad (2-14)$$

où les variables de décisions sont  $(u_1(t), \dots, u_q(t), \gamma)$ . De même, le problème sera faisable si et seulement si  $\gamma_{opt} \leq 0$ .

### **Problème de faisabilité pour un système multivariable**

Pareillement, le problème de faisabilité (2-6) peut être facilement étendu pour le cas de plusieurs contraintes E/S pour un système multivariable à  $n$  entrées et  $m$  sorties (cf. figure 2.4) :



Existe-t-il  $(u_{1,l}(t), \dots, u_{n,l}(t))_{l=1, \dots, q}$  tel que :

$$\left. \begin{array}{l} \left. \begin{array}{l} u_{1,l}(t) \in C_{u_{1,l}}, \quad C_{u_{1,l}} = \{u_{1,l}(t) \mid u_{1,l}^{\min}(t) \leq u_{1,l}(t) \leq u_{1,l}^{\max}(t), \forall t\} \\ \vdots \\ u_{n,l}(t) \in C_{u_{n,l}}, \quad C_{u_{n,l}} = \{u_{n,l}(t) \mid u_{n,l}^{\min}(t) \leq u_{n,l}(t) \leq u_{n,l}^{\max}(t), \forall t\} \end{array} \right\} \\ \left. \begin{array}{l} G_l(u_{1,l}(t), \dots, u_{n,l}(t)) \in C_{y_{1,l}}, \quad C_{y_{1,l}} = \{y_{1,l}(t) \mid y_{1,l}^{\min}(t) \leq y_{1,l}(t) \leq y_{1,l}^{\max}(t), \forall t\} \\ \vdots \\ G_m(u_{1,l}(t), \dots, u_{n,l}(t)) \in C_{y_{m,l}}, \quad C_{y_{m,l}} = \{y_{m,l}(t) \mid y_{m,l}^{\min}(t) \leq y_{m,l}(t) \leq y_{m,l}^{\max}(t), \forall t\} \end{array} \right\} \end{array} \right\}_{l=1, \dots, q} \quad (2-15)$$

où  $(u_{j,l}^{\min}(t), u_{j,l}^{\max}(t))$  et  $(y_{k,l}^{\min}(t), y_{k,l}^{\max}(t))$  désignent, respectivement, les encadrements correspondant à la  $l^{\text{ième}}$  contrainte pour la  $j^{\text{ième}}$  entrée et pour  $k^{\text{ième}}$  sortie du système  $G$ .

Comme dans la formulation monovariante, le problème d'optimisation équivalent sera :

$$\min \gamma \quad (2-16)$$

$$\left\{ \begin{array}{l} (u_{j,l}^{\min}(t) - \gamma) \leq u_{j,l}(t) \leq (u_{j,l}^{\max}(t) + \gamma) \quad \text{pour } j = 1, \dots, n \\ (y_{k,l}^{\min}(t) - \gamma) \leq G_k(u_{1,l}(t), \dots, u_{n,l}(t)) \leq (y_{k,l}^{\max}(t) + \gamma) \quad \text{pour } k = 1, \dots, m \end{array} \right\}_{l=1, \dots, q}$$

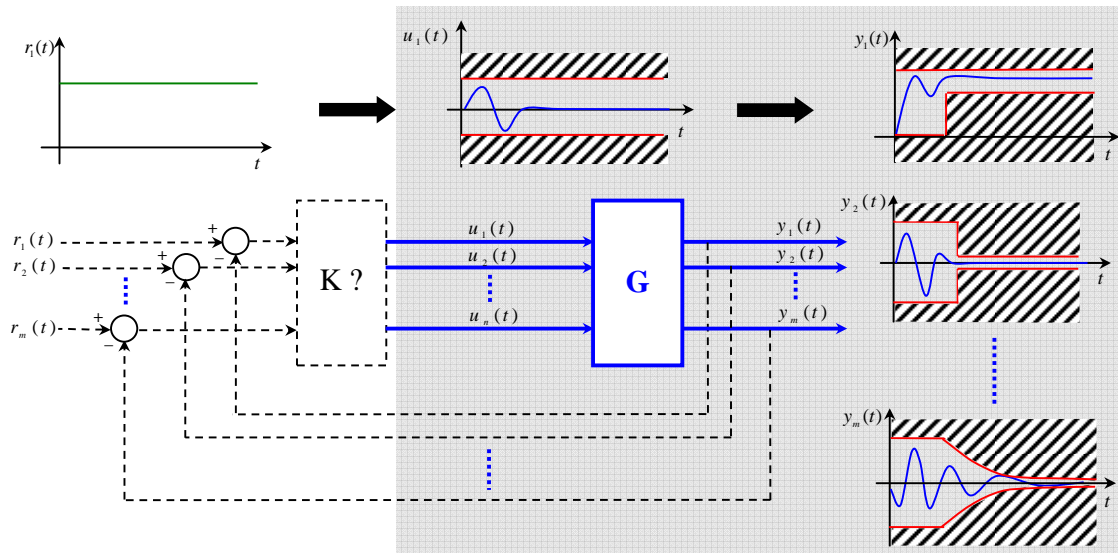


Fig. 2.4 Contraintes E/S d'un système multivariable

Des hypothèses peu restrictives permettent de garantir la faisabilité du problème (2-16). Par exemple :

- $G(0, \dots, 0) = (0, \dots, 0)$  ;
- Les gabarits inférieurs et supérieurs admettent, respectivement, un majorant fini commun  $M_1$  et un minorant fini commun  $M_2$ .

Sous ces hypothèses, le choix  $\gamma > (|M_1| + |M_2|)$  permet de garantir que les trajectoires nulles sont comprises dans les enveloppes relâchées de  $\gamma$ . Le problème correspondant aux spécifications initiales est faisable si et seulement si le minimum atteint  $\gamma_{opt}$  est négatif ou nul.

### Propriétés du problème d'optimisation formulé

Dans le cas où le système à commander  $G$  est linéaire, le problème d'optimisation formulé (2-16) est convexe. En effet, le critère de ce problème d'optimisation est linéaire par rapport à l'ensemble de variables de décision et donc il est convexe :

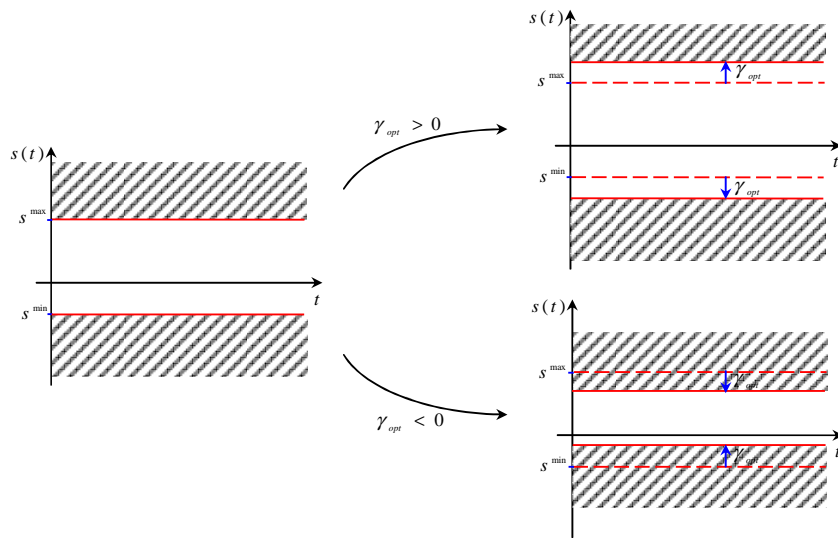
$$\gamma = (0, \dots, 0, 1) \left( (u_{1,i}(t), \dots, u_{n,i}(t))_{i=1, \dots, q}, \gamma \right)^T \quad (2-17)$$

En outre, dans le cas où le système  $G$  est linéaire, les contraintes du problème (2-16) sont également convexes par rapport au vecteur de variables d'optimisation  $((u_{1,i}(t), \dots, u_{n,i}(t))_{i=1, \dots, q}, \gamma)$ .

On rappelle qu'il est possible d'exprimer le problème (2-15) sous la forme d'un autre problème d'optimisation en introduisant une variable supplémentaire multiplicative  $\delta$  pour exprimer ses contraintes comme pour le problème d'optimisation (2-11). Le problème résultant aura comme variables  $((u_{1,i}(t), \dots, u_{n,i}(t))_{i=1, \dots, q}, \delta)$  et sera aussi convexe.

Cette propriété a des conséquences importantes sur la résolution numérique des problèmes d'optimisation. En effet, la convexité du problème implique que toute solution du problème (2-16) est un optimum global. Ainsi, le risque d'obtention des optimums locaux, qui ne permettent pas de juger la non faisabilité du problème initial, est éliminé.

Par conséquent, l'étude des trajectoires E/S permet d'aller plus loin que l'analyse de faisabilité. Elle permet aussi de chiffrer les limites de performances et aide à analyser les compromis à faire afin d'établir le meilleur cahier des charges.



**Fig. 2.5 Relâchement et renforcement des enveloppes temporelles**

Dans le cas où le minimum  $\gamma_{opt}$  est négatif, le problème est jugé faisable et les gabarits peuvent être durcis de  $\gamma_{opt}$ . Dans le cas contraire où ce paramètre est positif, les contraintes étudiées ne sont pas atteignables, l'analyse des trajectoires E/S obtenues permettra de déduire quelles sont celles qu'il faudrait relâcher (cf. figure 2.5). Ainsi, indépendamment des autres spécifications du problème, une contrainte élémentaire peut être formulée en un problème de faisabilité. Le résultat de l'optimisation nous renseigne sur les limites de performances que peut atteindre le système vis-à-vis de cette contrainte seule.

Nous pouvons appliquer ce même raisonnement pour évaluer les limites de performance d'un système avec des problèmes d'optimisation de type (2-11). Pour cela, les conditions de renforcement et de relâchement seront en fonction de la condition nécessaire et suffisante de faisabilité (2-12).

Dans le cas où certaines contraintes se sont révélées être dures, il est possible d'étudier la limite de performance que peut atteindre notre système vis-à-vis d'un critère donné sur les signaux E/S. Par exemple, on peut minimiser un critère de type  $\ell^1$ ,  $\ell^2$  ou  $\ell^\infty$  sur la commande sous une contrainte de temps de réponse. Étant donné que toute norme est une fonction convexe de son argument (conséquence de l'inégalité triangulaire), le problème formulé reste convexe.

Notons tout de même, que certaines contraintes ne peuvent être relâchées si elles sont considérées comme des contraintes dures dans le cahier des charges initial (exemple : un dépassement interdit dans un problème de régulation).

### ***Relation entre trajectoires de commande et correcteur***

Dans ce paragraphe, nous nous intéressons à montrer que dans le cas où le cahier des charges ne contient que des spécifications exprimées en E/S du système, l'équivalence entre existence de trajectoires de commande et existence d'un correcteur est vérifiée sous certaines conditions supplémentaires imposées au problème E/S.

Dans le cas où l'analyse des trajectoires E/S du système monovariante  $G$  est démontrée faisable ( $\gamma_{opt} \leq 0$ ), un ensemble de trajectoires satisfaisant les enveloppes associées est déterminé. Connaissant un couple de ces trajectoires, nous pouvons définir la trajectoire à l'entrée d'un correcteur définie par  $\varepsilon(t) = r(t) - y(t)$  et à sortie  $u(t)$ .

Dans le cas linéaire, le correcteur  $K$  est défini par l'opérateur qui à  $\varepsilon(t)$  associe  $u(t)$ . Si les signaux sont bornés et causaux, ils admettent donc une transformée de Laplace et  $K(p)$  est défini par :  $K(p) = u(p)/\varepsilon(p)$ .

Cette définition n'implique pas pour autant l'existence d'une boucle fermée bien posée. Cette propriété n'est vérifiée que si  $r(p) \neq 0$  et le correcteur  $K$  est défini de façon unique par la relation :

$$u(p) = K(p)[1 + G(p)K(p)]^{-1} r(p) \quad (2-18)$$

Pour un système monovariante et dans le cas où le cahier des charges contient deux contraintes E/S faisables (pour deux types de consigne  $r$  et  $r'$ ), la relation (2-18) permet d'éliminer le terme  $K(p)$  et d'obtenir la condition :

$$u(p)r'(p) = u'(p)r(p) \quad (2-19)$$

où  $u$  et  $u'$  sont les signaux de commande faisables associés aux consignes  $r$  et  $r'$  respectivement.

Cette condition se traduit dans le domaine temporel par l'égalité des produits de convolution suivante :

$$u(t)*r'(t) = u'(t)*r(t) \quad (2-20)$$

Ce résultat est généralisé, pour un système linéaire monovariante soumis à  $q$  contraintes E/S, par les conditions ;

$$u_i(t) * r_i(t) = u_{i'}(t) * r_{i'}(t), \quad \forall (i, i') \in \{1, \dots, q\}^2 \text{ et } i \neq i' \quad (2-21)$$

où  $r_i(t)$  et  $u_i(t)$  sont, respectivement, une entrée de consigne du système bouclé correspondant à la  $i^{\text{ème}}$  contrainte imposée à l'entrée de  $G$  et sa trajectoire de commande associée.

Ces relations d'intersection peuvent se réduire en  $(q-1)$  contraintes seulement, par exemple :

$$\{u_i(t) * r_{i+1}(t) = u_{i+1}(t) * r_i(t)\}_{i=1, \dots, q-1} \quad (2-22)$$

Ces contraintes s'ajoutent au problème (2-14) pour former le problème de faisabilité complet :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} (u_i^{\min}(t) - \gamma) \leq u_i(t) \leq (u_i^{\max}(t) + \gamma) \\ (y_i^{\min}(t) - \gamma) \leq G(u_i(t)) \leq (y_i^{\max}(t) + \gamma) \end{array} \right\}_{i=1, \dots, q} \\ & \{u_i(t) * r_{i+1}(t) = u_{i+1}(t) * r_i(t)\}_{i=1, \dots, q-1} \end{aligned} \quad (2-23)$$

Afin de simplifier ce problème d'optimisation convexe, S. Hbaïeb et S. Font [Hba02a] ont proposé une paramétrisation des trajectoires de commande  $u_i(t)$  par l'ajout d'une contrainte fictive correspondant à une consigne impulsionnelle  $r_0(t) = \delta(t)$  et une commande associée  $u_0(t)$ .

Comme pour les contraintes (2-22), nous choisissons, cette fois, les  $q$  relations suivantes :

$$\{u_i(t) * r_0(t) = u_0(t) * r_i(t)\}_{i=1, \dots, q} \quad (2-24)$$

Dans le cas présent,  $r_0(t) = \delta(t)$ , donc :

$$\{u_i(t) * r_0(t) = u_i(t) = u_0(t) * r_i(t)\}_{i=1, \dots, q} \quad (2-25)$$

Le problème (2-23) est donc équivalent au problème d'optimisation convexe :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} (u_i^{\min}(t) - \gamma) \leq u_0(t) * r_i(t) \leq (u_i^{\max}(t) + \gamma) \\ (y_i^{\min}(t) - \gamma) \leq G(u_0(t) * r_i(t)) \leq (y_i^{\max}(t) + \gamma) \end{array} \right\}_{i=1, \dots, q} \end{aligned} \quad (2-26)$$

où les variables d'optimisation sont  $(u_0(t), \gamma)$ .

Nous notons que le problème d'optimisation conserve sa caractéristique de convexité car l'opérateur produit de convolution est linéaire.

A une solution donnée, on associe les trajectoires de commande définies par :  $\{u_i(t) = u_0(t) * r_i(t)\}_{i=1, \dots, q}$ . Le correcteur permettant de les générer est défini par :

$$K(p) = u_0(p) [1 - G(p)u_0(p)]^{-1} \quad (2-27)$$

L'introduction de cette consigne fictive peut être interprétée comme une paramétrisation de l'ensemble des trajectoires de commande par une nouvelle variable  $u_0(t)$ .

L'extension de ce résultat de paramétrisation des trajectoires de commande pour le cas linéaire multivariable à plusieurs contraintes se fait, en gardant les mêmes notations, comme suit :

Dans un premier temps, nous supposons qu'il existe un ensemble de commande  $u(t) = (u_1(t), \dots, u_n(t))$  non vide, permettant d'obtenir des sorties  $y(t) = (y_1(t), \dots, y_m(t))$  appartenant à des gabarits donnés pour un seul jeu de références  $r(t) = (r_1(t), \dots, r_m(t))$ . La problématique consiste à étudier l'existence d'un correcteur  $K$  qui permet de générer le vecteur de commande  $u(t)$  selon le schéma de la figure 2.4.

Dans le cas linéaire,  $K$  est solution du problème de commande si et seulement si :

$$u(p) = K(p)(r(p) - y(p)) \quad (2-28)$$

Ce qui s'écrit également :

$$u_j(p) = \sum_{k=1}^m K_{j,k}(p)(r_k(p) - y_k(p)), \quad \forall j \in \{1, \dots, n\} \quad (2-29)$$

Si le système est de rang maximal, nous avons donc  $n$  équations à  $(n \times m)$  inconnues :

- dans le cas où  $m=1$  et où tous les couples E/S sont contraints, le système admet une solution ;
- dans tous les autres cas, il admet une infinité de solutions.

En outre, si le problème d'optimisation n'est pas correctement posé d'un point de vue physique, il est possible de trouver des trajectoires solutions qui ne donnent pas un système de rang maximal. Dans ce cas aussi, on obtient une infinité de solutions.

Dans le cas de deux contraintes E/S faisables correspondant à deux jeux de références  $r(t) = (r_1(t), \dots, r_m(t))$  et  $r'(t) = (r'_1(t), \dots, r'_m(t))$ , il s'agit d'étudier l'existence d'un correcteur  $K$  qui permet de générer un vecteur de commande  $u(t)$  (respectivement  $u'(t)$ ) appartenant à  $C_u$  (respectivement  $C_{u'}$ ) pour la consigne  $r(t)$  (respectivement  $r'(t)$ ) selon le schéma de régulation de la figure 2.4.

Dans le cas linéaire, un tel correcteur existe si et seulement s'il vérifie :

$$\begin{cases} u(p) = K(p)[I + G(p)K(p)]^{-1} r(p) \\ u'(p) = K(p)[I + G(p)K(p)]^{-1} r'(p) \end{cases} \quad (2-30)$$

Ces conditions impliquent une dépendance entre  $u$ ,  $u'$ ,  $r$  et  $r'$ . Ici aussi, la faisabilité des contraintes de trajectoires n'est qu'une condition suffisante à l'existence d'un correcteur permettant de répondre au problème initial.

Contrairement au cas monovariable, l'obtention d'une relation entre  $u$ ,  $u'$ ,  $r$  et  $r'$  indépendamment du correcteur, n'est pas immédiate. Pour l'exprimer, nous allons utiliser une paramétrisation similaire à celle appliquée dans le cas monovariable.

Pour cela, nous allons introduire  $m$  vecteurs de références fictives  $\{r_{0,k}(t)\}_{k=1, \dots, m}$  où toutes les composantes de  $r_{0,k}(t)$  sont nulles sauf la  $k^{\text{ième}}$  qui correspond à une impulsion de Dirac. Pour chaque  $r_{0,k}(t)$ , nous associons le vecteur de commande  $u_{0,k}(t)$ . Ces trajectoires sont générées par un même correcteur si et seulement si :

$$u_{0,k}(p) = K(p)[I + G(p)K(p)]^{-1} r_{0,k}(p), \quad \forall k \in \{1, \dots, m\} \quad (2-31)$$

Ce qui s'écrit aussi :

$$u_0(p) = K(p)[I + G(p)K(p)]^{-1} r_0(p) \quad (2-32)$$

où  $u_0 = (u_{0,1}, \dots, u_{0,m})$  et  $r_0 = (r_{0,1}, \dots, r_{0,m})$  ;  $r_0$  est la matrice identité.

Toute contrainte du problème de commande peut se formuler en fonction des  $u_{0,k}(t)$  car  $\{r_{0,k}(t)\}_{k=1, \dots, m}$  est une base de l'ensemble des références. Par conséquent, le problème de commande soumis à plusieurs contraintes, peut se formuler en un problème d'optimisation fonction des  $u_{0,j}(t)$ . Si une solution existe, les trajectoires de commande correspondant aux différentes contraintes s'obtiennent par combinaison linéaire des  $u_{0,k}(t)$  et le correcteur permettant de les générer est défini par la relation :

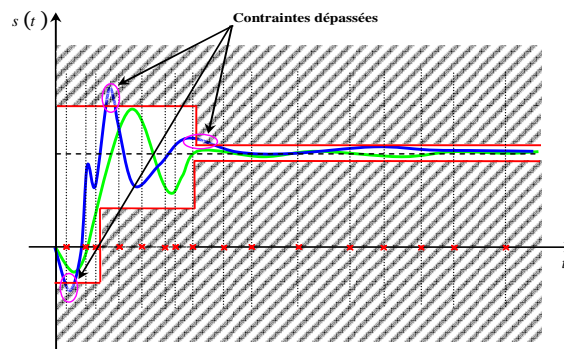
$$K(p) = u_0(p)[I_p - G(p)u_0(p)]^{-1} \quad (2-33)$$

**Résolution effective des problèmes d'optimisation temporels formulés**

Dans une approche d'analyse, la mise en pratique des problèmes d'optimisation convexe soulève plusieurs questions concernant la dimension infinie des variables de décision, l'évaluation numérique des contraintes et la complexité calculatoire du problème d'optimisation ainsi engendré.

En effet, les problèmes d'optimisation (2-9), (2-11), (2-14) et (2-16) ont pour variables de décision une ou plusieurs trajectoires appartenant donc à un espace de dimension infinie. La résolution de ce type de problème sous cette forme est irréalisable dans le cas général. Pour résoudre cette problématique, des approximations par des problèmes de dimension finie sont nécessaires. L'approximation des trajectoires entrée/sortie de la boucle par des fonctions affines par morceaux sous certaines hypothèses est envisagée.

En outre, les contraintes temporelles établies s'appliquent à des trajectoires E/S du système et leurs évaluations effectives nécessitent soit l'intégration des équations différentielles de la boucle, soit le calcul des produits de convolution de signaux dans le cas linéaire. Ces deux opérations utilisent des intégrations de fonctions et reposent numériquement sur des méthodes d'approximation et de résolution d'équations différentielles et ne permettent ainsi d'obtenir qu'un nombre fini de points estimant les trajectoires E/S du système (cf. figure 2.6).



**Fig. 2.6 Discrétisation des contraintes de gabarit temporel**

On écrit alors :

$$\hat{s}(t_i) = s(t_i) + b_s(t_i), \quad i = 1, \dots, n_f \quad (2-34)$$

où  $\hat{s}(t_i)$  et  $b_s(t_i)$  sont, respectivement, l'estimée du signal  $s(t)$  et son erreur d'estimation due au processus d'intégration à l'instant  $t_i$ .  $n_f$  est la taille de ces vecteurs.

Un exemple d'approximation de ces problèmes d'optimisation est donné par la formulation finie du problème (2-16) :

$$\min \gamma \quad (2-35)$$

$$\left\{ \begin{array}{l} (u_{j,l}^{\min}(t_i) - \gamma) \leq u_{j,l}(t_i) \leq (u_{j,l}^{\max}(t_i) + \gamma) \quad \text{pour } j = 1, \dots, n \\ (y_{k,l}^{\min}(t_i) - \gamma) \leq G_k(u_{1,l}(t_i), \dots, u_{n,l}(t_i)) \leq (y_{k,l}^{\max}(t_i) + \gamma) \quad \text{pour } k = 1, \dots, m \end{array} \right\}_{\substack{l=1, \dots, q \\ i=1, \dots, n_f}}$$

Il serait donc intéressant d'étudier la relation entre la faisabilité des contraintes initiales et celles des contraintes discrétisées (à base des trajectoires estimées).

L'étude qui a été menée dans les travaux de S. Hbaïeb [Hba02b] montre qu'il est possible de choisir une base finie qui permet de réduire la complexité calculatoire des problèmes d'optimisation formulés tout en conservant leur caractère convexe. En effet, sous l'hypothèse de trajectoire entrée/sortie  $s(t)$  à temps de réponse fini et à variations bornées appartenant à l'ensemble  $V_{\alpha-T_r}$  défini par :

$$V_{\alpha-T_r} = \left\{ s(t) \mid \left( \begin{array}{l} \forall t \in ]-\infty, 0] \cup ]T_r, +\infty[, \quad s(t) = c \quad (c \in \mathfrak{R}) \\ \forall t \in ]0, T_r], \quad t_1 \neq t_2, \quad \frac{s(t_1) - s(t_2)}{t_1 - t_2} \leq \alpha \end{array} \right) \right\} \quad (2-36)$$

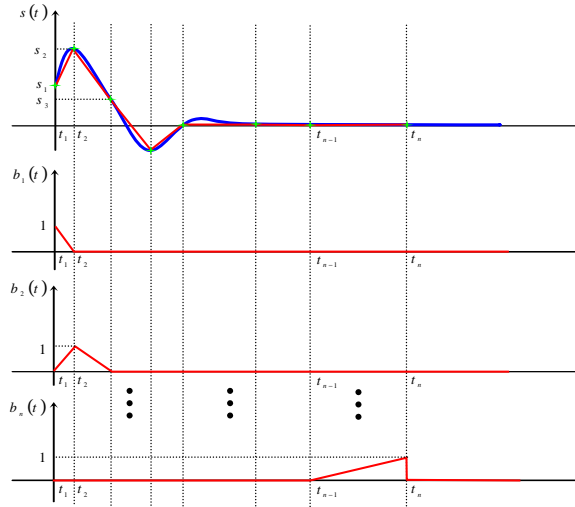
il existe pour toute précision  $\varepsilon$  donnée une discrétisation temporelle et donc une base finie de fonctions affines par morceaux  $\{b_i(t)\}_{i=1, \dots, n_f}$  telle que :

$$\forall s(t) \in V_{\alpha-T_r}, \quad \exists \{\alpha_i(t)\}_{i=1, \dots, n_f} \in \mathfrak{R}^{n_f} \quad \text{tel que} \quad \forall t > 0 \quad \left| s(t) - \sum_{i=1}^{n_f} \alpha_i b_i(t) \right| \leq \varepsilon \quad (2-37)$$

La base fonctionnelle  $\{b_i(t)\}_{i=1, \dots, n_f}$  est définie par :

$$\left\{ \begin{array}{l} b_1(t) = \begin{cases} 0 & \forall t \in ]-\infty, t_1[ \cup ]t_2, +\infty[ \\ \frac{t-t_2}{t_1-t_2} & \forall t \in [t_1, t_2] \end{cases} \\ b_{i=2, \dots, n_f-1} = \begin{cases} 0 & \forall t \in ]-\infty, t_{i-1}[ \cup ]t_{i+1}, +\infty[ \\ \frac{t-t_{i-1}}{t_i-t_{i-1}} & \forall t \in [t_{i-1}, t_i] \\ \frac{t-t_{i+1}}{t_i-t_{i+1}} & \forall t \in [t_i, t_{i+1}] \end{cases} \\ b_{n_f} = \begin{cases} 0 & \forall t \in ]-\infty, t_{n-1}[ \cup ]t_n, +\infty[ \\ \frac{t-t_{n-1}}{t_n-t_{n-1}} & \forall t \in [t_{n-1}, t_n] \end{cases} \end{array} \right. \quad (2-38)$$

Ce qui correspond au schéma suivant :



**Fig. 2.7** Approximation affine par morceau de  $s(t)$

Cette base fonctionnelle garantit un écart, entre les gabarits continus et les gabarits affines par morceau, inférieur à  $\varepsilon$ , ce qui s'écrit :

$$\forall \varepsilon > 0, \quad \exists t_1 < \dots < t_{n_f} \quad \text{tel que} \quad (2-39)$$

$$\left( \forall t_i \in \{t_1, \dots, t_{n_f}\} \right) \Rightarrow \left( \forall t \in \mathfrak{R} \right. \\ \left. s^{\min}(t_i) \leq s(t) \leq s^{\max}(t_i) \right) \Rightarrow \left( \forall t \in \mathfrak{R} \right. \\ \left. s^{\min}(t) - \varepsilon \leq s(t) \leq s^{\max}(t) + \varepsilon \right)$$

En utilisant cette approximation affine par morceau pour chaque trajectoire E/S, les problèmes continus (2-9), (2-11), (2-14) et (2-16) restent convexes par rapport aux nouveaux paramètres d'optimisation  $\alpha_i$  (coefficient de projection sur la base  $\{b_i(t)\}_{i=1, \dots, n_f}$ ).

Le caractère affine des critères et des contraintes des problèmes, susmentionnés, fait des techniques de programmation convexe (linéaire, quadratique et LMI) le meilleur moyen à explorer pour les résoudre.

Dans cette résolution des problèmes d'optimisation, les contraintes doivent être évaluées à chaque itération de l'optimisation. De ce fait, il est clair que la complexité du problème est fortement corrélée avec le nombre de contraintes imposées, le nombre de points de discrétisation choisi et le nombre de variables d'optimisation. La résolution effective de ces problèmes se confronte donc à des limitations en termes de temps de calcul et de place mémoire nécessaires.

En pratique, il est tout de même possible de choisir une discrétisation temporelle sans aucune garantie a priori, puis de les affiner après analyse des solutions trouvées. Ainsi pour un bon choix donné de discrétisation, si le minimum du problème d'optimisation défini par (2-35) est positif, alors nous concluons à la non faisabilité des contraintes initiales. Dans le cas où le minimum trouvé est négatif, la visualisation des trajectoires correspondant à la solution trouvée permet d'observer si les contraintes continues sont vérifiées. Si cette condition est vérifiée, le problème initial est faisable, sinon il faut affiner la discrétisation et itérer la même procédure de nouveau.



**2.3.2.2. Etude de la faisabilité des spécifications temporelles par une approche paramétrique générale**

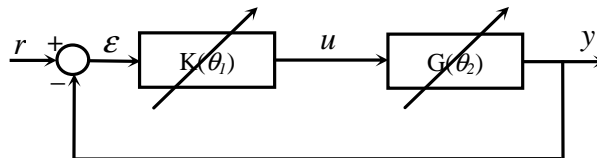
Nous présentons, dans cette section, l'étude de la faisabilité d'un cahier des charges via une approche de synthèse. Il s'agit d'explorer l'espace paramétrique d'une structure de commande donnée afin de décider de la faisabilité ou non d'un cahier des charges. Par rapport à l'approche d'analyse, cette formulation présente l'avantage de pouvoir considérer des cahiers des charges plus larges comprenant des spécifications liées à la structure de la loi de commande (souvent intéressant pour des applications industrielle et conséquence de contraintes liées à l'implantation). De plus, le traitement est identique que les opérateurs considérés soient linéaires ou non linéaires.

Quand les objectifs d'un cahier des charges sont modifiés, l'approche paramétrique peut s'avérer comme un bon choix. En effet, dans une telle situation, recommencer les différentes étapes de synthèse depuis le début n'est pas nécessairement le plus efficace. L'optimisation d'une structure paramétrique permet de retoucher les paramètres de l'ancienne loi de commande afin qu'elle garantisse les nouvelles spécifications. La méthode proposée peut alors être vue comme processus itératif de synthèse multi-objectif.

Le principe de cette formalisation est décrit via le schéma bloc de commande classique déjà introduit dans le premier chapitre (cf. figure 2.8).

**Dépendance implicite des trajectoires de la boucle de commande aux paramètres de décision**

Nous considérons une approche paramétrique globale où les paramètres du correcteur  $K$  sont désignés par un vecteur  $\theta_1$ . Le modèle du système à commander peut également dépendre de paramètres ajustables  $\theta_2$ . L'ensemble de ces paramètres constitue le vecteur de variables de décision  $\theta$ .



**Fig. 2.8 Schéma de la boucle fermée et des variables de décision**

Les grandeurs intervenant dans les contraintes du domaine temporel sont, en majorité, de nature implicite. En effet, en réponse à une sollicitation déterminée, chaque signal  $s(t)$  de la boucle fermée s'exprime comme une fonction implicite des paramètres de réglage  $\theta_1$  et  $\theta_2$ . Nous écrivons alors :

$$s(K(\theta_1), G(\theta_2), r, t) = s(\theta, t) \tag{2-40}$$

Dans le cas des systèmes linéaires, le signal  $s(t)$  se définit comme étant le produit de convolution de la réponse impulsionnelle du transfert  $T_{r \rightarrow s}$  et du signal de référence  $r(t)$  :  $s(\theta, t) = T_{r \rightarrow s}(\theta, t) * r(t)$ . Le calcul du signal  $s(\theta, t)$  est donc sujet à une intégration de fonction souvent évaluée numériquement à base de méthodes d'approximation ou de résolutions d'équations différentielles.

De même, le signal  $s(\theta, t)$  est défini, dans le domaine fréquentiel, via sa transformée de Laplace inverse par :  $s(\theta, t) = \mathcal{L}^{-1}[T_{r \rightarrow s}(\theta, p)r(p)]$ . Malgré l'existence de méthodes algébriques qui permettent d'exprimer formellement (exactement) le signal  $s(\theta, t)$ , la dépendance entre ce dernier et les

paramètres de décision  $\theta$  reste implicite car ces méthodes se basent sur la détermination des pôles de la boucle fermée qui ne sont que des racines estimées d'un polynôme caractéristique dépendant du vecteur  $\theta$ . Par conséquent, chaque signal  $s(\theta, t)$  de la boucle de commande est estimé via une résolution numérique  $\hat{s}(\theta, t_i)$  de l'équation différentielle qui lui est associée. La pertinence de cette estimation dépend de la méthode numérique et du pas de discrétisation utilisés.

### *Formulation des spécifications directes*

Contrairement à l'approche d'analyse où les spécifications génériques ne peuvent être considérées qu'à travers une formulation par gabarits, l'approche de synthèse permet de les formuler explicitement et indépendamment.

Les différents indices de performance temporels sont, également, dépendants des paramètres de décision  $\theta$ . Nous les noterons :

$$\alpha_i(\varepsilon(\theta, t), t) = \alpha_i(\theta) \quad (2-41)$$

Ces indicateurs  $\alpha_i$  peuvent bien sûr être exprimés en fonction d'autres signaux via les différentes relations E/S de la boucle. Leur évaluation dépend, évidemment, du calcul de ces trajectoires intermédiaires comme ici le signal d'erreur  $\varepsilon(t)$ . Cependant, ces trajectoires qui appartiennent à un espace de dimension infinie font de l'évaluation numérique de ces indicateurs un problème impossible dans le cas général. Afin d'obvier à cette problématique, des approximations des trajectoires entrée/sortie de la boucle par des estimées de dimension finie sont nécessaires.

Ainsi, bien que ces indicateurs temporels soient bien définis mathématiquement, leur calcul est approximatif et n'est qu'une solution numérique d'un problème algébrique ou d'un problème d'optimisation unidirectionnelle.

Le problème de faisabilité concernant une spécification temporelle directe peut être formulé comme étant le problème de recherche des paramètres  $\theta$  tels que la contrainte (2-1) soit vérifiée :

Existe-t-il  $\theta$  tel que :

$$\begin{cases} \alpha_i^{\min} \leq \alpha_i(\varepsilon(\theta, t), t) \leq \alpha_i^{\max} \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))) \end{cases} \quad (2-42)$$

Pour sa résolution numérique, le problème de faisabilité (2-42) peut être traduit sous forme d'un problème d'optimisation dont l'analyse des solutions permettrait de juger de la faisabilité des contraintes directes du problème initial. D'une manière équivalente, notre problème de faisabilité peut s'écrire :

Existe-t-il  $(\theta, \gamma)$  tel que :

$$\begin{cases} \alpha_i^{\min} - \gamma \leq \alpha_i(\varepsilon(\theta, t), t) \leq \alpha_i^{\max} + \gamma \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))) \\ \gamma \leq 0 \end{cases} \quad (2-43)$$

où  $\gamma$  est une variable réelle additive supplémentaire qui permet de traduire le problème de faisabilité en un problème d'optimisation :

$$\begin{aligned} & \min \gamma \\ & \begin{cases} \alpha_i^{\min} - \gamma \leq \alpha_i(\varepsilon(\theta, t), t) \leq \alpha_i^{\max} + \gamma \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))) \end{cases} \end{aligned} \quad (2-44)$$

Le problème (2-42) est faisable si et seulement si :  $\gamma_{opt} \leq 0$ .

Comme dans le cas des contraintes sous forme d'encadrement, une deuxième mise en forme du problème de faisabilité (2-42) est possible en utilisant une variable réelle multiplicative  $\delta$  positive :

$$\begin{aligned} & \min \delta \\ & \begin{cases} \delta(\alpha_i^{\min} - \bar{\alpha}_i) \leq \alpha_i(\varepsilon(\theta, t), t) - \bar{\alpha}_i \leq \delta(\alpha_i^{\max} - \bar{\alpha}_i) \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))), \quad \bar{\alpha}_i = (\alpha_i^{\max} + \alpha_i^{\min})/2 \\ \delta > 0 \end{cases} \end{aligned} \quad (2-45)$$

Dans ce cas, les variables d'optimisation sont  $(\theta, \delta)$  et le problème (2-42) est faisable si et seulement si  $0 < \delta_{opt} \leq 1$ .

Il est clair que la formulation des problèmes d'optimisations (2-44) et (2-45) est directe (sans approximation) et si ce problème est résolu efficacement, il permettra de répondre à une partie de la problématique du cahier des charges. Toutes fois, ce problème d'optimisation est très difficile à résoudre car il est non convexe et présente des dépendances non linéaires-implicites par rapport aux paramètres de décision  $\theta$ .

### ***Formulation des spécifications temporelles d'encadrement***

Dans l'approche paramétrique, le principe de la formulation sous forme de gabarits est similaire à celui d'une approche d'analyse. Les contraintes sont définies sur tous les signaux de la boucle fermée et pour différents types de consigne. La principale différence entre les deux approches provient de la paramétrisation des problèmes d'optimisation qui ne sont plus en fonction de trajectoires mais en fonction d'un vecteur de variables de décision  $\theta$ .

Le problème de faisabilité des contraintes temporelles sous forme de gabarits peut être intégralement formulé par :

$$\begin{aligned} & \text{Existe-t-il } \theta \text{ tel que :} \\ & \begin{cases} s_{j,l}(\theta, t) \in C_{s_{j,l}}, \quad j=1, \dots, n \quad \text{et} \quad C_{s_{j,l}} = \{s_{j,l}(t) \mid s_{j,l}^{\min}(t) \leq s_{j,l}(t) \leq s_{j,l}^{\max}(t), \quad \forall t\} \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t) \end{cases} \end{aligned} \quad (2-46)$$

où  $s_{j,l}(t)$  est une trajectoire donnée de la boucle fermée sous contraintes et  $s_{j,l}^{\min}(t)$  et  $s_{j,l}^{\max}(t)$  des bornes à respecter pour cette trajectoire.  $n$  est le nombre de contraintes.  $q$  est le nombre de types d'excitations considérées.  $C_{s_{j,l}}$  est l'ensemble des signaux  $s_{j,l}(t)$  vérifiant ces contraintes, tandis que, l'opérateur  $F_{j,l}$  exprime la dépendance entre la  $j^{\text{ième}}$  trajectoire  $s_{j,l}(t)$  et le  $l^{\text{ième}}$  entrée de référence  $r_l(t)$ .

En procédant de la même manière que dans le cas des contraintes directes, le problème (2-46) est exprimé par un des problèmes d'optimisation suivants :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} s_{j,l}^{\min}(t) - \gamma \leq s_{j,l}(\theta, t) \leq s_{j,l}^{\max}(t) + \gamma \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t) \end{array} \right\}_{\substack{j=1, \dots, n \\ l=1, \dots, q}} \end{aligned} \quad (2-47)$$

$$\begin{aligned} & \min \delta \\ & \left\{ \begin{array}{l} \delta \cdot (s_{j,l}^{\min}(t) - \bar{s}_{j,l}(t)) \leq s_{j,l}(\theta, t) - \bar{s}_{j,l}(t) \leq \delta \cdot (s_{j,l}^{\max}(t) - \bar{s}_{j,l}(t)) \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t), \quad \bar{s}_{j,l}(t) = (s_{j,l}^{\max}(t) + s_{j,l}^{\min}(t))/2 \\ \delta > 0 \end{array} \right\}_{\substack{j=1, \dots, n \\ l=1, \dots, q}} \end{aligned} \quad (2-48)$$

La condition de faisabilité du problème (2-46) dépend de la solution des deux problèmes (2-47) et (2-48). Leur faisabilité est respectivement donnée par :  $\gamma_{opt} \leq 0$  et  $0 < \delta_{opt} \leq 1$ .

### **Formulation d'un cahier des charges temporel en un problème d'optimisation non linéaire**

La mise en commun des deux types de spécifications permet de généraliser notre approche en réunissant les deux problèmes de faisabilité élémentaires (2-42) et (2-46) en un seul problème global :

Existe-t-il  $\theta$  tel que :

$$\left\{ \begin{array}{l} \alpha_t^{\min} \leq \alpha_t(\varepsilon(\theta, t), t) \leq \alpha_t^{\max} \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))) \\ \left\{ \begin{array}{l} s_{j,l}(\theta, t) \in C_{s_{j,l}}, \quad j=1, \dots, n \quad \text{et} \quad C_{s_{j,l}} = \{s_{j,l}(t) \mid s_{j,l}^{\min}(t) \leq s_{j,l}(\theta, t) \leq s_{j,l}^{\max}(t), \quad \forall t\} \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t) \end{array} \right\}_{l=1, \dots, q} \end{array} \right. \quad (2-49)$$

La formulation en un problème d'optimisation de ce nouveau problème de faisabilité revient à résoudre les deux problèmes (2-44) et (2-47) simultanément :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} \alpha_t^{\min} - \gamma \leq \alpha_t(\varepsilon(\theta, t), t) \leq \alpha_t^{\max} + \gamma \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))) \\ \left\{ \begin{array}{l} s_{j,l}^{\min}(t) - \gamma \leq s_{j,l}(\theta, t) \leq s_{j,l}^{\max}(t) + \gamma \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t) \end{array} \right\}_{\substack{j=1, \dots, n \\ l=1, \dots, q}} \end{array} \right. \end{aligned} \quad (2-50)$$

Un problème d'optimisation non linéaire similaire peut être formulé en assemblant les problèmes (2-45) et (2-48) :

$$\begin{aligned} & \min \delta \\ & \left\{ \begin{array}{l} \delta \cdot (\alpha_t^{\min} - \bar{\alpha}_t) \leq \alpha_t(\varepsilon(\theta, t), t) - \bar{\alpha}_t \leq \delta \cdot (\alpha_t^{\max} - \bar{\alpha}_t) \\ \varepsilon(\theta, t) = r(t) - G(K(\varepsilon(\theta, t))), \quad \bar{\alpha}_t = (\alpha_t^{\max} + \alpha_t^{\min})/2 \\ \delta > 0 \\ \left\{ \begin{array}{l} \delta \cdot (s_{j,l}^{\min}(t) - \bar{s}_{j,l}(t)) \leq s_{j,l}(\theta, t) - \bar{s}_{j,l}(t) \leq \delta \cdot (s_{j,l}^{\max}(t) - \bar{s}_{j,l}(t)) \\ s_{j,l}(\theta, t) = F_{j,l}(\theta, r_l, t), \quad \bar{s}_{j,l}(t) = (s_{j,l}^{\max}(t) + s_{j,l}^{\min}(t))/2 \end{array} \right\}_{\substack{j=1, \dots, n \\ l=1, \dots, q}} \end{array} \right. \end{aligned} \quad (2-51)$$

Notons que des contraintes supplémentaires sur les paramètres de réglage  $\theta$  (telle que la positivité des paramètres) peuvent être ajoutées afin de réaliser des correcteurs à structure bien définie.

Les conditions de faisabilité des spécifications temporelles ne changent pas et sont toujours données par les inégalités (2-10) et (2-12).

Afin de chiffrer les limites de performances atteignables par la boucle fermée, il est possible d'utiliser un critère intégral (de norme), sur l'une des trajectoires entrées/sortie, qu'on minimisera sous les contraintes du problème de faisabilité (2-47).

### ***Propriétés du problème d'optimisation formulé***

L'ensemble des contraintes intervenant dans la formulation par synthèse dépend des variables  $\theta$  de la boucle de commande. Cette dépendance est de nature très complexe car ces relations sont en majorité implicites et non linéaires. C'est pourquoi les problèmes d'optimisation établis sont non linéaires non convexes, et le risque d'atteindre des minimums locaux ou des zones singulières, lors du processus d'optimisation, est très probable. Ceci nous empêche de juger de la non faisabilité du cahier des charges. Néanmoins, les problèmes d'optimisation non linéaire exprimés en fonction de  $\theta$  présentent un grand intérêt. Ils sont très utiles à des fins de synthèse de correcteurs structurés ou d'ordre réduit. Dans ces cas, la nature de l'optimum calculé (local ou global) peut ne pas être importante si le système corrigé présente un comportement satisfaisant.

D'autre part, les évaluations effectives des contraintes temporelles ne peuvent se faire que sur la base d'une estimée des signaux de la boucle qui résulte d'une intégration numérique des relations E/S de la boucle fermée. Ce processus d'évaluation approximatif risque d'être lent et très coûteux en espace mémoire ce qui rend le processus d'optimisation moins efficace.

Par conséquent, il est important d'analyser profondément les caractéristiques variationnelles et calculatoires des critères et contraintes constituant ce type de problèmes afin d'identifier les techniques de calcul et les algorithmes d'optimisation les plus efficaces pour les résoudre.

### ***Mise en œuvre des problèmes d'optimisation temporels formulés***

Malgré la forte similitude entre les deux approches présentées, les problèmes d'optimisation formulés présentent de grandes disparités dues principalement à l'appartenance à deux familles de problèmes d'optimisation complètement différentes : les problèmes d'optimisation convexe et les problèmes d'optimisation non linéaire.

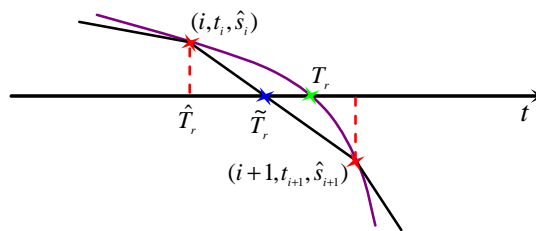
Outre les complications liées à l'évaluation des contraintes de dimension infinie, dans une approche de synthèse, on rencontre des difficultés supplémentaires liées aux définitions mathématiques des indicateurs de performance temporels.

En effet, ces indicateurs sont, en majorité, difficiles à calculer car ils nécessitent la résolution de problèmes implicites algébriques et/ou différentiels. Nous citons quelques exemples :

La "rapidité" d'un signal  $s(t)$  est évaluée par sa pente initiale, cette dernière nécessite le calcul de la dérivée de ce signal à  $t = 0$ . Comme cette évaluation ne se fait généralement qu'à partir d'une estimée finie, elle sera approchée (par exemple :  $\hat{s}(t_i) = (\hat{s}(t_{i+1}) - \hat{s}(t_i)) / (t_{i+1} - t_i)$ ) et dépendra de la qualité d'estimation (le pas de discrétisation) du signal  $\hat{s}(t_i)$ .

Le calcul du temps de montée et du temps d'établissement nécessite de la résolution d'un ou plusieurs problèmes algébriques du type :  $f(\hat{s}(t_i)) = 0$  (pour déterminer le passage d'une trajectoire par une valeur donnée ou pour en déterminer un maximum local). Ce calcul est, aussi, lié à la qualité de l'approximation du signal  $s(t)$ . Ainsi, quelle que soit la méthode utilisée pour le calcul (maillage, par dichotomie, optimisation unidirectionnelle), le résultat restera biaisé et son erreur d'estimation sera liée à la densité des échantillons du signal  $\hat{s}(t_i)$ . Ceci doit être pris en compte par l'algorithme d'optimisation.

Par exemple, la figure 2.9 représente une situation dans laquelle un temps de réponse se déduit du point d'intersection entre une trajectoire et un axe horizontal. En ne se basant que sur le vecteur temps issu de la résolution différentielle du système, le temps de réponse  $T_r$  peut être estimé par  $\hat{T}_r$  donné par l'échantillon calculé le plus proche.



**Fig. 2.9 Estimation d'un temps de réponse**

Il est clair que cette estimation est grossière, une solution plus fine peut être obtenue par une interpolation entre les coordonnées  $i$  et  $i + 1$ . Une interpolation linéaire donnera l'estimée suivante :

$$\tilde{T}_r = t_{i+1} - (\hat{s}_{i+1} - \hat{s}_k) \left( \frac{t_{i+1} - t_i}{\hat{s}_{i+1} - \hat{s}_i} \right) \quad \text{avec} \quad \hat{s}_k = (1 \mp 0.05) \cdot \hat{s}(\infty)$$

Le même type d'outils de calcul est employé pour la recherche unidirectionnelle qui permet de calculer les indicateurs dépassement et valeur maximale. Ces deux derniers sont définis implicitement comme des solutions d'un problème d'optimisation sur la variable temps qui n'est représenté, en réalité, que par un vecteur de temps fini  $\hat{t} = (t_0, \dots, t_n)$ .

Ces deux dernières difficultés (estimation d'une dérivée et résolution d'un problème algébrique à base d'une estimée) sont rencontrées simultanément lors du calcul du temps du premier maximum d'un signal donné.

Les critères de performance intégraux réaffirment l'imprécision du calcul et sont également évalués approximativement à partir des échantillons du signal.

Par conséquent, l'évaluation des contraintes de type (2-1) dépend de l'évaluation des indicateurs de performance  $\alpha_i$ . Elle nécessite souvent une résolution numérique d'un problème algébrique, qui lui-même requiert une solution estimée (un nombre fini d'instants  $t_i$ ) d'un premier problème différentiel.

Le problème de faisabilité global (2-50) ne peut en fait n'être résolu qu'en utilisant des estimations des signaux  $\hat{e}(\theta, t_i)$ :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{array}{l} \alpha_i^{\min} - \gamma \leq \alpha_i(\hat{\varepsilon}(\theta, t_i), t_i) \leq \alpha_i^{\max} + \gamma \\ \varepsilon(\theta, t_i) = r(t_i) - G(K(\hat{\varepsilon}(\theta, t_i))) \end{array} \right\} \\ & \left\{ \begin{array}{l} s_{j,l}^{\min}(t_i) - \gamma \leq s_{j,l}(\theta, t_i) \leq s_{j,l}^{\max}(t_i) + \gamma \\ s_{j,l}(\theta, t_i) = F_{j,l}(\theta, r_l, t_i) \end{array} \right\}_{\substack{j=1,\dots,n \\ l=1,\dots,q}} \end{aligned} \quad (2-52)$$

D'autre part, les techniques de calcul de ce type de contraintes sont très coûteuses en temps de calcul et elles ne fournissent qu'une estimée biaisée de la vraie solution. En effet, cette estimée n'est pas crédible pour décrire les variations paramétriques des différentes variables dans un processus d'optimisation. En plus, les algorithmes d'optimisation sont souvent très sensibles à la fiabilité de l'information requise en terme de précision et à l'efficacité des méthodes de résolution en terme de temps de calcul.

### 2.3.3. Expression des contraintes fréquentielles du système

Dans le cas des systèmes linéaires, la représentation fréquentielle joue un rôle prépondérant pour l'analyse et la synthèse des systèmes. Cette importance est surtout due à deux raisons majeures :

- La complémentarité entre la formulation fréquentielle et la formulation temporelle des systèmes. En effet, malgré l'apparente redondance sur le plan mathématique, en pratique l'analyse ou la synthèse se font en parallèle dans le domaine temporel et fréquentiel car certaines propriétés sont immédiatement lues sur l'une des deux représentations tandis que d'autres sont immédiatement lues sur la deuxième. En plus, l'utilisation parallèle de ces approches fait apparaître des compromis fondamentaux qui sont au cœur même de l'ingénierie des systèmes.
- La pertinence des outils et des résultats fréquents développés tout au long de l'évolution de l'Automatique. Nous citons, par exemple, l'Automatique fréquentielle classique telle qu'elle s'est développée des années 30 aux années 60 avec les travaux de Black, Nyquist, Bode, Horowitz, etc., et leurs différents diagrammes fréquents et critères graphiques qui permettent d'analyser un système et de synthétiser sa commande. L'intérêt de cette approche est d'autant plus confirmée par l'émergence, vers la fin des années 80, d'une nouvelle branche de l'Automatique fréquentielle dite néoclassique ou avancée ou encore  $H_\infty$ .

Comme dans le domaine temporel, nous distinguons, deux catégories de spécifications fréquentielles et donc deux formulations possibles : la formulation à base d'indicateurs de robustesse et la formulation des spécifications sous forme de gabarits.

#### 2.3.3.1. Formulation des spécifications fréquentielles à base d'indicateurs de robustesse

Mise à part la bande passante qui est souvent définie comme un indicateur de rapidité, le reste des indicateurs fréquents mesure la robustesse du système bouclé vis-à-vis des imperfections du modèle. Cette catégorie de spécifications concerne les indicateurs classiques de robustesse, constitués des marges de phase, de gain, de retard et de module, la pulsation de coupure (bande passante), mais également des paramètres plus avancés, composés des normes infinies des fonctions de sensibilité ou leurs facteurs de résonance.

En conservant les mêmes notations que dans le premier chapitre, nous rappelons la définition des principaux indicateurs fréquentiels pour le cas monovariante :

- Marge de gain ;  $\Delta G = |L(j\omega_{- \pi})|^{-1} \ \ \ \ \ \angle L(j\omega_{- \pi}) = -\pi$
- Marge de phase ;  $\Delta \phi = \inf_i \{ \arg(L(j\omega_c^i)) + \pi \} \ \ \ \ \ |L(j\omega_c^i)| = 1$
- Marge de retard ;  $\Delta R = \Delta \phi / \omega_c \ \ \ \ \ |L(j\omega_c)| = 1$
- Marge de module ;  $\Delta M = \inf_{\omega \geq 0} \{ |1 + L(j\omega)| \} = \|S(j\omega)\|_{\infty}^{-1}$
- Bande passante ;  $B_p = \{ \omega \ \ \ \ \ |L(j\omega)| > 1 \}$ , souvent caractérisée par la pulsation de coupure  $\omega_c$ .
- Facteur de résonance ;  $Q_r = |G(j\omega_r)| / |G(0)| \ \ \ \ \ \omega_r = \arg \left( \sup_{\omega \geq 0} \{ |G(j\omega)| \} \right)$
- Norme  $H_{\infty}$  ;  $\|G(j\omega)\|_{\infty} = \sup_{\omega \geq 0} \{ \bar{\sigma}(G(j\omega)) \}$

où  $L(j\omega)$  est le transfert en boucle ouverte,  $S(j\omega)$  la fonction de sensibilité et  $G(j\omega)$  un transfert stable quelconque de la boucle.

L'ensemble de ces indices de robustesse présente quelques redondances qu'il faut éviter lors de la formulation du cahier des charges. Certaines de ces redondances découlent des définitions (par exemples :  $\Delta R = \Delta \phi / \omega_c$ ,  $Q_r = \|G(j\omega)\|_{\infty} / |G(0)|$ ), d'autres sont plus subtiles comme les relations liant les marges de gain, de phase et de module.

En effet, dans le premier chapitre, il a été montré que les marges de gain et de phase sont moins représentatives que la marge de module. Si l'une d'elles est faible, le système considéré est proche de l'instabilité. Cela peut cependant aussi être le cas pour des systèmes complexes, lorsque les deux marges sont relativement grandes. En d'autres termes, les marges de gain et de phase ne garantissent pas stricto sensu une borne inférieure pour la marge de module  $\Delta M$ . Ces deux marges sont néanmoins très appréciées, car leur ajustement est souvent plus aisé que celui de  $\Delta M$ .

Enfin, une spécification sur la marge de module ou la norme infinie de la fonction de sensibilité garantit une borne inférieure autant pour la marge de phase que pour celle de gain

$$\Delta G \geq \frac{1}{1 - \Delta M} \tag{2-53}$$

$$\Delta \phi \geq 2 \cdot \arcsin \left( \frac{\Delta M}{2} \right) \tag{2-54}$$

***Extension des marges classiques au cas multivariable***

En raison du nombre élevé de correcteurs du type PID utilisés dans le domaine industriel (90% approximativement), les indicateurs fréquentiels classiques jouent un rôle prépondérant dans les cahiers des charges génériques. Leurs définitions ont même été étendues pour le cas multivariable et elles sont même parfois utilisées pour les systèmes non linéaires [Fer96]

En multivariable, on retient deux méthodes pour généraliser le calcul des marges d'un système corrigé :

- Soit on ouvre chacune des boucles de retour élémentaires (monovariante) et on estime tour à tour les marges. Le résultat est alors trop optimiste car il présuppose qu'il n'y a aucune

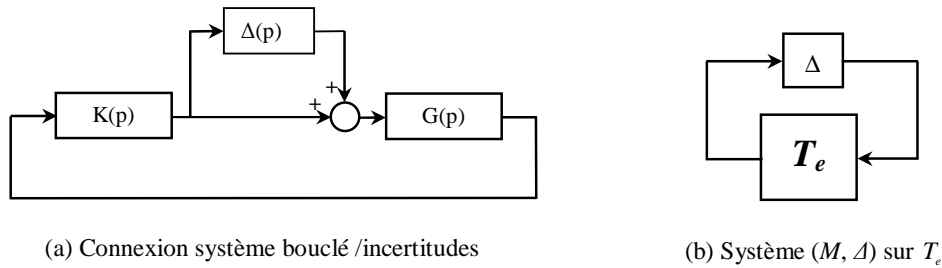


perturbation ou incertitude simultanée sur les autres retours ou, tout au moins, il ne sait pas les prendre en compte.

- Soit on calcule une marge de module multivariable correspondant à une perturbation s’appliquant simultanément sur toutes les boucles élémentaires. On peut alors fournir une interprétation en gain et phase de cette marge de module qui sera néanmoins plus pessimiste.

Nous avons ici choisi de maximiser les marges multivariables et présentons ci-après un complément permettant d’interpréter le résultat multivariable en marges de gains et phases.

Considérons une matrice d’incertitude diagonale multiplicative  $\Delta = \text{diag}[\Delta_1, \dots, \Delta_m]$  appliquée simultanément sur toutes les entrées du système représenté par la fonction de transfert  $G(p) \in \mathcal{RH}_\infty^{m \times p}$  (cf. figure 2.10(a)).



**Fig. 2.10 Incertitudes multiplicatives directes en entrée**

L’application d’une telle perturbation équivaut à multiplier le gain de la  $i^{\text{ième}}$  boucle par  $(1 + \Delta_i)$ . Le transfert  $M$  vu par  $\Delta$  (figure 2.10(b)) est égal à la fonction de sensibilité complémentaire en entrée  $T_e = KG(I_m - KG)^{-1}$ .

Les incertitudes considérées ici sont scalaires et correspondent à des modifications de gain de la boucle MIMO. Si elles étaient complexes, elles correspondraient à des variations en phase sur le cercle unité  $1 + \Delta_i = e^{j\theta_i}$ .

D’après le théorème du petit gain (cf. Théorème 1.9), la boucle fermée est stable si :

$$\bar{\sigma}(\Delta) < \frac{1}{\bar{\sigma}(T_e(j\omega))} \quad \forall \omega \in \mathfrak{R} \quad (2-55)$$

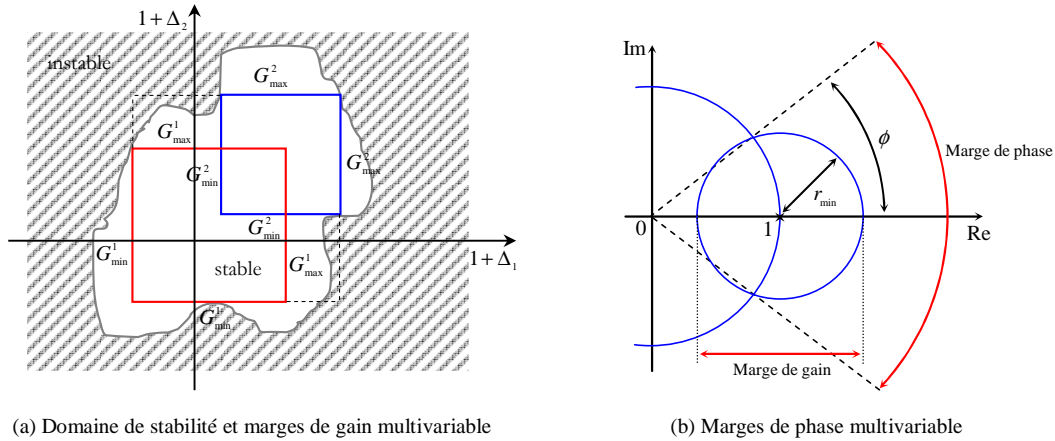
Ainsi on reste stable si :

$$\forall i \quad |\Delta_i| < \frac{1}{\bar{\sigma}(T_e(j\omega))} \quad \forall \omega \in \mathfrak{R} \quad (2-56)$$

Pour remplir cette condition suffisante de stabilité, le gain de boucle modifiée  $(1 + \Delta_i)$  doit rester à l’intérieur du cercle de centre 1 et de rayon  $r_{\min} = \max_{\omega}(\bar{\sigma}(T_e(j\omega)))$ .

**Définition 2.1 (Marge de gain multivariable)** La marge de gain multivariable est définie par un intervalle  $[G_{\min}, G_{\max}]$  tel que la boucle fermée reste stable pour toute perturbation réelle  $\Delta_i$  sur la  $i^{\text{ième}}$  boucle élémentaire (SISO) telle que  $G_{\min} \leq 1 + \Delta_i \leq G_{\max}$ ; les  $\Delta_i$  variant de manière indépendante.

Comme le montre le schéma ci-après (cf. figure 2.11(a)), la marge de gain multivariable est définie autour d'un point nominal et dépend de la valeur de ce point.



**Fig. 2.11 Marges de gain et de phase multivariable**

**Définition 2.2 (Marge de phase multivariable)** La marge de phase multivariable est le plus grand intervalle réel  $[-\phi, \phi]$  tel que la boucle fermée reste stable pour toute perturbation de phase  $1 + \Delta_i = e^{j\phi}$  sur la  $i^{\text{ème}}$  boucle élémentaire (monovariable) telle que  $-\phi \leq \phi_i \leq \phi$  et les  $\Delta_i$  indépendants.

Ainsi, on trouve que sur l'axe des réels, l'intervalle de variation autorisé est de  $1 \pm r_{\min}$ . D'où la marge de gain :

$$\Delta G = [1 - r_{\min}, 1 + r_{\min}] \quad (2-57)$$

De la figure 2.11(b), on en déduit par de simples considérations géométriques que :

$$\Delta\phi = [-\phi, \phi] \quad \text{avec} \quad \phi = 2 \cdot \arcsin\left(\frac{r_{\min}}{2}\right) \quad (2-58)$$

Nous avons considéré ici le cas d'une perturbation en entrée, mais cette méthodologie pourrait également être appliquée pour le cas d'une perturbation en sortie et avec n'importe quel type d'incertitudes structurées (cf. paragraphe 1.5.3). Ainsi, on peut toujours traduire la marge de stabilité de la boucle en équivalent marge de phase et marge de gain garantis. Seule l'expression des transferts change.

#### **Expression des spécifications fréquentielles à base d'indicateurs de robustesse :**

L'expression des spécifications fréquentielles concernant les indicateurs de robustesse se traduit souvent par une contrainte de type :

$$\alpha_{\omega}^{\min} \leq \alpha_{\omega} \leq \alpha_{\omega}^{\max} \quad (2-59)$$

où  $\alpha_{\omega}^{\min}$  et  $\alpha_{\omega}^{\max}$  sont, respectivement, la borne minimale et la borne maximale de l'indicateur de robustesse  $\alpha_{\omega}$ .

### 2.3.3.2. Formulation des spécifications fréquentielles sous forme de gabarits

Avec le développement de la méthode  $H_\infty$ , l'expression des objectifs du cahier des charges sous forme de gabarits fréquentiels est devenue une technique courante. Le principe de cette approche est similaire à celui des enveloppes temporelles ; il s'agit de forcer les modules des transferts à respecter des gabarits désirés. Ces gabarits traduisent la performance et/ou la robustesse de la loi de commande via des contraintes sur le gain en boucle ouverte ou en boucle fermée

Ce type de spécifications a été employé, initialement, dans les méthodes fréquentielles classiques où dans une première étape, les cahiers des charges temporels sont inévitablement traduits par des spécifications sur le module des transferts en boucle fermée (cf. section 1.5.2 et 1.5.3). Comme ces fonctions dépendent non linéairement du correcteur  $K(p)$  que l'on cherche à mettre au point (cf. tableau 1.1), une recherche "manuelle" du correcteur  $K(p)$  de façon à satisfaire ces contraintes peut être très complexe, même dans le cas d'une structure simple pour  $K(p)$ . L'idée fondamentale des méthodes fréquentielles classiques de synthèse de lois de commande consiste à transformer ces contraintes portant sur les transferts du système en boucle fermée en contraintes portant sur le transfert du système en boucle ouverte  $L(p) = G(p)K(p)$ , en utilisant les liens qui existent entre ces transferts (cf. figure 1.19)<sup>1</sup>. L'intérêt est que  $L(p)$  est une fonction linéaire de  $K(p)$ . L'étape suivante du processus de conception est donc de rechercher les contraintes que doit satisfaire le transfert en boucle ouverte  $L(p) = G(p)K(p)$  pour que le cahier des charges soit rempli. Une fois que celles-ci sont déterminées, le correcteur  $K(p)$  doit être recherché manuellement de façon à ce que  $L(p)$  les satisfassent. La tâche est facilitée par le fait que  $L(p)$  est une fonction linéaire de  $K(p)$ . Cette recherche se fait à l'aide du diagramme de Bode, du diagramme de Black, etc., de la fonction de transfert  $L(p)$ . Pour cela, il est nécessaire de choisir une structure adaptée pour la loi de commande comme par exemple un Proportionnel Intégral avec ou sans avance de phase.... La structure étant déterminée, les paramètres du correcteur doivent être réglés de façon à ce que le transfert en boucle ouverte  $L(p)$  remplisse le cahier des charges. Il est enfin impératif de vérifier si le correcteur obtenu remplit bien le cahier des charges initial. Si ce dernier n'est pas satisfait, alors soit il n'existe pas de correcteur remplissant le cahier des charges, soit un mauvais choix a été fait à l'une des étapes du processus de conception : cela peut venir d'une mauvaise traduction des spécifications temporelles du cahier des charges en contraintes sur les transferts en boucle fermée<sup>2</sup> ou des contraintes sur les transferts en boucle fermée en contraintes sur le transfert en boucle ouverte ou encore d'un mauvais choix de structure ou de paramètres pour le correcteur. Etant donnée l'abondance des possibilités, sans un bon savoir-faire et de la patience, il est donc difficile de clairement identifier pourquoi le correcteur obtenu ne remplit pas le cahier des charges, où le processus de conception doit être repris et enfin comment il doit être modifié. Dans le cas multivariable, la difficulté est rapidement accrue avec le nombre de spécifications ce qui rend la recherche d'un compromis très difficile.

La synthèse  $H_\infty$ , quand à elle, est née de la recherche d'un outil qui, à partir des contraintes sur les modules des transferts en boucle fermée, recherche directement s'il existe un correcteur  $K(p)$  tel que les transferts en boucle fermée satisfassent les contraintes sur leur module et, si oui, fournit un tel correcteur. Avec un tel outil, si l'algorithme ne fournit pas de correcteur, c'est que soit il n'existe pas

<sup>1</sup> Ces contraintes sur la fonction de transfert en boucle ouverte sont traduites sur une ou plusieurs représentations graphiques (diagrammes de Bode, de Nyquist et/ou de Black Nichols). C'est l'utilisation de ces représentations graphiques qui rend ces méthodes extrêmement attractives.

<sup>2</sup> Rappelons que le passage des unes aux autres est assez approximatif.

de correcteur remplissant le cahier des charges, soit que la traduction des spécifications temporelles du cahier des charges en contraintes sur les fonctions de transfert en boucle fermée est mauvaise. Il en résulte, d'une part, une considérable simplification du processus de conception et, d'autre part, la possibilité de réellement tester l'existence d'un correcteur vérifiant le cahier des charges (après reformulation dans le domaine fréquentiel).

Les formulations fréquentielles établies dans le premier chapitre (cf. section 1.5.2 et 1.5.3), s'écrivent, sous la forme semi-infinie suivante :

$$\forall \omega, \quad |T_{\alpha \rightarrow \beta}(j\omega)| \leq |T_{\alpha \rightarrow \beta}^{\max}(j\omega)| \quad (2-60)$$

où  $|T_{\alpha \rightarrow \beta}(j\omega)|$  et  $|T_{\alpha \rightarrow \beta}^{\max}(j\omega)|$  sont respectivement, le module de la fonction de transfert entre l'entrée  $\alpha$  et la sortie  $\beta$  et son gabarit maximal.

Le gabarit  $|T_{\alpha \rightarrow \beta}^{\max}(j\omega)|$  est appelé aussi pondération et il est définie selon la classe de signaux d'entrée et de sortie associé au transfert  $T_{\alpha \rightarrow \beta}(j\omega)$ .

Dans le cas où  $T_{\alpha \rightarrow \beta}^{\max}(j\omega)$  est inversible, la contrainte (2-60) peut aussi s'écrire en utilisant la norme  $H_{\infty}$  :

$$\left\| T_{\alpha \rightarrow \beta} \cdot (T_{\alpha \rightarrow \beta}^{\max})^{-1} \right\|_{\infty} \leq 1 \quad (2-61)$$

Nous rappelons (cf. section 1.5.2) que la performance d'un cahier des charges revient à tester, si pour les pondérations définissant les signaux de référence  $W_r$ , de perturbation  $W_b$  et de bruit  $W_d$  et les pondérations définissant les signaux d'erreur  $W_e$  et de commande  $W_u$ , les conditions suivantes sont satisfaites :

- Suivi de trajectoires de référence  $\|W_e T_{r \rightarrow e} W_r\|_{\infty} \leq 1$  ;
- Rejet de perturbations  $\|W_e T_{b \rightarrow e} W_b\|_{\infty} \leq 1$  ;
- Atténuation des bruits  $\|W_u T_{d \rightarrow u} W_d\|_{\infty} \leq 1$  ;
- Commandes modérées  $\|W_u T_{r \rightarrow u} W_r\|_{\infty} \leq 1$  ;

Les transferts, de la boucle fermée  $T_{b \rightarrow e}(j\omega)$ ,  $T_{r \rightarrow u}(j\omega)$ ,  $T_{w \rightarrow u}(j\omega)$  et  $T_{r \rightarrow e}(j\omega)$ , qui définissent la performance d'un cahier des charges s'écrivent à partir des fonctions de sensibilité, ce qui engendre des contraintes contradictoires mais toutes relatives à la nature de chacun des signaux de la boucle (hautes et basses pulsations). Les gabarits typiques d'un système performant sont illustrés dans la figure 1.20.

Nous rappelons aussi que des conditions similaires sont obtenues pour exprimer la robustesse du système bouclé vis-à-vis des incertitudes. Dans ce cas, les gabarits dépendent de la structure d'incertitude considérée (cf. tableau 1.4).

Par exemple, pour une incertitude additive directe  $\Delta$  vérifiant  $\|\Delta(j\omega)\|_{\infty} \leq \|W_a(j\omega)\|_{\infty}$ , la condition nécessaire et suffisante de stabilité est donnée par  $\|W_a(j\omega)K(j\omega)S(j\omega)\|_{\infty} < 1$ .

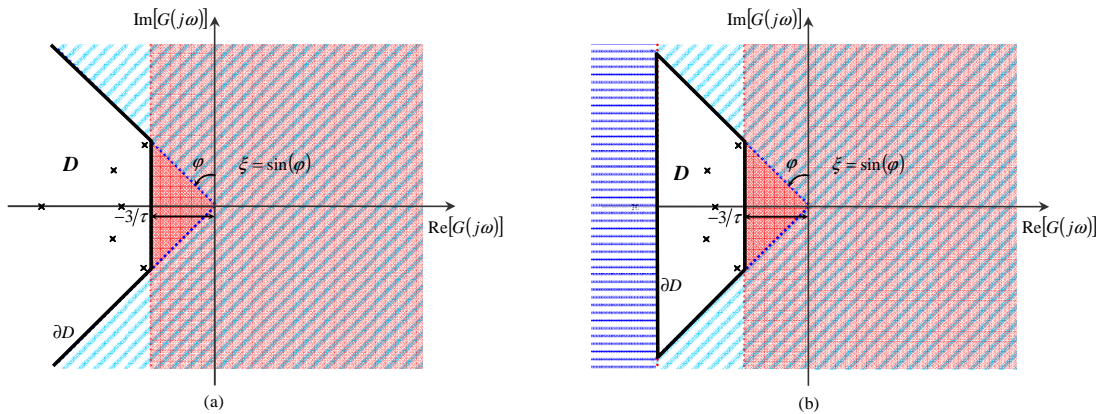
**2.3.3.3. Formulation des spécifications fréquentielles modales**

Ce type de demandes est issu, initialement, de l’approche fréquentielle classique et a été développé, par la suite, à l’aide des méthodes d’espace d’état. Quelque soit l’approche, le premier objectifs d’une commande est de stabiliser le système, si celui-ci est instable ou de conserver sa stabilité. De plus, il faut souvent augmenter son “degré de stabilité” et éviter des oscillations mal amorties dans le régime transitoire. Parallèlement, on peut constamment chercher à améliorer la rapidité du système sans dégrader son amortissement. Ces performances transitoires du système bouclé dépendent fortement de la localisation du spectre de la matrice d’état en boucle fermée (ou des racines du polynôme caractéristique) dans le plan de Laplace. Dans cette approche, les spécifications modales s’interprètent indirectement en termes de placement des pôles de la boucle fermée.

Ces modes ont des contributions différentes qui peuvent être évaluées grâce à une simulation modale [Mag02]. Apparaît alors la notion de modes dominants. Pour ces derniers, il est possible d’énoncer les règles typiques suivantes : pour un temps de réponse désiré  $\tau_d$  et un amortissement désiré  $\xi_d$ , les valeurs propres dominantes en boucle fermée  $\lambda$  doivent vérifier :

$$\begin{cases} \text{Re}(\lambda) \leq -3/\tau_d \\ |\text{Re}(\lambda)/\lambda| \geq \xi_d \end{cases} \tag{2-62}$$

Ces contraintes définissent la région  $D$  du plan complexe où doivent se situer les pôles (cf. figure 2.12(a)).



**Fig. 2.12 Zone du plan complexe correspondant aux performances temporelles désirées.**

La commande est réalisée par des organes de puissance (servocommandes) ayant des bandes passantes limitées. C’est pourquoi une contrainte supplémentaire est imposée aux modes de la boucle fermée qui doivent se situer dans cette bande passante. Il conviendra donc de fermer ce domaine (contour  $\partial D$  fermé) en imposant une borne supérieure à la partie réelle des pôles  $\text{Re}(\lambda)$  (cf. figure 2.12(b)).

Similairement aux indicateurs de robustesse, les spécifications modales sont formulées par des contraintes minmax, on écrit :

$$\Lambda^{\min} \leq \Lambda(\lambda) \leq \Lambda^{\max} \tag{2-63}$$

**Définition 2.3 (La  $D$ -stabilité)** Une matrice  $A \in \mathbb{R}^{n \times n}$  est  $D$ -stable si et seulement si l’ensemble de ses valeurs propres est strictement contenu à l’intérieur d’une région  $D$  du plan complexe.

La notion de stabilité est alors généralisée selon le domaine du plan complexe traduisant les performances temporelles désirées et limitant donc la dispersion des pôles. Par exemple : on parle d' $\alpha$ -stabilité en imposant une contrainte  $\text{Re}(\lambda) \leq -\alpha$  qui permet de garantir que tous les régimes transitoires du système soient plus rapide que  $e^{-\alpha t}$  et ceci sans aucune contrainte sur le dépassement.

Dans certains systèmes à structure complexe (modes rigides/souples, modes lents/rapides), il est possible de définir plusieurs régions de stabilité disjointes afin de caractériser les performances du système.

### 2.3.4. Faisabilité des cahiers des charges fréquentiels

En suivant une démarche similaire à celle employée pour formuler la problématique de faisabilité des spécifications temporelles en un problème d'optimisation, nous allons présenter deux approches qui consistent à exprimer les spécifications fréquentielles en deux sortes de problèmes d'optimisation.

La première approche est à base d'opérateurs (matrices de fonctions de transfert). Elle caractérise l'ensemble de correcteurs qui permet une paramétrisation linéaire des transferts en boucle fermée et qui stabilise le système bouclé. Cela nous permet de formuler les contraintes fréquentielles de gabarits en problèmes d'optimisation convexes avec des propriétés semblables à celles obtenues dans l'approche temporelle entrées/sortie.

La deuxième approche est générique car elle consiste à considérer un cahier des charges fréquentiel avec les trois classes de spécifications fréquentielles. Le problème de faisabilité formulé est non convexe et sa résolution dépend fortement de la structure de correcteur considérée.

#### 2.3.4.1. Etude de la faisabilité des spécifications fréquentielles par une approche par opérateurs

Considérons à nouveau le schéma fonctionnel d'une régulation classique (cf. figure 2.13) :

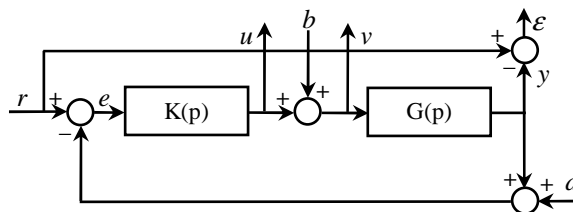


Fig. 2.13: Schéma fonctionnel d'une boucle de régulation

Nous notons  $\Phi$  l'opérateur qui associe à tout vecteur d'entrées  $(r \ b \ d)^T$  le vecteur de sorties  $(\varepsilon \ u \ v \ y)^T$ .  $\Phi$  est définie par l'équation suivante :

$$\begin{pmatrix} \varepsilon \\ u \\ v \\ y \end{pmatrix} = \Phi \begin{pmatrix} r \\ b \\ d \end{pmatrix} \quad \text{avec} \quad \Phi = \begin{pmatrix} S & -GS_e & T \\ KS & -T_e & -KS \\ KS & S_e & -KS \\ T & GS_e & -T \end{pmatrix} \quad (2-64)$$

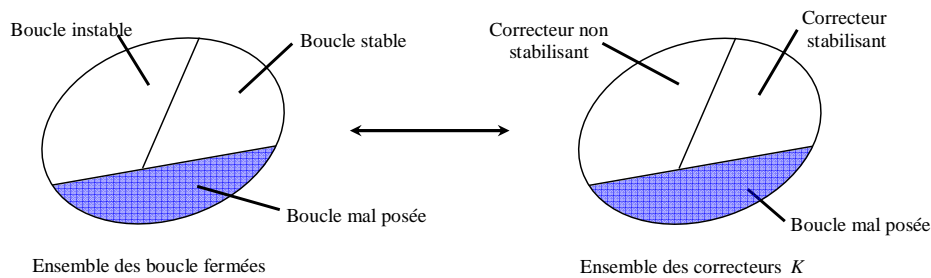
On rappelle les propriétés du système bouclé suivantes :

**Définition 2.4 (Boucle bien posée)** Une boucle de régulation conforme à la figure 2.13 est dite bien posée si à tous signaux  $r, b$  et  $d$  correspond un et un seul ensemble de signaux  $\varepsilon, u, v$  et  $y$ . En d'autres termes si l'application  $\Phi$  est injective.

Pour le cas des systèmes linéaires invariants, la boucle de régulation est bien posée si et seulement si  $(I_p + GK)$  est inversible ( $\Leftrightarrow (I_m + KG)$  inversible), c'est-à-dire que l'un de ces deux transferts est identiquement non nul.

**Définition 2.5 (Stabilité entrée/sortie d'un système linéaire invariant)** Un système bouclé est stable E/S si et seulement si, tous les transferts définis par sa boucle sont stables

La figure 2.14 représente les différents cas possibles dans l'espace des boucles fermées ainsi que le schéma équivalent dans l'espace des correcteurs.



**Fig. 2.14: Dualité boucle fermée - correcteur**

**Changement de variable linéarisant**

Chaque transfert de la boucle fermée est exprimé en fonction du correcteur  $K$  et fait intervenir l'inverse de  $(I_p + GK)$  ou de  $(I_m + KG)$ . Ainsi, les contraintes fréquentielles du cahier des charges sont exprimées non linéairement en fonction du correcteur  $K$  et ils nécessitent une inversion matricielle ce qui rend les problèmes d'optimisation formulés complexes et difficile à résoudre. Afin de pallier à cette difficulté, on introduit un changement de variable qui permet d'exprimer tous les transferts des boucles fermées bien posées sous une forme affine.

En analysant tous les transferts de la boucle et les différents changements de variable envisageables, on constate que le choix  $\mathbf{L} = T_{r \rightarrow u} = KS = K(I_p + GK)^{-1}$  permet de linéariser tous les transferts de la boucle sans aucune hypothèse particulière sur l'inversibilité de  $\mathbf{L}$  ou de  $G$ . L'ensemble des relations affines par rapport au paramètre fonctionnel  $\mathbf{L}$  est donné par :

$$\begin{cases} T = G \mathbf{L} \\ S = I_p - G \mathbf{L} \\ T_e = \mathbf{L} G \\ S_e = I_m - \mathbf{L} G \\ GS_e = G - G \mathbf{L} G \end{cases} \tag{2-65}$$

Pour tout système  $G$  donné, la relation entre  $K$  et  $\mathbf{L}$  est une bijection sur l'espace de définition de la transformation. Cet espace correspond aux cas des boucles bien posées. On écrit :

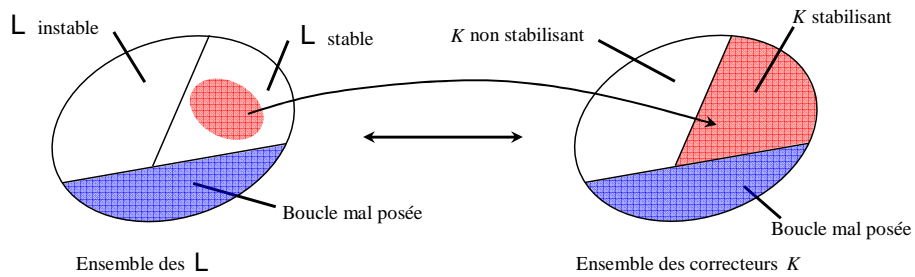
$$\mathbf{L} = K(I_p + GK)^{-1} \Leftrightarrow K = \mathbf{L} (I_p - G\mathbf{L})^{-1} \tag{2-66}$$

Ce transfert linéarisant permet également d’obtenir une interprétation au résultat de paramétrisation des trajectoires de commande obtenu dans l’approche E/S (cf. section 2.3.2.1).

En effet, soient  $\mathbf{L}_{ij}$  les différents éléments de  $\mathbf{L}$ , et  $u_j$  le vecteur définie par :  $u_j = (\mathbf{L}_{1j} \dots \mathbf{L}_{mj})^T$ . Ce dernier s’interprète comme la commande obtenue en appliquant un Dirac à la  $j^{\text{ième}}$  entrée. Comme les transferts de la boucle fermée sont des fonctions linéaires de  $\mathbf{L}$ , ils sont aussi des fonctions linéaires des  $u_j$ . Nous retrouvons ici le même vecteur introduit dans le paragraphe 2.3.2.1 ainsi que le fait de pouvoir paramétrer les comportements d’une boucle fermée par un ensemble bien choisi de commandes.

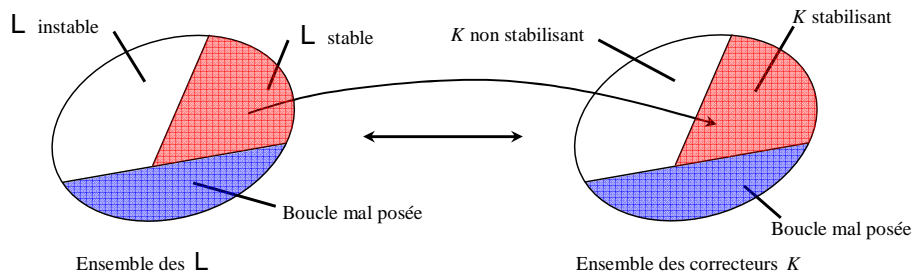
**La stabilité dans l’espace des paramètres  $\mathbf{L}$**

Dans ce paragraphe, on s’intéresse à la caractérisation de l’ensemble des paramètres  $\mathbf{L}$  associé à l’ensemble des boucles fermées stables. Il est clair qu’un paramètre linéarisant  $\mathbf{L}_0$  assurant la stabilité de la boucle fermée est stable puisque c’est l’un des transferts particuliers de la boucle fermée :  $\{\mathbf{L} \setminus \text{Boucle fermée stable}\} \subset \{\mathbf{L} \setminus \mathbf{L} \text{ stable}\}$ . Ce cas général est représenté par la figure 2.15 dans l’espace des correcteurs et dans l’espace des paramètres  $\mathbf{L}$ .



**Fig. 2.15: Caractérisation des paramètres  $\mathbf{L}$  stabilisants**

En dehors du transfert linéarisant  $\mathbf{L}$ , les transferts en boucle fermée donnés par (2.60) dépendent du système  $G$ . Ainsi, si le système  $G$  est stable, les deux ensembles  $\{\mathbf{L} \setminus \text{Boucle fermée stable}\}$  et  $\{\mathbf{L} \setminus \mathbf{L} \text{ stable}\}$  seront identiques. À ce moment, on écrit :  $\{\mathbf{L} \setminus \text{Boucle fermée stable}\} = \{\mathbf{L} \setminus \mathbf{L} \text{ stable}\}$  et la représentation graphique 2.13 devient :



**Fig. 2.16: Caractérisation des paramètres  $\mathbf{L}$  stabilisants pour  $G$  stable**

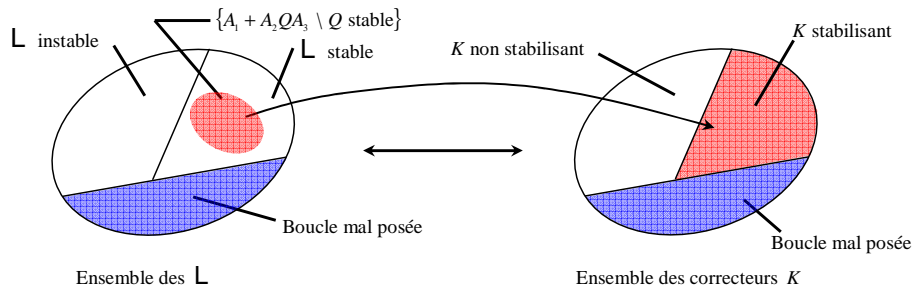


Dans le cas où  $G$  est instable, un choix dans l'ensemble  $\mathbf{L}$  ne garantit a priori que la stabilité des transferts  $T_{u \rightarrow r} = T_{v \rightarrow r} = KS = \mathbf{L}$ . Afin de conserver la propriété linéaire du changement de variable, on propose de chercher une restriction à l'ensemble  $\mathbf{L}$  sous la forme suivante :

$$\mathbf{L} = A_1 + A_2QA_3 \quad (2-67)$$

où les  $A_1$ ,  $A_2$  et  $A_3$  sont des transferts fonction du système et  $Q$  un transfert stable.

Le but est de chercher les conditions sur  $A_1$ ,  $A_2$  et  $A_3$  pour que l'égalité suivante soit vérifiée :  $\{\mathbf{L} \setminus \text{Boucle fermée stable}\} = \{\mathbf{L} = A_1 + A_2QA_3 \setminus Q \text{ stable}\}$ . Cette égalité se traduit dans l'espace des paramètres  $\mathbf{L}$  stabilisants par la figure 2.17.



**Fig. 2.17: Caractérisation des paramètres  $\mathbf{L}$  stabilisants dans le cas général**

Le transfert E/S global  $\Phi$  correspond à une expression affine en fonction du nouveau paramètre  $Q$ . On écrit :

$$\Phi = T_1 + T_2QT_3 \quad (2-68)$$

avec :

$$\Phi = \begin{pmatrix} S & -GS_e & T \\ KS & -T_e & -KS \\ KS & S_e & -KS \\ T & GS_e & -T \end{pmatrix} = \underbrace{\begin{pmatrix} -G \\ I \\ I \\ G \end{pmatrix} A_1 (I \ -G \ -I)}_{T_1} + \underbrace{\begin{pmatrix} I & -G & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & G & 0 \end{pmatrix}}_{T_2} + \underbrace{\begin{pmatrix} -G \\ I \\ I \\ G \end{pmatrix} A_2 QA_3 (I \ -G \ -I)}_{T_3}$$

Pour que le système bouclé soit stable, les transferts  $A_1$ ,  $A_2$  et  $A_3$  doivent vérifier :

$$\begin{cases} T = GA_1 + GA_2QA_3 & \text{stable} \\ KS = A_1 + A_2QA_3 & \text{stable} \\ T_z = A_1G + A_2QA_3G & \text{stable} \\ GS_e = G - GA_1G - GA_2QA_3G & \text{stable} \end{cases} \quad (2-69)$$

Dans le cas particulier où le transfert  $Q$  est nul ( $Q = 0$ ), on obtient un ensemble de transferts de boucle stable et en particulier un  $\mathbf{L}_0 = A_1$ . Ayant une bijection entre les transferts de la boucle ouverte et les correcteurs, nous avons donc un correcteur  $K_0$  stabilisant la boucle, tel que :

$$\begin{cases} GA_1 = T_0 & \text{stable} \\ A_1 = K_0 S_0 = L_0 & \text{stable} \\ A_1 G = T_{z_0} & \text{stable} \\ G - GA_1 G = GS_{e_0} & \text{stable} \end{cases} \quad (2-70)$$

Le correcteur stabilisant associé aux transferts de la boucle pour ( $Q=0$ ) est nommé solution centrale et peut être choisi parmi tous les correcteurs stabilisants. En effet, pour une paramétrisation de type (2-66) et un paramètre  $Q_0$  correspondant à une solution particulière  $\Phi_0 = T_1 + T_2 Q_0 T_3$ , le choix  $Q' = Q - Q_0$  permet de rendre cette solution centrale. On écrit alors  $\Phi = \Phi_0 + T_2 Q' T_3$ .

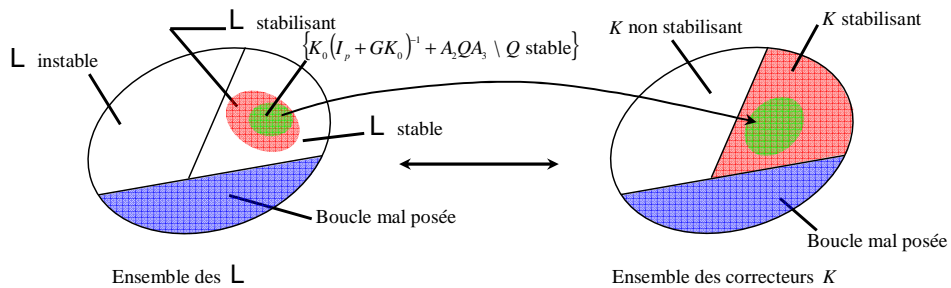
Cette propriété reste valable dans le cas d'un système  $G$  stable. Dans ce cas, on peut choisir  $K_0 = 0$ . Ceci est équivalent à mettre  $L_0 = A_1 = 0$ , ce qui simplifie l'expression de  $T_1$  dans (2-68).

Ce choix "naturel" n'est pas toujours le meilleur. On verra par la suite qu'il est préférable parfois de se placer autour d'une solution centrale plus adaptée au problème considéré. Cela aura des conséquences sur le comportement numérique et sur les temps de résolution des problèmes d'optimisation qu'on va formuler.

Si les transferts  $A_2$  et  $A_3$  sont stables, alors le paramètre  $L = K_0 (I_p + GK_0)^{-1} + A_2 Q A_3$  est stabilisant, ceci est équivalent à :

$$\{L = K_0 (I_p + GK_0)^{-1} + A_2 Q A_3 \mid Q \text{ stable}\} \subset \{L \mid \text{Boucle fermée stable}\}$$

Cette condition suffisante garantit la stabilité mais ne permet pas d'affirmer la paramétrisation de tous les correcteurs  $K$  stabilisants (cf. figure 2.18).



**Fig. 2.18: Caractérisation d'un mauvais choix de  $A_2$  et  $A_3$ .**

Cependant, si les transferts  $A_2$  et  $A_3$  sont stables et si pour tout correcteur central  $K_0$  stabilisant, il existe un transfert  $Q$  stable tel que :  $L_0 + A_2 Q A_3 = K_0 (I_p + GK_0)^{-1}$  alors :

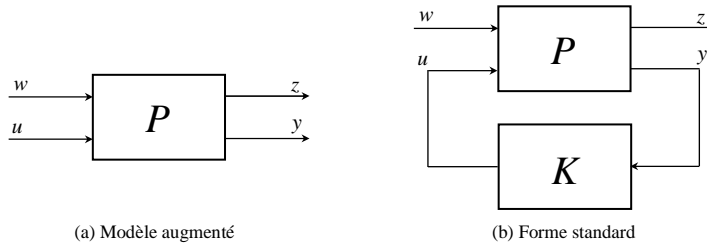
$$\{L = K_0 (I_p + GK_0)^{-1} + A_2 Q A_3 \mid Q \text{ stable}\} = \{L \mid \text{Boucle fermée stable}\}$$

### Généralisation à la forme standard

On appelle modèle augmenté le système  $P(p)$  formé des quatre transferts multivariables existant entre les entrées  $u$  et  $w$  et les sorties  $w$  et  $z$  où :

- $u$  est la commande du système
- $w$  sont les entrées exogènes qui peuvent être les consignes ou les perturbations

- $y$  sont les mesures
- $z$  sont les sorties régulées.



**Fig. 2.19: Modèle augmenté et forme standard**

On décompose classiquement  $P$  en ces quatre transferts multivariables de la manière suivante :

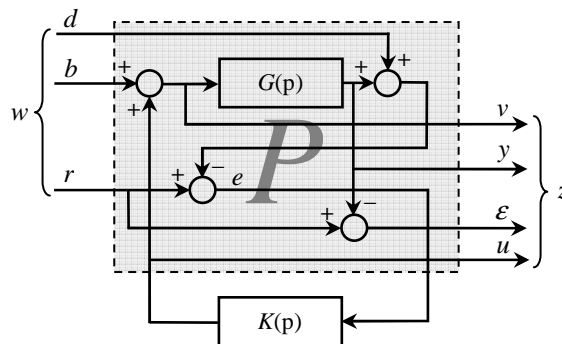
$$\begin{pmatrix} z(p) \\ y(p) \end{pmatrix} = P \cdot \begin{pmatrix} w(p) \\ u(p) \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \cdot \begin{pmatrix} w(p) \\ u(p) \end{pmatrix} \tag{2-71}$$

L'intérêt principal de l'introduction de la forme standard est avant tout méthodologique. Elle apporte en effet une certaine clarté de formulation en permettant de représenter à la fois le système à contrôler et le cahier des charges associé (cf. figure 2.19 (b)).

Lorsqu'on applique une loi de commande de type retour de sortie  $u = K.y$  sur le système, on obtient l'expression suivante du transfert entre les entrées exogènes  $w$  et les sorties régulées  $z$ , qui est appelée Transformation Fractionnaire Linéaire (LFT) inférieure :

$$H_{w \rightarrow z}(p) = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}.K)^{-1}P_{21} \tag{2-72}$$

Cette formulation permet de prendre en compte, en plus des transferts de boucles, des entrées et des sorties supplémentaires pour traduire les spécifications du cahier des charges. Le schéma bloc 2.10 peut être mis sous la forme standard comme indiqué sur la figure 2.20 :



**Fig. 2.20: Mise sous forme standard du schéma bloc classique**

Cette boucle de régulation est bien posée si et seulement si  $(I - P_{22}K)$  est inversible ( $\Leftrightarrow (I - KP_{22})$  inversible). Son transfert  $H_{w \rightarrow z}$  est une fonction non linéaire du correcteur  $K$  (cf. relation (2-68)).

Afin de paramétrer linéairement l'ensemble des boucles bien posées, on introduit le transfert  $L$  défini par :

$$\mathbf{L} = K(I - P_{22}K)^{-1} \quad (2-73)$$

L'application qui associe à tout correcteur  $K$  le paramètre  $\mathbf{L}$  est un changement de variable linéarisant défini sur l'espace des correcteurs correspondant à des boucles bien posées. On écrit :

$$H_{w \rightarrow z} = F_l(P, K) = P_{11} + P_{12} \mathbf{L} P_{21} \quad (2-74)$$

Des propriétés similaires à celles développées dans le paragraphe précédent peuvent être formulées pour caractériser l'ensemble des boucles fermées stables par un nouveau changement de variable, ce qui revient à caractériser l'ensemble des paramètres  $\mathbf{L}$  correspondant à des boucles fermées stables.

Les différents paramètres  $A_1, A_2, A_3, \mathbf{L}$  et  $Q$  peuvent être retrouvés par identification de la forme standard avec les expressions (2-67) et (2-68).

### ***Paramétrisation de Youla-Kučera***

Nous avons montré jusqu'à ici que sous certaines hypothèses, l'ensemble des transferts de boucle fermée stable peut se mettre sous la forme  $T_1 + T_2 Q T_3$  où  $Q$  est un transfert stable et  $T_1, T_2$  et  $T_3$  sont des transferts dépendant du système  $G$  (cf. équation (2-68)).

Pour les systèmes linéaires invariants, il existe une paramétrisation particulière de cette forme c'est celle de Youla- Kučera dite aussi  $Q$ -paramétrisation. Ces origines remontent aux travaux de J.R. Ragazzini et G.F. Franklin sur la commande optimale [Rag58], puis les travaux de Kučera pour les systèmes discrets [Kuc74], et ensuite ceux de Youla et co-auteurs dans le cadre du filtrage de Wiener Hopf [You76]. C'est en l'occurrence Youla-Kučera que l'histoire a retenu, donnant le nom de Youla-Kučera paramétrisation à ces techniques.

Les résultats de paramétrisation précités permettent de donner un sens au paramètre  $Q$  de cette paramétrisation, et en particulier à son rapport au paramètre linéarisant  $\mathbf{L}$  (non nécessairement associé à une boucle fermée stable), aux transferts de la boucle fermée et au choix d'une solution centrale.

Nous introduirons le principe de la paramétrisation de la boucle fermée à travers les travaux de [Vid85] avant d'exposer une méthode pratique pour la réaliser utilisant la forme estimation/commande des correcteurs initiée par [Sch80].

L'existence de la paramétrisation de Youla-Kučera pour tout système linéaire invariant se fait à base de la factorisation co-première [Vid85].

***Théorème 2.1 (Factorisation première [Vid85; Fra87])*** *Pour tout système  $P_{22}$  propre, il existe huit matrices de transfert propres et stables vérifiant les équations :*

$$P_{22} = N M^{-1} = \tilde{M}^{-1} \tilde{N} \quad (2-75)$$

$$\begin{pmatrix} \tilde{X} & -\tilde{Y} \\ -\tilde{N} & \tilde{M} \end{pmatrix} \begin{pmatrix} M & Y \\ N & X \end{pmatrix} = I \quad (2-76)$$

**Théorème 2.2** ([Vid85; Fra87]) Soit une double factorisation première de  $P_{22}$  fournie par les équations du théorème 2.1. Tout correcteur stabilisant  $P_{22}$ , et donc  $P$ , est donné par l'équation suivante :

$$K = (Y - M Q)(X - N Q)^{-1} = (\tilde{X} - Q \tilde{N})^{-1} (\tilde{Y} - Q \tilde{M}) \quad (2-77)$$

où  $Q$  est une matrice de transfert propre et stable.

Ainsi, tout correcteur stabilisant  $P_{22}$  vérifie (2-77) et réciproquement, tout correcteur donné par (2-77) stabilise  $P_{22}$ . Ces deux théorèmes nous permettent d'exprimer le transfert linéarisant  $L$  par :

$$L = K(I - P_{22}K)^{-1} = (Y M - Q) \tilde{M} \quad (2-78)$$

En remplaçant dans (2-74), on obtient finalement :

$$H_{w \rightarrow z} = \underbrace{P_{11} + P_{12} Y \tilde{M} P_{21}}_{T_1} + \underbrace{(-P_{12} M)}_{T_2} \underbrace{Q \tilde{M} P_{21}}_{T_3} \quad (2-79)$$

On obtient ainsi une représentation paramétrée de l'ensemble des boucles fermées atteignables par un correcteur stabilisant :

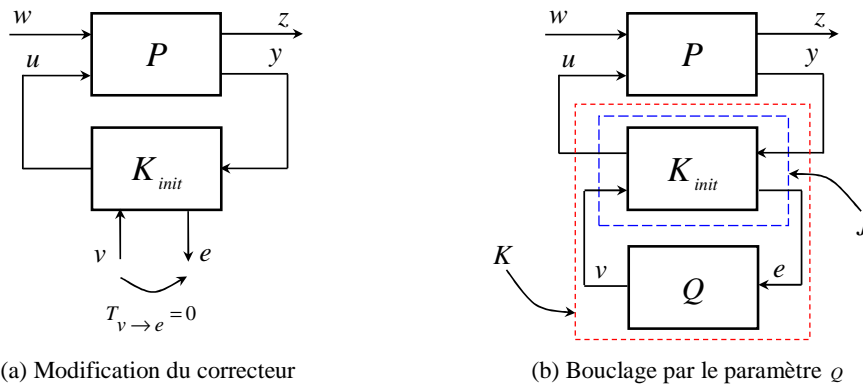
$$H_{stable}(P) = \{T_1 + T_2 Q T_3 \mid Q \text{ stable}\} \quad (2-80)$$

$T_1$ ,  $T_2$  et  $T_3$  étant uniquement définis à partir de la factorisation première du système initial, le parcours de l'espace  $Q_{stable} = \{Q_3 \mid Q \text{ stable}\}$  permet d'explorer  $H_{stable}(P)$  totalement (cf. figure 2.17).

### Mise en œuvre de la paramétrisation de Youla-Kučera

Considérons un correcteur initial  $K_{init}$  stabilisant la boucle fermée. On cherche à paramétrer par le paramètre  $Q$  l'ensemble des correcteurs stabilisants. On procède en deux étapes :

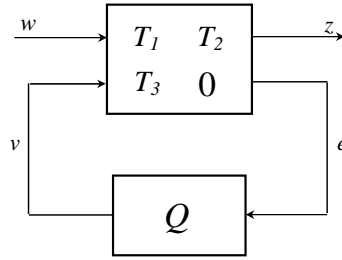
- On modifie le correcteur initial en rajoutant une entrée  $v$  et une sortie  $e$  (cf. figure 2.21(a)).  $v$  est de même dimension que  $u$  et  $e$  de même dimension que  $y$ . Cette modification du correcteur  $K_{init}$  est réalisée de façon à ce que le transfert  $T_{v \rightarrow e}$  soit nul.



**Fig. 2.21: Paramétrisation des correcteurs stabilisants**

- On rajoute entre  $v$  et  $e$  une matrice de transfert  $Q$  stable de dimension  $(n_y \times n_u) = (p \times m)$  (cf. figure 2.21(b)). On obtient un nouveau correcteur  $K$  construit à partir du correcteur  $K_{init}$  et paramétré par  $Q$ . Comme le transfert  $T_{v \rightarrow e}$  est nul, la boucle fermée initiale et  $Q$  étant stables, le nouveau correcteur ne peut déstabiliser la boucle fermée. Par contre  $Q$  agit sur le transfert  $T_{v \rightarrow e}$ , ce qui permet d’agir sur les performances et la robustesse du correcteur. On note par  $J$  la matrice d’interconnexion résultante

En notant par  $T_1, T_2$  et  $T_3$  les transferts entre  $w$  et  $z$ ,  $v$  et  $z$  et enfin  $w$  et  $e$  respectivement, le système bouclé peut se mettre sous la forme standard décrite par la figure 2.22.



**Fig. 2.22: Schéma de la paramétrisation de Youla-Kučera**

Le transfert en boucle fermée  $T_{w \rightarrow z}$  s’écrit donc :

$$H_{w \rightarrow z} = F_l(T, Q) = T_1 + T_2 Q T_3 \quad (2-81)$$

Cette forme affine en  $Q$  est similaire à celle d’une paramétrisation de Youla-Kučera mais elle ne garantit pas un changement de variable qui permet d’atteindre tous les transferts stables possibles. On propose alors une paramétrisation permettant de garantir une bijection entre l’ensemble des paramètres  $Q$  linéarisants et stables et l’ensemble des correcteurs  $K$  stabilisant la boucle.

Si on considère un correcteur initial  $K_{init}$  sous la forme retour d’état-observateur, J.C. Doyle a montré qu’il était possible de le paramétrer à l’aide de  $Q$ . Soit le système strictement propre  $P$  régi par la forme d’état suivante :

$$P = \left( \begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right) \quad (2-82)$$

La représentation d’état du correcteur initial est donnée par :

$$\begin{cases} \dot{\hat{x}} = (A - E C_2 - B_2 F) \hat{x} + E y \\ u = -F \hat{x} \end{cases} \quad (2-83)$$

où,  $E$  est le gain de l’observateur,  $F$  le gain du retour d’état et  $\hat{x}$  l’état estimé. Finalement le transfert du correcteur  $K_{init}$  est :

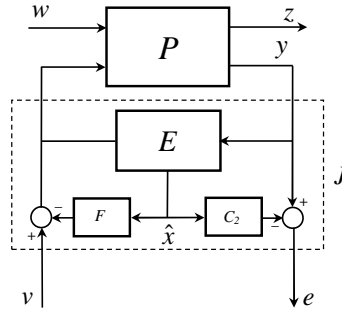
$$K_{init}(p) = -F (pI_n - A + B_2 F + E C_2)^{-1} E \quad (2-84)$$

Il s’agit maintenant de modifier le correcteur retour d’état-observateur afin de faire apparaître les signaux  $v$  et  $e$ , tout en assurant un transfert  $T_{v \rightarrow e}$  nul. Une solution est de prendre pour  $e$  l’erreur de

prédiction  $y - \hat{y}$  et en ajoutant  $v$  à la commande  $-F \hat{x}$  [Mac89]. Cette modification du correcteur initial est décrite par la figure 2.23.

Nous obtenons pour le correcteur modifié les équations d'état suivantes :

$$J : \begin{cases} \dot{\hat{x}} = (A - E C_2 - B_2 F) \hat{x} + E y + B_2 v \\ u = -F \hat{x} + v \\ e = y - C_2 \hat{x} \end{cases} \quad (2-85)$$



**Fig. 2.23: Modification du correcteur observateur-retour d'état**

Il ne reste plus qu'à connecter le paramètre de synthèse  $Q$  permettant de paramétrer l'ensemble des boucles fermées stables. L'approche présentée et utilisée est une méthode directe se basant sur une structure de correcteur de type retour d'état estimé. Il suffit d'écrire la représentation d'état de la boucle fermée avant la connexion de  $Q$ .

$$\begin{pmatrix} \dot{x} \\ \dot{\hat{x}} \end{pmatrix} = \begin{pmatrix} A & -B_2 F \\ E C_2 & A - E C_2 - B_2 F \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} + \begin{pmatrix} B_1 & B_2 \\ E D_{21} & B_2 \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} \quad (2-86)$$

$$\begin{pmatrix} z \\ e \end{pmatrix} = \begin{pmatrix} C_1 & -D_{12} F \\ C_2 & -C_2 \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} + \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} \quad (2-87)$$

On établit les transferts  $T_1$ ,  $T_2$  et  $T_3$ . On est donc capable d'exprimer l'ensemble des transferts en boucle fermée  $T_{w \rightarrow z}$  en fonction de  $Q$  puisque  $T_{w \rightarrow z} = T_1 + T_2 Q T_3$ . Une fois que l'on a obtenu  $Q$  par optimisation, il s'agit de revenir au correcteur  $K$ . Pour cela on cherche à déterminer  $K(Q)$ .

Soit la représentation d'état du paramètre  $Q$  :

$$\begin{cases} \dot{x}_Q = A_Q x_Q + B_Q e \\ v = C_Q x_Q + D_Q e \end{cases} \quad (2-88)$$

En éliminant les variables  $v$  et  $e$  à l'aide des équations (2-85), on obtient la représentation d'état du nouveau correcteur  $K$  en fonction du paramètre de synthèse  $Q$  et de  $K_{int}$  :

$$\begin{cases} \begin{pmatrix} \dot{\hat{x}} \\ \dot{\hat{x}}_o \end{pmatrix} = \begin{pmatrix} A - B_2 F - EC_2 - B_2 Q C_2 & -B_2 C_o \\ B_o C_2 & A_o \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{x}_o \end{pmatrix} + \begin{pmatrix} E + B_2 D_o \\ B_o \end{pmatrix} y \\ u = (-F - D_o C_2 \quad C_o) \begin{pmatrix} \hat{x} \\ \hat{x}_o \end{pmatrix} + D_o y \end{cases} \quad (2-89)$$

En résumé, l'étude des changements de variable qui permettent de paramétrer l'ensemble des boucles fermées atteignables par un système donné nous a montré qu'il existe un seul changement de variable, ne nécessitant aucune autre propriété que le caractère bien posé de la boucle fermée. Ce changement de variable correspond, dans le cas des systèmes linéaires stables, à la paramétrisation de Youla-Kučera qui peut être vue comme le transfert entrée/commande. Dans le cas des systèmes instables en boucle ouverte, ce changement de variable apparaît comme une restriction par une expression affine (2-66) de l'ensemble de ces transferts.

### **Expression des contraintes de gabarits fréquentiels en un problème d'optimisation**

Il s'agit de traiter le problème de compatibilité de plusieurs contraintes de type (2-60). Ceci se traduit par le problème de faisabilité suivant :

$$\begin{aligned} &\text{Existe-t-il } K \text{ tel que :} \\ &K \in \bigcap_i C_{i,\omega} \end{aligned} \quad (2-90)$$

L'ensemble  $C_{i,\omega} = \{ K / |H_i(j\omega)| \leq |H_i^{\max}(j\omega)|, \forall \omega \}$  représente l'ensemble des solutions faisables associé à la  $i^{\text{ième}}$  contrainte d'enveloppe fréquentielle. Ici,  $H_i$  représente le transfert en boucle fermée correspondant au couple d'entrée/sortie de même indice  $i$ . On écrit :  $H_i = H_{w_i \rightarrow z_i}$ .

Comme les spécifications fréquentielles sont souvent imposées sur le comportement en boucle fermée, leur expression en fonction du correcteur  $K$  fait intervenir la LFT de  $P$  et  $K$  (cf. équation (2-72)). Les problèmes ainsi obtenus sont difficiles à résoudre puisque ils dépendent non linéairement de  $K$ .

En utilisant le changement de variable linéarisant  $L$  (cf. équations (2-73) et (2-74)), nous reformulons le problème (2-90) en fonction de  $L$  :

$$\begin{aligned} &\text{Existe-t-il } L \text{ tel que :} \\ &L \in \bigcap_i C'_{i,\omega} \end{aligned} \quad (2-91)$$

où  $C'_{i,\omega} = \{ L / |(P_{11,i} + P_{12,i} L P_{21,i})(j\omega)| \leq |H_i^{\max}(j\omega)|, \forall \omega \}$ . Il est clair que cet ensemble de contrainte est convexe en fonction du paramètre  $L$ .

Notons que dans le cas où  $H_i^{\max}(j\omega)$  est inversible, les contraintes des ensembles  $C'_{i,\omega}$  s'écrivent en utilisant la norme  $H_\infty$  :

$$\left\| (P_{11,i} + P_{12,i} L P_{21,i}) (H_i^{\max})^{-1} \right\|_\infty \leq 1 \quad (2-92)$$

Afin d'assurer la stabilité de la boucle, il est judicieux de restreindre l'ensemble de recherche à celui des correcteurs stabilisants. Il s'agit donc de caractériser l'ensemble des paramètres  $L$  correspondant à des transferts de boucle stables et d'étudier les contraintes du problème (2-90) dans le nouvel espace ainsi défini. Pour ce faire, nous utilisons la fait que l'ensemble  $L_{\text{stable}} = \{ L \mid \text{Boucle fermée stable} \}$  est



convexe. La paramétrisation de Youla-Kučera introduite dans le paragraphe précédent permet de paramétrer pour un système linéaire invariant  $P$  donné, l'ensemble des correcteurs  $K$  stabilisants, à l'aide d'une matrice de transfert rationnelle  $Q$  propre et stable selon les relations (2-69).

Ainsi, tout transfert stable monovarié en boucle fermée peut s'écrire :

$$H_i = \mathbf{G}_i H_{w \rightarrow z} \mathbf{D}_i = \mathbf{G}_i T_1 \mathbf{D}_i + \mathbf{G}_i T_2 Q T_3 \mathbf{D}_i = T_{1,i} + T_{2,i} Q T_{3,i} \quad (2-93)$$

où  $\mathbf{G}_i$  et  $\mathbf{D}_i$  sont des matrices dites de sélection, elles sont constituées de 0 et une seule valeur 1 correspondant respectivement la  $i^{\text{ème}}$  sortie et la  $j^{\text{ème}}$  entrée.

Tout en conservant la convexité, on réécrit le problème de faisabilité (2-90) en fonction du paramètre de Youla-Kučera  $Q$  comme suit :

$$\begin{aligned} &\text{Existe-t-il } Q \text{ tel que :} \\ &Q \in \bigcap_i C_{i,\omega}'' \end{aligned} \quad (2-94)$$

$$\text{où } C_{i,\omega}'' = \left\{ Q / \left| (T_{1,i} + T_{2,i} Q T_{3,i})(j\omega) \right| \leq |H_i^{\max}(j\omega)|, \quad \forall \omega \right\}.$$

En introduisant un nouveau paramètre réel additif  $\gamma$ , chaque contrainte de gabarit fréquentiel s'écrit sous la forme :

$$\left\{ Q / \left\{ \begin{array}{l} \left| (T_{1,i} + T_{2,i} Q T_{3,i})(j\omega) \right| \leq |H_i^{\max}(j\omega)| + \gamma, \quad \forall \omega \\ \gamma \leq 0 \end{array} \right. \right\} \quad (2-95)$$

En conséquence, le problème de faisabilité se traduit par le problème d'optimisation suivant :

$$\begin{aligned} &\min \gamma \\ &\left\{ \left| (T_{1,i} + T_{2,i} Q T_{3,i})(j\omega) \right| \leq |H_i^{\max}(j\omega)| + \gamma, \quad \forall \omega \right\}_{i=1, \dots, N_\omega} \end{aligned} \quad (2-96)$$

où  $N_\omega$  est le nombre de contraintes semi-infinies.

Les paramètres d'optimisation sont donc  $\gamma$  et  $Q$ . Le problème d'optimisation formulé est convexe et les contraintes seront faisables si et seulement si  $\gamma_{opt} \leq 0$ .

Notons que l'utilisation des contraintes à base de la norme  $H_\infty$  (cf. équation (2-92)) ne change pas la nature du problème car toute norme est convexe. D'autre part, il est possible de formuler un problème d'optimisation équivalent à (2-96) en utilisant une variable réel multiplicative  $\delta$ . Dans ce cas, on écrit :

$$\begin{aligned} &\min \delta \\ &\left\{ \left| (T_{1,i} + T_{2,i} Q T_{3,i})(j\omega) \right| \leq \delta \cdot |H_i^{\max}(j\omega)|, \quad \forall \omega \right\}_{i=1, \dots, N_\omega} \end{aligned} \quad (2-97)$$

Le problème reste toujours convexe et sera faisable si et seulement si  $0 < \delta_{opt} \leq 1$ .

Comme dans le cas de gabarits temporels, la convexité du problème d'optimisation permet d'analyser les différentes contraintes et d'évaluer leur convenance dans le problème d'optimisation. On définit alors des contraintes souples et des contraintes dures.

**Extension aux critères de performance**

Dans le cas où les contraintes d’un problème d’optimisation sont montrées faisables, il est possible d’étudier la limite de performance que peut atteindre le système vis-à-vis d’un ou de plusieurs critères donnés. Dans ce cas, on formule des problèmes de type :

$$\min_{\text{contraintes dures}} J(Q) \tag{2-98}$$

où  $J(Q)$  est la fonction coût à minimiser sous les contraintes jugées dures après analyse.

Les critères temporels de type norme  $\ell^1$ ,  $\ell^2$  et  $\ell^\infty$  sont convexes et assurent ainsi la convexité du problème (2-98).

Il est aussi possible de prendre en compte simultanément plusieurs critères  $J_i(Q)$  en utilisant par exemple un critère global avec des coefficients de pondération positifs :

$$J(Q) = \sum_i \lambda_i J_i(Q) \quad \text{avec} \quad \lambda_i > 0 \tag{2-99}$$

Cette forme du critère global conserve la convexité du problème (2-98).

**Extension aux contraintes d’encadrement temporelles**

Outre les gabarits fréquentiels, l’approche par opérateurs permet de traduire les spécifications d’enveloppes temporelles. En effet, chaque signal de sortie  $z_i$  de la forme standard s’exprime en fonction du transfert  $H_i$  et de l’entrée  $w_i$ . On écrit :

$$\begin{aligned} z_i(t) &= \mathcal{L}^{-1}(z_i(p)) = \mathcal{L}^{-1}(w_i T_{1,i}) + \mathcal{L}^{-1}(w_i T_{2,i} Q T_{3,i}) \\ &= \mathcal{L}^{-1}(w_i T_{1,i}) + \mathcal{L}^{-1}(w_i T_{2,i}) * \mathcal{L}^{-1}(Q) * \mathcal{L}^{-1}(T_{3,i}) \end{aligned} \tag{2-100}$$

L’ensemble des paramètres  $Q$  correspondant aux sorties  $z_i$  appartenant à une enveloppe temporelle donnée est défini par :  $C_{i,t}^* = \{ Q / z_i^{\min}(t) \leq z_i(t) \leq z_i^{\max}(t), \forall t \}$ . On démontre facilement en utilisant la linéarité du produit de convolution et de la transformée de Laplace que cet ensemble est convexe.

Ainsi, la faisabilité des spécifications sous forme de gabarits est traduite par un problème d’optimisation global donné par :

$$\min_{\gamma} \left\{ \begin{aligned} & \left\{ |(T_{1,i} + T_{2,i} Q T_{3,i})(j\omega)| \leq |H_i^{\max}(j\omega)| + \gamma, \quad \forall \omega \right\}_{i=1, \dots, N_i} \\ & \left\{ z_i^{\min}(t) - \gamma \leq z_i(t) \leq z_i^{\max}(t) + \gamma, \quad \forall t \right\}_{i=1, \dots, N_i} \end{aligned} \right. \tag{2-101}$$

où  $N_i$  est le nombre de contraintes d’enveloppes temporelles.

**Mise en pratique de l’approche par opérateurs**

L’implémentation de l’approche par opérateurs nécessite une réponse à deux problèmes majeurs : la dimension infinie du correcteur et la dimension infinie de l’espace de fréquences (ou de pulsations).

Nous avons montré comment obtenir une forme affine en  $Q$  pour la LFT (2-79). Cette formulation affine en  $Q$  est une condition nécessaire pour avoir un problème d'optimisation convexe mais non suffisante. C'est pour cela que l'on choisit de paramétrer  $Q$  sous la forme suivante :

$$Q = \sum_{j=1}^q \psi_j Q_j \quad (2-102)$$

Les  $Q_j$  sont des filtres dont les pôles sont fixés a priori et les  $\psi_j$  les paramètres d'optimisation correspondant aux zéros des filtres. Ces filtres forment une base des matrices de transfert stables et rationnelles. Finalement la LFT (2-79) s'exprime de la façon suivante :

$$F_i(P, K) = T_1 + \sum_{i=1}^q \psi_i T_2 Q_i T_3 \quad (2-103)$$

On constate que les réponses fréquentielles et temporelles sont toujours affines par rapport aux paramètres de synthèses  $\psi_j$ . Ceci permet de conserver le caractère convexe du problème d'optimisation.

Comme l'ordre du correcteur dépend de l'ordre de la base, on choisira une base dont la taille est compatible avec les spécifications et avec le but recherché : synthèse d'un correcteur d'ordre raisonnable ou exploration de compromis sans limite a priori sur l'ordre du correcteur. Dans le cas de l'étude de la faisabilité d'un cahier des charges, pour décrire l'ensemble des fonctions de transfert, il serait nécessaire d'utiliser une base infinie pour couvrir tout l'espace. La difficulté réside dans le choix de la base de filtres et de son ordre. Choisir la base de filtres revient à fixer les pôles et à laisser les zéros libres qui serviront de paramètres d'optimisation. Il est important de remarquer que les pôles de la base de filtres sont des pôles qui vont se retrouver dans la boucle fermée par propriété de la  $Q$ -paramétrisation. Il est donc possible, en ayant une bonne expérience du procédé, de les choisir a priori.

Néanmoins pour avoir une approche plus méthodologique et donc plus systématique, il est intéressant de considérer une base de filtres. De nombreux travaux existent dans le domaine de l'identification pour la génération de ces bases de filtres. A titre d'exemple, nous citons la base des filtres à réponse impulsionnelle finie (FIR) qui est souvent utilisée dans les problèmes de commande :

$$Q_i = \left( \frac{p - \alpha}{p + \alpha} \right)^i, \quad \alpha \in \mathbb{R}^+ \quad (2-104)$$

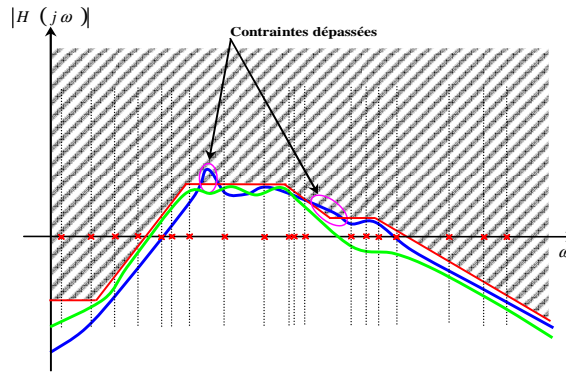
Comme notre objectif est d'étudier la faisabilité d'un cahier des charges, alors il est permis de prendre un ordre élevé pour la base. Ceci permet de donner un maximum de degrés de liberté au problème d'optimisation et d'être quasiment certain, si le problème n'a pas de solution, de l'infaisabilité du cahier des charges et non d'un mauvais choix de la base. En première approche, il est intéressant d'utiliser un FIR de taille importante. Si le problème est faisable, pour affiner la recherche, on peut utiliser une base qui introduit a priori différentes dynamiques pour couvrir le domaine fréquentiel plus rapidement, ce qui permet de réduire l'ordre du correcteur final.

Par ailleurs, il faut mentionner que l'objectif est de satisfaire chaque contrainte fréquentielle (2-95) sur un ensemble continu de pulsations  $\omega \in [\omega_1, \omega_n]$ . Ceci peut être traité par deux méthodes :

La première méthode consiste à transformer ces contraintes en une contrainte augmentée de type LMI en utilisant le lemme borné réel. Dans ce cas, on constate que la méthode devient très vite inexploitable car la taille du nouveau problème LMI croît rapidement avec le nombre de contraintes initiales ce qui rend la résolution de ce type de problème trop coûteuse en temps de calcul [Hba02a].

La deuxième méthode est plus pratique : elle consiste à utiliser un maillage fréquentiel afin d'évaluer les différentes contraintes (2-95). Autrement dit, on choisit un ensemble fini de pulsations  $\omega \in \{\omega_1, \dots, \omega_{n_j}\}$  pour résoudre le problème d'optimisation qui est initialement de taille infinie (cf. figure 2.24).

Ainsi, si on veut vérifier les contraintes du problème (2-95) sur un ensemble fini de pulsations  $\omega \in \{\omega_1, \dots, \omega_{n_j}\}$ , il sera nécessaire d'empiler ces contraintes pour chaque pulsation.



**Fig. 2.24** *Discrétisation des contraintes de gabarit fréquentiel*

Dans ce cas, si le nouveau problème est faisable, on est sûr a priori que le transfert  $H_j$  satisfait la contrainte fréquentielle mais seulement aux fréquences du maillage. On ne peut pas assurer qu'entre ces fréquences cette contrainte sur  $H_j$  est bien satisfaite. Il est alors nécessaire soit de procéder à une validation fréquentielle pour vérifier que  $H_j$  vérifie bien la contrainte sur un continuum de pulsations soit de trouver des conditions sur ce maillage afin d'assurer une équivalence entre la faisabilité des contraintes initiales et celle des contraintes discrétisées

Par ailleurs, nous avons présenté dans le cas des enveloppes temporelles dans le paragraphe 2.3.2.1, que pour une tolérance  $\varepsilon$  donnée, un échantillonnage fini permet de garantir a priori un dépassement des gabarits inférieur à  $\varepsilon$ . Pour garantir a priori une telle propriété dans le cas fréquentiel, il faudrait introduire des hypothèses de variations bornées sur les modules des transferts de la boucle fermée et sur les gabarits. Dans ce cas, on pourra énoncer de la même manière que pour un gabarit temporel, que pour une tolérance  $\varepsilon$  donnée, il existe une discrétisation  $\{\omega_1, \dots, \omega_{n_j}\}$  vérifiant :

$$\forall \varepsilon > 0, \quad \exists \omega_1 < \dots < \omega_{n_j} \quad \text{tel que} \quad (2-105)$$

$$\left( \forall \omega_k \in \{\omega_1, \dots, \omega_{n_j}\} \right) \Rightarrow \left( \forall \omega \in \mathfrak{R} \right)$$

$$\left( |H_i(\omega_k)| \leq |H_i^{\max}(\omega_k)| \right) \Rightarrow \left( |H_i(\omega)| \leq |H_i^{\max}(\omega)| + \varepsilon \right)$$

Toutefois, l'hypothèse de variation bornée dans le cas fréquentiel pour certaines applications, telle que les systèmes résonants, risque d'être une hypothèse très restrictive. Sa mise en œuvre nécessiterait l'utilisation d'un échantillonnage dense ce qui induit un très grand nombre de paramètres (les coefficients de projection sur la base des fonctions affines par morceau).

Dans la pratique, on opère d'une façon itérative, en choisissant une première discrétisation à partir de laquelle on résout le problème d'optimisation. Chaque contrainte fréquentielle du problème (2-94) est alors approchée par :

$$\left\{ |(T_{1,i} + T_{2,i} \mathcal{Q} T_{3,i})(j\omega_k)| \leq |H_i^{\max}(j\omega_k)| \right\}_{k=1,\dots,n_i} \quad (2-106)$$

A posteriori, on analyse la faisabilité de la contrainte fréquentielle pour la bande de pulsations continue considérée et on affine l'échantillonnage si la contrainte continue n'est pas vérifiée.

### 2.3.4.2. Etude de la faisabilité des spécifications fréquentielles par une approche paramétrique générale

Malgré le caractère exhaustif de l'approche par opérateurs (prise en compte de spécifications fréquentielles et temporelles), cette technique ne permet pas de traiter de nombreux cahiers des charges. En effet, dans cette approche, il n'est pas possible de considérer les spécifications fréquentielles de type marges de stabilité parce qu'elles ne peuvent s'exprimer exactement comme des contraintes convexes par rapport au paramètres d'optimisation. De même, pour les contraintes de structure qui ne garantissent pas en général la convexité du problème d'optimisation.

Etant donnée l'importance de ces spécifications qui sont souvent considérées, pour différentes raisons, comme des contraintes dures dans un cahier des charges industriel, nous proposons de généraliser l'approche précédente afin de définir des problèmes d'optimisation qui traduisent formellement le cahier des charges générique.

De la même manière que dans le cas temporel, on propose de formuler, directement cette fois, un problème général qui regroupe tous les types de spécifications fréquentielles d'un cahier des charges, y compris les contraintes liées à la structure du correcteur.

D'une façon générale, le cahier des charge d'un problème de commande comprend des contraintes sur les indicateurs de robustesse  $\alpha_\omega$ , sur le placement des valeurs propres  $\lambda$  ainsi que des contraintes de type gabarit. En conservant les mêmes notations que précédemment, ces demandes se transcrivent dans le domaine fréquentiel par le problème de faisabilité global suivant :

Existe – t – il  $\theta$  tel que :

$$\left\{ \begin{array}{l} \left\{ |H_i(\theta, j\omega)| \leq |H_i^{\max}(j\omega)|, \quad \forall \omega \right\}_{i=1,\dots,N_1} \\ \left\{ \alpha_{\omega,j}^{\min} \leq \alpha_{\omega,j}(\theta, \omega) \leq \alpha_{\omega,j}^{\max} \right\}_{j=1,\dots,N_2} \\ \left\{ \Lambda_l^{\min} \leq \Lambda_l(\lambda(\theta)) \leq \Lambda_l^{\max} \right\}_{l=1,\dots,N_3} \end{array} \right. \quad (2-107)$$

où :

- $H_i$  et  $H_i^{\max}$  sont, respectivement, un transfert en boucle ouverte ou fermée et son gabarit maximal
- $\alpha_{\omega,j}^{\min}$  et  $\alpha_{\omega,j}^{\max}$  sont les bornes minimale et maximale de l'indice de robustesse  $\alpha_\omega$ .
- $\Lambda_l^{\min}$  et  $\Lambda_l^{\max}$  sont les bornes minimale et maximale de la fonction de placement de pôles  $\Lambda_l(\lambda(\theta))$ .

Ce problème de faisabilité est transformé en un problème d'optimisation non linéaire sur les paramètres de décision  $(\lambda, \theta)$  :

$$\begin{aligned} & \min \gamma \\ & \left\{ \begin{aligned} & |H_i(j\omega, \theta)| \leq |H_i^{\max}(j\omega)| + \gamma, \quad \forall \omega \Big|_{i=1, \dots, N_1} \\ & \left\{ \alpha_{\omega, j}^{\min} - \gamma \leq \alpha_{\omega, j}(\theta, \omega) \leq \alpha_{\omega, j}^{\max} + \gamma \right\}_{j=1, \dots, N_2} \\ & \left\{ \Lambda_i^{\min} - \gamma \leq \Lambda_i(\lambda(\theta)) \leq \Lambda_i^{\max} + \gamma \right\}_{i=1, \dots, N_3} \end{aligned} \right. \end{aligned} \quad (2-108)$$

Le cahier des charges fréquentiel traduit par le problème d'optimisation (2-108) sera faisable si et seulement si la valeur de  $\gamma_{opt}$  est non positive.

Afin de juger de la faisabilité d'une contrainte particulièrement dure, des critères de performances de type  $H_2$  ou  $H_\infty$  peuvent être introduits afin de mesurer la limite des performances du système.

### *Mise en œuvre des problèmes d'optimisation fréquentiels formulés*

Comme dans l'approche par opérateurs, les contraintes semi infinies fréquentielles s'appliquent à des modules de transferts (ou des valeurs singulières) en boucle fermée du système et leur évaluation numérique est toujours approximative et ne se fait que sur une plage de pulsations bornée et à partir d'un nombre fini de points (cf. figure 2.24). On écrit alors :

$$m(\omega_k) = |H(j\omega_k)| + b_H(\omega_k), \quad k = 1, \dots, n_f \quad (2-109)$$

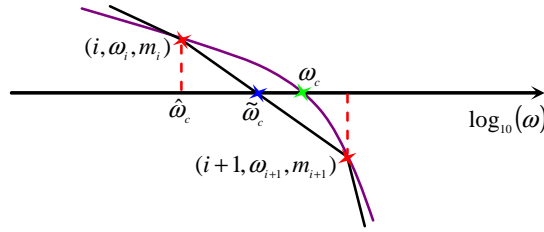
où  $m(\omega_i)$  et  $b_H(\omega_i)$  sont, respectivement, l'estimée du module de la fonction de transfert  $H(j\omega)$  et son erreur d'estimation due au gridding de taille  $n_f$ .

La validation des ces contraintes se fait alors à partir des pulsations choisies. Leur faisabilité est nécessaire mais elle ne suffit pas pour juger de la faisabilité des contraintes initiales continues (cf. figure 2.24).

Ces approximations engendrent d'autres complexités supplémentaires liées au calcul des indicateurs de robustesse fréquentiels tels que les marges de stabilité et la norme infinie. Le calcul des contraintes liées à ces indicateurs devient très subtil et peut peser lourdement lors de la résolution du problème d'optimisation en termes de précision et de temps de calcul. En effet, ces indicateurs fréquentiels sont tous définis implicitement sous la forme d'un problème algébrique.

Ainsi, la marge de phase nécessite la détermination de la pulsation de coupure qui nécessite elle-même la résolution d'un premier problème polynomial. La pulsation de coupure est alors issue soit d'une résolution formelle de l'équation polynomiale  $|L(j\omega)|=1$  dans les cas simples (jusqu'à l'ordre 8), soit numériquement en utilisant une recherche unidirectionnelle par optimisation ou par maillage. Le résultat obtenu sera utilisé pour évaluer l'estimée de la marge de phase.

Le schéma 2.25 décrit deux méthodes approximatives pour l'estimation de la pulsation de coupure.



**Fig. 2.25 Estimation d'une pulsation de coupure**

L'estimée  $\tilde{\omega}_c$  est obtenue par l'interpolation linéaire des deux pulsations  $\omega_i$  et  $\omega_{i+1}$ , elle est donnée par :

$$\tilde{\omega}_c = \omega_{i+1} - m_{i+1} \left( \frac{\omega_{i+1} - \omega_i}{m_{i+1} - m_i} \right) \text{ avec } m_{i,i+1} = |L(j\omega_{i,i+1})|$$

Le constat est le même pour la marge de retard qui dépend de l'estimée de la pulsation de coupure et de la marge de phase résultante ainsi que pour la marge de gain qui est similairement corrélée à l'estimée de la pulsation  $\omega_{-\pi}$ .

Concernant la marge de module, le facteur de résonance et la norme infinie, leur estimation requiert une recherche unidirectionnelle sur l'espace des pulsations et ceci est réalisable soit par optimisation ou par gridding.

Ainsi, les contraintes liées à ces indicateurs nécessitent une résolution numérique efficace (dichotomie, optimisation unidirectionnelle, gridding...), rapide et précise. Leurs temps de calcul dépendent fortement de la précision requise, qui pèse sur la qualité et le temps de calcul des problèmes d'optimisation. Pour les systèmes monovariabes, ces marges restent bien estimées, calculables en temps fini et avec un faible degré de complexité ce qui permet une formulation plus au moins exacte.

Quant au calcul des pôles, il est fait en utilisant soit des méthodes polynomiales soit des méthodes matricielles. Dans les deux cas le résultat est numérique et approximatif.

Les méthodes matricielles sont souvent plus robustes que les méthodes polynomiales et posent moins de problèmes numériques. Cependant une attention particulière doit être prise à l'égard des résultats obtenus qui sont souvent sensibles aux perturbations lorsqu'il s'agit d'un problème mal posé. Dans la suite de ce manuscrit, on montrera que cette situation est inévitable durant un processus d'optimisation.

Le problème de faisabilité global (2-107) peut être réécrit en sa version approchée comme suit :

$$\begin{aligned} & \min \gamma \\ & \left\{ \left\{ m_i(\omega_k, \theta) \leq |H_i^{\max}(j\omega_k)| + \gamma \right\}_{i=1, \dots, N_1} \right\} \\ & \left\{ \left\{ \alpha_{\omega, j}^{\min} - \gamma \leq \hat{\alpha}_{\omega, j}(\theta, \omega_k) \leq \alpha_{\omega, j}^{\max} + \gamma \right\}_{j=1, \dots, N_2} \right\}_{k=1, \dots, n_f} \\ & \left\{ \Lambda_l^{\min} - \gamma \leq \Lambda_l(\lambda(\theta)) \leq \Lambda_l^{\max} + \gamma \right\}_{l=1, \dots, N_3} \end{aligned} \quad (2-110)$$

## 2.4. Bilan et approche paramétrique générale pour l'analyse de cahiers des charges

Après avoir exposé les différentes techniques pour la formulation et l'analyse de faisabilité de cahiers des charges, il ressort deux grandes catégories d'approches qui dépendent des caractéristiques du problème d'optimisation formulé : les approches convexes accessibles pour les problèmes linéaires et pour des critères qui conservent la convexité et les approches non linéaires (non convexes).

La première approche convexe relève d'une vision signal type E/S et fait apparaître la nécessité d'exprimer des liens entre les signaux pour qu'ils puissent bien être le résultat d'un même correcteur. Ces relations permettent d'obtenir des conditions nécessaires et suffisantes pour prouver la faisabilité d'un cahier des charges et la construction d'une solution. Les problèmes sont formulés dans un espace paramétrique de dimension infini ; les paramètres étant des trajectoires de la boucle fermée. Pour la résolution, une approximation de dimension finie des trajectoires est nécessaire. Cette approche constitue un outil pour l'analyse des contraintes temporelles imposées aux trajectoires de commandes et de sorties. Elle permet d'étudier le comportement intrinsèque du système sans aucune contrainte structurelle sur la correction.

La seconde approche convexe relève d'une vision opérateur. Pour cette technique, il a été montré que l'ensemble des boucles fermées atteignables par un système donné peut être paramétré par le transfert entre l'entrée de référence et la commande du système. Dans le cas des systèmes à commander stables, cette paramétrisation coïncide avec la paramétrisation de Youla-Kučera. Dans le cas contraire, la paramétrisation de Youla-Kučera s'interprète comme une restriction par une expression affine de l'ensemble des transferts en boucle fermée. En utilisant cette paramétrisation, le problème de commande a été formulé en un problème d'optimisation convexe sur le paramètre linéarisant stabilisant  $Q$  avec la faculté de considérer simultanément plusieurs contraintes temporelles et fréquentielles lors de la synthèse.

La dimension de l'espace des paramètres étant infinie, le problème obtenu l'est aussi et nécessite une projection sur une base finie de l'opérateur  $Q$ . Afin de conserver l'importante propriété de convexité du problème d'optimisation, l'optimisation est faite seulement sur les numérateurs des transferts de la base choisie. L'évaluation des contraintes temporelles et fréquentielles ne peut se faire que sur un horizon fini, ainsi des discrétisations temporelles et fréquentielle sont nécessaires. La faisabilité des contraintes originales doit être vérifiée a posteriori par un processus essai/erreur.

Généralement, on évoque quatre applications majeures pour les formulations convexes :

- le dimensionnement de contraintes dans le cas où l'on se propose de définir un cahier des charges ;
- l'analyse de la faisabilité de certaines contraintes dans le cas où l'on étudie un cahier des charges a priori posé ;
- l'analyse des limites de performance atteignables par le système dans le cas où l'on s'autorise tous les degrés de liberté au niveau de la structure d'asservissement ;
- la synthèse de correcteurs optimaux dans le cadre, multicritères temporels et fréquentiels

L'implémentation de ces approches est fortement motivés par les avancés réalisées dans le domaine de l'optimisation convexe (programmation linéaire, quadratique et LMI), d'ailleurs c'est cette propriété



de convexité qui est exploitée par les méthodes numériques pour répondre à la question de faisabilité d'un cahier des charges.

Contrairement aux précédentes, l'approche par optimisation générale permet de regrouper toutes les spécifications génériques qu'elles soient temporelles ou fréquentielles en un seul problème de faisabilité sur les paramètres d'un correcteur à structure fixe. Dans le domaine temporel, cette formulation permet de considérer tout type de contraintes (enveloppe temporelle, indice de performance et critère intégral) ce qui permet même d'envisager son extension aux problèmes de commande non linéaires où les contraintes se résument à des contraintes sur les trajectoires de la boucle fermée seulement. Dans le domaine fréquentiel, les formulations sont similaires. Elles sont directes et ne nécessitent que la traduction des spécifications du cahier des charges en contraintes de type inégalité. Ces contraintes peuvent porter sur les différents transferts de la boucle fermée et/ou ouverte, sur les marges de stabilité ou sur les caractéristiques modales telles que les pôles et les zéros.

Les contraintes de structures qui ne peuvent être exprimées dans les approches convexes trouvent donc tout naturellement leur place en optimisation générale. Les domaines d'application couverts par une approche paramétrique générale sont plus larges que ceux déjà précités pour les approches convexes et ceci pour deux raisons :

- il est possible de prendre en compte la structure du correcteur dans tous les problèmes de dimensionnement, analyse et validation du cahier des charges ;
- même si les problèmes d'optimisation formulés sont non convexes et ne peuvent ainsi nous fournir une réponse définitive à la non faisabilité d'un cahier des charges (car il n'est pas possible de garantir qu'un optimum trouvé est global), l'utilisation de l'information a priori, comme le fait d'initialiser la recherche d'une solution autour de l'optimum et d'ajouter des contraintes une par une, permet de les rendre très efficaces lorsqu'il s'agit d'un problème complexe.

Cette utilisation de l'information a priori n'est pas incompatible ni trop différente des hypothèses faites lors de l'approximation des trajectoires par des fonctions affines par morceaux dans les approches convexes. D'ailleurs le résultat est le même car dans les deux cas on revient à un processus de synthèse essais/erreurs.

Dans la suite de ce manuscrit, l'accent sera mis particulièrement sur cette approche d'analyse par optimisation non linéaire. Le problème de faisabilité des contraintes sera étudié afin d'identifier les meilleures techniques d'optimisation susceptibles de le résoudre simplement et efficacement. Pour ce faire, une étude des différentes propriétés des critères et des contraintes est nécessaire. Principalement, elle consiste à analyser les propriétés de variations des critères et des contraintes par rapport au paramètre d'optimisation (étude de sensibilité des critères vis-à-vis des paramètres). Les résultats de cette étude définiront la démarche à suivre et les techniques d'optimisation à adopter par la suite.

### **2.4.1. Formulation du problème d'optimisation global de l'approche paramétrique générale**

Nous avons montré, au cours de ce chapitre, qu'un cahier des charges générique est souvent exprimé en utilisant l'opérateur logique "et". Par conséquent, le problème de faisabilité global peut s'énoncer par un problème de type :

$$\begin{aligned} & \text{Existe-t-il } \theta \text{ tel que :} \\ \theta \in \bigcap_i C_i \setminus C_i = \{ \theta \mid s_i^{\min}(\nu) \leq s_i(\theta, \nu) \leq s_i^{\max}(\nu), \forall \nu \} \end{aligned} \quad (2-111)$$

où  $\nu$  une variable représentant le temps  $t$  ou la pulsation  $\omega$ .

L'intersection des ensembles  $C_i$  est équivalente à la faisabilité simultanée de toutes les contraintes.

Afin de généraliser l'approche pour un cahier des charges comportant des spécifications temporelles et fréquentielles, il suffit d'agréger les problèmes de faisabilité temporelle (2-49) et fréquentielle (2-107) en un seul problème global. Ce problème se formule sous la forme générale suivante :

$$\begin{aligned} & \text{Existe-t-il } \theta \text{ tel que :} \\ & f(\theta, \nu) \leq 0 \end{aligned} \quad (2-112)$$

où  $f$  est le vecteur des fonctions contraintes où chaque contrainte d'encadrement sur la grandeur  $s(\theta, \nu)$  sous la forme  $s_i^{\min}(\nu) \leq s_i(\theta, \nu) \leq s_i^{\max}(\nu)$  se réécrit en deux inégalités de la forme  $f_i(\theta, \nu) = s_i(\theta, \nu) - s_i^{\max}(\nu) \leq 0$  et  $f_{i'}(\theta, \nu) = s_i^{\min}(\nu) - s_i(\theta, \nu) \leq 0$ .

Un problème d'optimisation équivalent à ce problème global de faisabilité est donnée par :

$$\begin{aligned} & \min \gamma \\ & f(\theta, \nu) - \gamma I_{N \times 1} \leq 0 \end{aligned} \quad (2-113)$$

où  $\gamma$  est un paramètre réel,  $I_{N \times 1}$  est un vecteur de même dimension que  $f$  ne contenant que des uns.

Le problème de faisabilité initial serait faisable si est seulement si le  $\gamma_{opt}$  calculé (pas forcément optimum global) est non positif.

## 2.4.2. Transformation en un problème d'optimisation non linéaire sans contraintes

Le problème de faisabilité (2-112) peut être posé, d'une manière équivalente, sous la forme d'un problème d'optimisation sans contraintes :

$$\min_{\theta} J(\theta) = \min_{\theta} \left\{ \max_{i=1, \dots, N} \left( \max_{\nu} (f_i(\theta, \nu)) \right) \right\} \quad (2-114)$$

où  $J$  est la fonction critère (ou coût) et  $f_i$  est la  $i^{\text{ième}}$  contrainte du vecteur  $f$ .

Ce problème sera faisable si et seulement si le critère optimal (ou sous optimal) calculé vérifie :  $J_{opt} \leq 0$ .

Une autre formulation équivalente en utilisant un critère à base de pénalisations exactes pondérées est donnée par :

$$\min_{\theta} J(\theta) = \min_{\theta} \left\{ \max_{i=1, \dots, N} \left( \eta_i \cdot \left| \max_{\nu} (f_i(\theta, \nu)) \right|_+ \right) \right\} \quad (2-115)$$

où  $|\cdot|_+ = \max(\cdot, 0)$  est la fonction partie positive et  $\eta_i$  est une pondération positive associée à la contrainte  $f_i$ .

Un exemple de pondération serait de choisir une normalisation. Ainsi pour  $f_i(\theta, \nu) = s_i(\theta, \nu) - s_{i,\max}(\nu) \leq 0$  et  $f_r(\theta, \nu) = s_{r,\min}(\nu) - s_r(\theta, \nu) \leq 0$ , on choisit  $\eta_i = 1/s_{i,\max}(\nu)$  et  $\eta_r = 1/s_{r,\min}(\nu)$ . Sous cette formulation, le problème (2-112) n'est faisable que si et seulement si  $J_{opt} = 0$ .

## 2.5. Sensibilité des critères et contraintes d'une approche paramétrique générale

Une analyse variationnelle des critères (2-114) et (2-115) montre qu'ils ne sont pas partout différentiables. Cette propriété est due d'abord à la structure de ces critères qui s'expriment en fonction de la fonction non différentiable "max", mais également à la dépendance de ces critères en des variables non différentiables.

En effet, les critères faisant intervenir des maximums ne sont pas différentiables quand le maximum est atteint en plusieurs points simultanément. Ainsi, même si la variation des contraintes  $f_i$  par rapport aux paramètres de réglage  $\theta$  est lisse, celle de leur intersection ne l'est pas lorsqu'il y a plus d'une contrainte active (cf. figure 2.26).

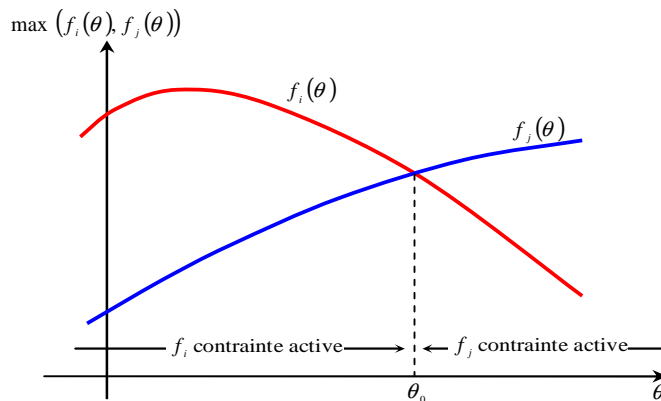


Fig. 2.26 Non différentiabilité de la fonction "max"

Ce type de non différentiabilité peut être atteint dans la plupart des cahiers des charges bien formulés. Il est particulièrement manifeste pour les exigences exprimées sous forme de contraintes d'encadrement.

Le meilleur exemple de ce type de contraintes est celui du problème de synthèse  $H_\infty$  standard où le but est de trouver un correcteur stabilisant la boucle fermée et qui permet de limiter le gain de certains transferts de la boucle ou même de leur imposer des gabarits fréquentiels en multipliant les entrées et les sorties par des filtres de pondération appropriés [Duc99].

Par exemple, en choisissant les couples E/S  $w^T = (r, b)$  et  $z^T = (\varepsilon, u)$  avec les fonctions de pondération  $W_1$  à la sortie du signal erreur  $\varepsilon$ ,  $W_2$  à la sortie du signal commande  $u$  et  $W_3$  à l'entrée du signal

perturbation  $b$  dans la forme standard, le problème  $H_\infty$  résultant consiste à déterminer un scalaire positif  $\gamma > 0$  et un correcteur  $K$  stabilisant le système bouclé et assurant la condition suivante :

$$\max(\|W_1 S\|_\infty, \| -W_1 S G W_3 \|_\infty, \|W_2 K S\|_\infty, \| -W_2 T W_3 \|_\infty) \leq \gamma \quad (2-116)$$

Ce problème présente la même structure que (2-113) et peut ainsi se formuler sous la même forme que le problème d'optimisation non différentiable (2-114).

Cependant, les méthodes analytiques (convexes) qui résolvent ce problème ne traitent qu'une version conservatrice où la condition précédente est remplacée par une contrainte plus dure mais assurant une résolution efficace du problème :

$$\left\| \begin{array}{cc} W_1 S & -W_1 S G W_3 \\ W_2 K S & -W_2 T W_3 \end{array} \right\|_\infty \leq \gamma \quad (2-117)$$

Le problème  $H_\infty$  est aussi un bon exemple pour illustrer la non différentiabilité des contraintes modales sur les pôles du système bouclé. En effet, la condition de stabilité dans ce problème peut être exprimée directement sur l'abscisse spectrale du système par l'inégalité stricte suivante :

$$f_i(\theta) = \max_j (\text{Re}(\lambda_j(\theta))) < 0 \quad (2-118)$$

Cette contrainte est non différentiable par rapport aux paramètres  $\theta$  lorsque le maximum des parties réelle des pôles est atteint simultanément pour plusieurs pôles de la boucle fermée et peut devenir parfois même non Lipschitz (discontinue) lorsque ce maximum est atteint pour un pôle multiple (d'ordre de multiplicité algébrique supérieur à un). Ce dernier cas est décrit via l'exemple suivant ; considérons le polynôme caractéristique dépendant de deux paramètres de réglage  $\alpha$  et  $\beta$  :

$$C(\lambda) = \lambda^5 + \alpha \lambda^4 - \alpha \lambda^3 + \beta \lambda^2 + \beta \lambda + \beta \quad (2-119)$$

Le tracé de l'abscisse spectrale de ce polynôme est donné par la figure 2.27:

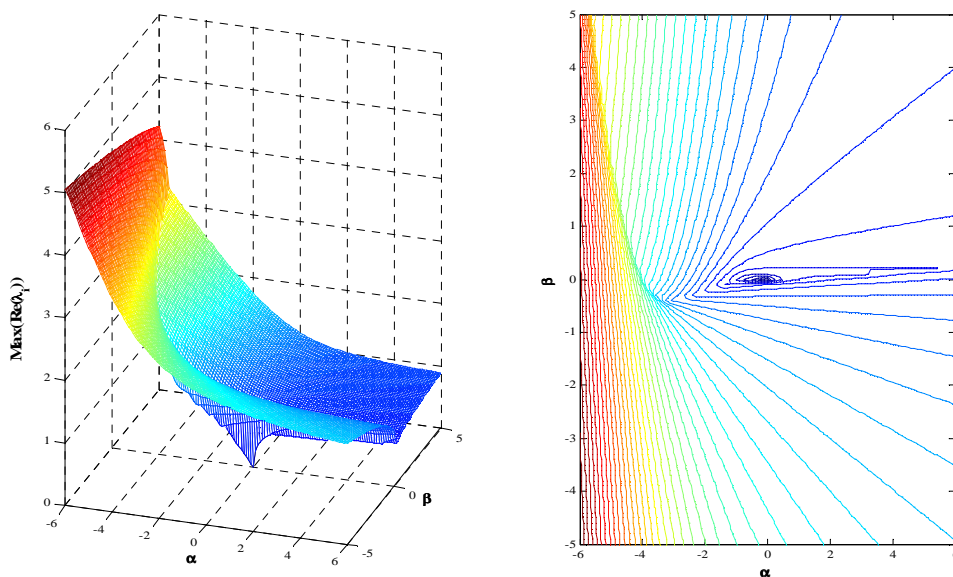


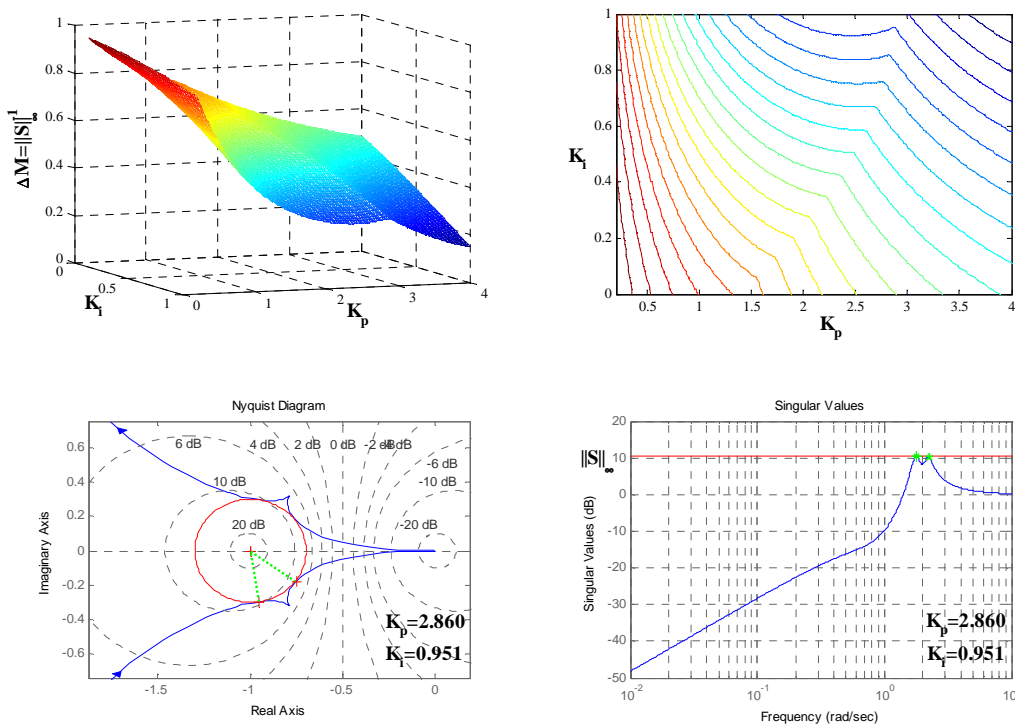
Fig. 2.27 Non différentiabilité de l'abscisse spectrale

La courbe présente une zone non différentiable autour de l'origine où le minimum est atteint. En réalité ce point minimum est non Lipschitz car il correspond à un seul pôle multiple d'ordre cinq. Par la suite, cet exemple servira comme un problème test pour les algorithmes d'optimisation qui seront développés dans ce document.

Nous présenterons également un exemple concernant la variation de la marge de module. Cette marge de robustesse, qui est définie à partir d'un problème d'optimisation, apparaît comme une contrainte structurellement non différentiable car elle fait intervenir la fonction "min" ou "max" dans sa définition. On propose d'illustrer cette propriété à travers une étude de la sensibilité de la marge de module par rapport aux paramètres d'un correcteur Proportionnel-Intégral  $K(p) = K_p(1 + K_i/p)$  et ceci pour l'asservissement d'un système à second ordre  $G$  comportant un mode souple :

$$G(p) = \frac{(p^2 + 0.4p + 4)}{(p^2 + p + 1)(p^2 + 0.4p + 4.2)} \tag{2-120}$$

L'analyse graphique du lieu Nyquist correspondant montre bien que le minimum est parfois atteint pour deux pulsations différentes à la fois, ce qui génère forcément des variations différentes de la marge de module de part et d'autre d'un jeu de paramètres  $(K_p, K_i)$ . Cela se traduit sur le module de la fonction de sensibilité  $S(p)$  par deux maximums simultanés dans le diagramme de Bode (cf. figure 2.28), ce qui confirme bien la difficulté du problème  $H_\infty$  standard où la variation du module de un ou plusieurs transferts en boucle fermée peut être non lisse dans (2-116) ce qui rend le problème encore une fois non lisse et donc plus compliqué à résoudre.



**Fig. 2.28 Sensibilité de la marge de module par rapport aux paramètres d'un correcteur PI**

La marge de phase dépend à son tour de la résolution d'un problème algébrique polynomial qui peut admettre plusieurs solutions  $\omega_i$ . Il est clair qu'une variation paramétrique dans le transfert en boucle ouverte peut bien faire croiser les phases correspondantes à ces pulsations et donc changer la variation

de la marge de phase du système. Plus compliqué encore, ces variations paramétriques peuvent même faire apparaître ou disparaître des racines réelles ce qui se traduit parfois par des discontinuités dans la marge de phase.

Une analyse similaire peut être faite pour la marge de gain qui dépend des pulsations  $\omega_{-\pi}^i$  solutions du problème algébrique non linéaire  $\angle L(j\omega) = 0$ . La marge de phase est définie par le plus petit des gains associés à ces pulsations  $\omega_{-\pi}^i$  d'où l'éventualité d'une variation non lisse. Ces situations ont souvent lieu dans le cas de présence de modes souples.

Dans le domaine temporel et mise à part les contraintes d'encadrement qui présentent la même propriété de non différentiabilité que les contraintes d'encadrement fréquentielles, on peut citer l'indicateur de précision dépassement qui revient à rechercher le maximum d'un signal. Ce dernier est donc non différentiable par construction et peut présenter des variations non lisses lorsque le maximum du signal est atteint simultanément pour plusieurs instants. Ceci se produit souvent pour des systèmes avec des modes dispersés.

On évoque pareillement, le temps d'établissement qui est défini à base d'une fonction "valeur absolue" et d'un problème de recherche unidirectionnel sur l'espace temps. Une simple étude de la sensibilité de cet indicateur de rapidité par rapport aux paramètres d'une fonction de transfert donnée, montre qu'il est même parfois non Lipschitz (discontinu). En effet, on considère la réponse indicielle du système de second ordre stable décrit par la fonction de transfert  $G(p)$  :

$$G(p) = \frac{1}{p^2/\omega_0^2 + 2\xi p/\omega_0 + 1} \tag{2-121}$$

Le dépassement de la réponse indicielle de ce système est donné par la relation :  $D = \exp(\xi\pi/\sqrt{1-\xi^2})$ . Ainsi, nous distinguons deux plages de variation du facteur d'amortissement délimitées par  $\xi_{crit} = -\log(\alpha)/\sqrt{\pi^2 + \log(\alpha)}$ . Dans le cas où  $\alpha = 5\%$ , ce facteur d'amortissement est donné par :  $\xi_{crit} = 0.690107$ .

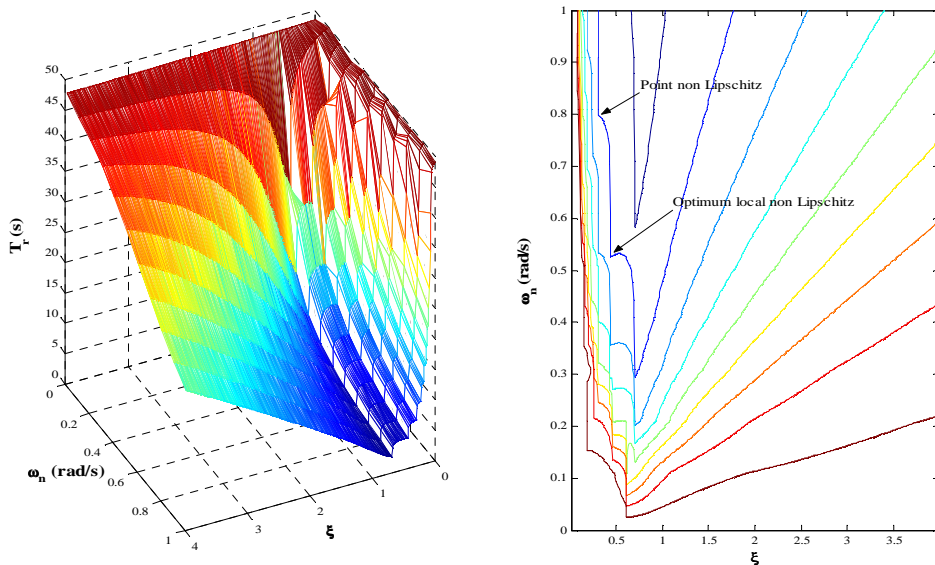


Fig. 2.29 Sensibilité du temps de réponse d'un système de second ordre

L'évaluation numérique du temps d'établissement à  $\alpha = 5\%$  par rapport à une variation du facteur d'amortissement  $\xi$  et de la pulsation propre  $\omega_n$  est donné par la figure 2.29.

Sur la plage de variation  $0 < \xi \leq \tilde{\xi}$ , nous remarquons que le temps de réponse présente des points et des minima locaux non lisses et même discontinus, ceci est dû à la variation des amplitudes des oscillations par rapport à l'encadrement de la réponse indicielle :  $0.95 \cdot s_{\infty} \leq s(t) \leq 1.05 \cdot s_{\infty}$ . Au-delà de cette limite ( $\xi > \tilde{\xi}$ ), le système est bien amorti et le dépassement est inférieure à 5% d'où l'allure régulière de la courbe.

Par conséquent, l'analyse variationnelle des différents critères liés aux cahiers des charges nous montre que plusieurs de ces critères ne sont pas partout différentiables. Toutefois, ils présentent la particularité d'être différentiables presque partout : l'espace des points non différentiables est généralement de mesure nulle (ensemble négligeable) [Bur02, Bur04].

Bien que de mesure nulle, on remarque que la rencontre de cet espace lors d'une optimisation est 'quasi certaine'. Ceci est dû au fait qu'un algorithme visant à minimiser le maximum absolu, aplatit la forme globale des courbes et aboutit spontanément à plusieurs maximums égaux. La convergence vers les espaces non différentiables est donc naturelle et ressemble beaucoup à la récupération des eaux pluviales dans une noue d'un toit en pente.

Dans le cadre d'une approche globale, cette propriété de non différentiabilité complexifie davantage la résolution des problèmes d'optimisation formulés, surtout que ces problèmes ne permettent pas en général de conserver les propriétés mathématiques sur lesquelles on s'appuie pour résoudre des problèmes classiques (différentiabilité ou convexité locale du problème).

Dans un sens différent de celui consacré par la théorie de la complexité, le problème d'optimisation global formulé peut être qualifié de difficile : critères et contraintes non-linéaires, non convexes, non connus analytiquement (implicites algébrique ou différentiel), bruités, non différentiables, de nature différente (temporelle, fréquentielle), dépendants implicitement de variables parfois corrélées, non définis partout, et présentant des plages de variations d'ordres de grandeur très différents.... Toutes ces spécificités font du problème de formulation des cahiers des charges génériques un problème complexe et doivent être tenues en compte pour le développement d'outils de calcul et d'optimisation efficaces et précis.

## 2.6. Conclusions

Nous avons développé dans ce chapitre deux classes d'approches qui permettent de traiter la problématique de faisabilité d'un cahier des charges.

La première se base sur la formulation en un problème d'optimisation convexe obtenu soit par une analyse de trajectoires E/S du système linéaire à commander soit par une paramétrisation convexe des transferts en boucle fermée. Malgré le caractère décisif de cette formulation qui permet de répondre à la non faisabilité d'un cahier de charges, la dimension infinie des problèmes formulés pose un problème de résolution. Néanmoins, sous les hypothèses d'un horizon fini et de variabilité limitée, des résultats sur la convergence des ensembles montrent l'existence d'une projection des courbes

temporelles ou fréquentielles sur une base finie. Celle-ci permet de conserver la convexité du problème approché mais ne garantit pas une équivalence entre la faisabilité des contraintes discrétisées et continues. Le processus essais/erreurs est alors incontournable pour décider de la non faisabilité du cahier des charges initial.

Contrairement à la formulation convexe, la deuxième classe d'approches se base sur une formulation paramétrique globale visant à formuler finement les spécifications du cahier des charges sans trop se préoccuper des éventuelles garanties de résolution des problèmes d'optimisation formulés. Les spécifications sont alors formulées soit stricto sensu en contraintes sur les différents indices de performance et de marges de robustesse comme dans l'approche de commande classique, soit par des contraintes d'encadrement temporelles ou fréquentielles. L'ensemble de ces contraintes est ensuite mis sous la forme d'un problème d'optimisation global traduisant la faisabilité totale du cahier des charges. Les problèmes obtenus sont non convexes et peuvent présenter des optima locaux qui ne permettent pas de juger de la non faisabilité des exigences du cahier des charges. Cependant, l'utilisation de l'information a priori dans une telle approche comme dans le cadre d'une retouche de correcteurs peut être très utile et fera de cette approche une efficace alternative pour la validation des cahiers des charges évolutifs.

Toutefois, l'étude de la complexité de ce type de problèmes d'optimisation montre qu'ils sont difficiles à résoudre parce qu'ils comprennent des critères et des contraintes qui font appel à des grandeurs temporelles, fréquentielles, implicites et explicites qui ne peuvent qu'être estimées numériquement ce qui influe grandement sur la processus d'optimisation et risque même de fausser ces résultats. Par ailleurs, la sensibilité paramétrique des critères et des contraintes formulés montre que la structure du problème global de faisabilité génère souvent des zones non différentiables dues principalement à l'utilisation de fonctions non différentiables comme la fonction "max" dans l'expression des critères globaux et qui est souvent présente dans la définition des indicateurs de performance et de robustesse.

Enfin et surtout, nous insistons sur le fait que les problèmes d'optimisation tels qu'ils sont formulés dans de nombreux problèmes d'ingénierie présentent la propriété d'être presque partout différentiables. Ce "presque" peut être trompeur et il ne faut garder en mémoire que les zones non différentiables sont souvent atteintes lors d'une optimisation. Cela montre la nécessité de développer des algorithmes d'optimisation adaptés : algorithmes permettant une résolution précise et efficace des problèmes presque partout différentiables.





# Chapitre 3

## Techniques d'optimisation non linéaire

Avant d'introduire les algorithmes que nous avons développés pour la résolution des problèmes d'optimisation formulés au deuxième chapitre, un exposé des concepts de base et des méthodes employés en optimisation non linéaire, est nécessaire. Ainsi, ce chapitre débute par un bref rappel de la problématique traitée et quelques définitions générales sur les problèmes d'optimisation.

Le concept de pénalisation est introduit afin de transformer le problème d'optimisation avec contraintes en un problème ou en une suite de problèmes d'optimisation sans contraintes. Cet outil, à la fois théorique et numérique, permet d'unifier l'étude des problèmes d'optimisation avec et sans contraintes dans un même formalisme et d'obtenir une solution de qualité suffisante rapidement sans avoir à entrer dans l'algorithmique sophistiquée de l'optimisation avec contraintes. Pour la résolution effective de problème, la pénalisation n'est pas une technique universelle parce qu'elle engendre souvent des complexités supplémentaires telles que l'apparition de nouveaux extremums et la non différentiabilité des nouveaux critères formulés. Dans le cas présent, ces inconvénients ne sont pas rédhibitoires car ils s'incluent naturellement dans notre approche qui concerne les problèmes presque partout différentiables. D'ailleurs, c'est pour cette principale raison que la technique de pénalisation est choisie pour la résolution de nos problèmes avec contraintes.

Les problèmes d'optimisation résultants sont non contraints et peuvent être résolus par différentes méthodes d'optimisation non linéaire. Dès lors, une étude bibliographique, en quatre sections, des classes liées aux algorithmes envisagées est présentée : les méthodes stochastiques, les méthodes de descente, les méthodes mixtes nées de l'association des ces deux classes de méthodes et les méthodes de recherche directe.

### 3.1. Contexte du travail

Pour résoudre les problèmes abordés au deuxième chapitre, il existe plusieurs grandes classes de méthodes qui regroupent chacune de nombreuses heuristiques. On se propose de faire une brève revue de ces méthodes dans la section suivante. Particulièrement, comme les problèmes traités ne sont pas partout différentiables et comportent des critères et des contraintes souvent mal connus ou bruités, l'accent est mis sur les techniques susceptibles de déjouer ces difficultés. Notre étude concerne principalement les méthodes déterministes de recherche directe et sur les méthodes de descente à base de gradient. Ces dernières construiront le noyau des algorithmes qui seront développés dans le chapitre suivant.

Les méthodes stochastiques sont brièvement évoquées. Les méthodes hybrides, quant à elles, ne sont pas abordées. Nous renvoyons le lecteur, par exemple, à [Sia03] pour une description didactique de ces deux classes de méthodes.

Bien entendu, dans l’exposé que nous faisons de toutes ces méthodes d’optimisation, comme dans l’exposé que nous ferons au chapitre 5 des stratégies de calcul du gradient, nous ne prétendons à aucune exhaustivité. Il s’agit simplement de situer les méthodes sur lesquelles notre choix se fixera ensuite dans l’ensemble des méthodes que nous avons pu trouver dans la littérature.

## 3.2. Principes généraux

Les problèmes d’optimisation se présentent dans de nombreux domaines de l’ingénieur, ainsi qu’en science et en économie, souvent après avoir conduit à leur terme les étapes de simulation. Il arrive souvent que ces problèmes se posent en dimension infinie, c’est-à-dire que l’on cherche une fonction optimale plutôt qu’un nombre fini de paramètres optimaux. Il faut alors passer par une phase de discrétisation (en temps, en fréquence, en espace, ...) pour retrouver le cadre qui est le nôtre et se ramener ainsi à un problème qui peut être résolu en un temps fini sur ordinateur. Les problèmes d’optimisation, formulés au deuxième chapitre, pour l’analyse de la faisabilité du cahier des charges par une approche entrées/sorties ou par opérateurs suivent une telle procédure de discrétisation.

Dans cette étude, une attention particulière sera portée aux algorithmes pouvant traiter les problèmes non partout différentiels qui sont souvent présents dans les problèmes de faisabilité de cahier des charges génériques (cf. sections 2.4 et 2.5).

### 3.2.1. Problème d’optimisation

Les méthodes numériques de l’optimisation ont principalement été développées après la seconde guerre mondiale, en parallèle avec l’amélioration des ordinateurs, et n’ont cessé depuis de s’enrichir. En optimisation non linéaire avec et sans contrainte, on peut ainsi distinguer plusieurs tendances : méthodes de pénalisation, méthode du lagrangien augmenté (1958), méthodes de quasi-Newton (1959), recherche directe (1965), méthodes newtoniennes SQP (Sequential Quadratic Programming) (1976), algorithmes de points intérieurs (1984), métaheuristiques (1986). Un courant ne remplace pas le précédent mais permet d’apporter de meilleures réponses à certaines classes de problèmes.

Le but de ce paragraphe est de rappeler les notions générales et de préciser la nomenclature et les notations qui seront utilisées le long de ce chapitre. Sauf mentions contraires, ces notations seront reconduites pour le reste du document.

De manière assez formelle, un problème d’optimisation se pose lorsque l’on cherche un point d’un ensemble  $X$  en lequel une fonction  $J$ , définie sur cet ensemble, prend une valeur minimale<sup>1</sup>. Nous l’écrivons de la manière suivante :

<sup>1</sup> Un problème d’optimisation peut aussi se présenter comme une maximisation (par exemple, maximiser les marges de robustesse, etc.), mais on peut toujours le transformer en un problème de minimisation, en changeant, par exemple,  $\max(J)$  en  $\min(-J)$ .

$$(P_x) : \begin{cases} \min J(x) \\ x \in X \end{cases} \quad (3-1)$$

La fonction  $J$  est le *critère* ou *fonction coût* du problème. L’ensemble  $X$  est l’*ensemble admissible* (ou *espace de recherche*) du problème. Il indique quel type de variables sont considérées : réelles, entières, mixtes (réelles et entières dans un même problème), discrètes, continues, bornées, etc.

Une solution de  $(P_x)$  est un point  $x^* \in X$  tel que  $J(x^*) \leq J(x)$  pour tout  $x \in X$ . On parle aussi de *minimum global*, par opposition à un *minimum local*  $x^* \in X$  qui ne vérifie  $J(x^*) \leq J(x)$  que pour des  $x \in X$  voisins de  $x^*$  :  $(J(x^*) \leq J(x) \forall x \in X / \|x - x^*\| \leq \varepsilon, \varepsilon > 0)$ . On dit que ces minima sont stricts si on a l’inégalité stricte  $J(x^*) < J(x)$  pour des  $x \in X$  (éventuellement voisins de  $x^*$ ) et différents de  $x^*$ .

La formulation de  $(P_x)$  est très générale. Dans cette revue nous nous restreindrons au cas où  $X$  est une partie de  $\mathfrak{R}^n$  décrite par des contraintes fonctionnelles d’égalité et d’inégalité :

$$(P_{EI}) : \begin{cases} \min J(x) \\ c_i(x) \leq 0, \quad i \in I \\ c_j(x) = 0, \quad j \in E \end{cases} \quad (3-2)$$

Les deux ensembles d’indices  $E$  et  $I$  sont supposés former une partition de  $\{1, \dots, m\}$ , c’est-à-dire que,  $E \cup I = \{1, \dots, m\}$  et  $E \cap I = \emptyset$  tandis que  $J : \mathfrak{R}^n \rightarrow \mathfrak{R}$  et les  $c_i : \mathfrak{R}^n \rightarrow \mathfrak{R}$  sont des fonctions différentiables.

Dans ce cas, l’ensemble admissible s’écrit :  $X = \{x \mid c_i(x) \leq 0, c_E(x) = 0\}$ . Le problème  $(P_{EI})$  est dit convexe, si  $f$  est convexe, si les composantes  $c_i$  sont convexes et si  $c_E$  est affine.

En face d’un problème d’optimisation comme  $(P_x)$ , plusieurs questions se posent. La première a trait à l’existence d’une solution et à l’unicité de celle-ci. L’unicité est une propriété appréciée par beaucoup d’algorithmes, mais est moins essentielle. Si le problème de l’existence est souvent difficile, il ne faut pas manquer de vérifier si le résultat standard suivant ne s’applique pas.

***Théorème 3.1 (Théorème de Weierstrass) :*** *Si  $X$  est un compact non vide et si  $J : X \rightarrow \mathfrak{R}$  est continue, alors le problème  $(P_x)$  a au moins une solution.*

Ce résultat a diverses extensions intéressantes. D’une part, on peut remplacer la continuité de  $J$  par sa semi-continuité inférieure. D’autre part, en dimension finie, on peut aussi remplacer  $X$  compact par  $X$  fermé et une hypothèse de croissance à l’infini de  $J$  :

$$\lim_{x \in \mathfrak{R}^n \text{ et } \|x\| \rightarrow \infty} (J(x)) = +\infty \quad (3-3)$$

En ce qui concerne l’unicité d’une solution, le résultat le plus simple, mais bien utile, est le suivant :

***Théorème 3.2 (Unicité de la solution) :*** *Si  $X$  est une partie convexe d’un espace vectoriel  $E$  et si  $J$  est strictement convexe sur  $X$ , alors  $(P_x)$  a au plus une solution.*

### 3.2.2. Conditions d’optimalité

Les deux résultats ci-dessus ne sont d’aucune aide pour trouver une solution de  $(P_{EI})$ . Ce qu’il nous faut, c’est une version analytique de l’optimalité, c’est-à-dire un ensemble d’équations et d’inéquations qui pourront être résolues par les algorithmes.

#### 3.2.2.1. Conditions d’optimalité en l’absence de contraintes

On sait qu’en l’absence de contraintes, une fonction  $J$  a sa dérivée qui s’annule en un minimum  $x^*$  :  $J'(x^*)=0$ , ce que l’on peut aussi écrire  $\nabla J(x^*)=0$ , où  $\nabla J(x)$  désigne le gradient de  $J$  en  $x$ . Cette condition nécessaire (CN) est attribué à Euler et Fermat et devient suffisante (CS) dans le cas où  $J$  est convexe.

**Théorème 3.3 (CS : cas convexe) :** Soit  $J : \mathfrak{R}^n \rightarrow \mathfrak{R}$  convexe et différentiable. Si  $\hat{x}$  vérifie  $\nabla J(\hat{x})=0$ , alors on a  $J(\hat{x}) \leq J(x)$  pour tout  $x \in \mathfrak{R}^n$ .

Lorsque la fonction n’est pas convexe, on ne peut donner qu’une condition nécessaire et suffisante d’optimalité locale. Dans le cas où  $J$  est deux fois différentiable, on peut alors énoncer le résultat suivant :

**Théorème 3.4 (CNS : cas non convexe) :** Soit  $J : \mathfrak{R}^n \rightarrow \mathfrak{R}$  deux fois différentiable. Si  $\hat{x}$  vérifie  $\nabla J(\hat{x})=0$  et  $\nabla^2 J(\hat{x}) > 0$ , alors  $\hat{x}$  est un minimum local de  $J$ .

où  $\nabla^2 J(x)$  désigne la matrice hessienne de  $J$  au point  $x$  et  $\nabla^2 J(x) > 0$  signifie sa définie positivité.

#### 3.2.2.2. Conditions d’optimalité en optimisation avec contraintes

Le but est de généraliser les conditions d’optimalité précédentes au problème  $(P_{EI})$ . Une observation préliminaire permettra de mieux comprendre ces conditions d’optimalité ; on dit qu’une contrainte d’inégalité ( $i \in I$ ), est active en  $x$  si  $c_i(x)=0$ . Seules les contraintes actives en une solution interviennent dans les conditions d’optimalité. Il sera donc utile de les désigner, ce qui se fera en introduisant les ensembles  $I_0 = I_0(x) = \{i \in I \mid c_i(x)=0\}$  et  $I_0^* = I_0(x^*)$ .

Dans ce qui suit, on se contentera de rappeler les conditions nécessaires d’optimalité du premier ordre attribuées à Karush, Kuhn et Tucker (KKT).

**Théorème 3.5 (CN : Karush, Kuhn et Tucker) :** Soit  $x^*$  un minimum local de  $(P_{EI})$ . Supposons que  $J$  et  $c_{E \cup I_0^*}$  soient dérivables en  $x^*$ . Alors, il existe  $\lambda^* \in \mathfrak{R}^m$  tel que l’on ait

$$(KKT) : \begin{cases} (a) : \nabla J(x^*) + c'(x^*)^T \lambda^* = 0 \\ (b) : c_E(x^*) = 0 \\ (c) : c_i(x^*) \leq 0 \\ (d) : (\lambda^*)_i \geq 0 \\ (e) : (\lambda^*)_i c_i(x^*) = 0 \end{cases} \quad (3-4)$$

avec  $c'(x)^T = [\nabla c_1(x), \dots, \nabla c_m(x)]$ .

L’identité (a) s’écrit aussi  $\nabla_x \ell(x^*, \lambda^*) = 0$ , où  $\ell$  est le lagrangien du problème ( $P_{EI}$ ), c’est-à-dire la fonction  $\ell: \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}$  définie en  $(x, \lambda)$  par :

$$\ell(x, \lambda) = J(x) + \lambda^T c(x) = J(x) + \sum_{i=1}^m \lambda_i c_i(x) \tag{3-5}$$

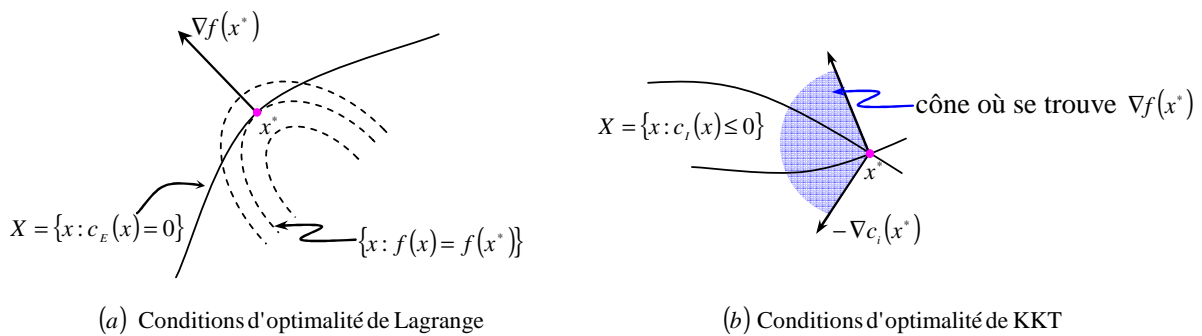
Un point  $x^*$  pour lequel il existe un multiplicateur de Lagrange  $\lambda^*$  tel que (3-4) ait lieu est appelé *stationnaire*.

Les conditions KKT traduisent, analytiquement, une condition géométrique de l’optimalité, qu’il est relativement aisée de retrouver à partir de (3-4).

Observons d’abord que l’on retrouve la condition  $\nabla J(\hat{x}) = 0$  s’il n’y a pas de contrainte.

S’il n’y a que des contraintes d’égalité ( $I = \phi$ ), elles expriment l’admissibilité de  $x^*$  (condition (b)) et le fait que  $\nabla f(x^*) \in I(c'(x^*))^\perp = \text{Ker}(c'(x^*))^\perp$  (condition(a)), c’est-à-dire que  $\nabla f(x^*)$  est orthogonal à l’espace tangent aux contraintes (le noyau  $\text{Ker}(c'(x^*))$ ) est l’ensemble des directions suivant lesquelles  $c$  ne varie pas au premier ordre en  $x^*$ ; il s’agit donc bien de l’espace tangent aux contraintes en  $x^*$ . Géométriquement (cf. figure 3.1(a)), cette condition exprime que le plan tangent à la variété  $\{x: f(x) = f(x^*)\}$  contient le plan tangent à  $X$  en  $x^*$ .

Supposons à présent qu’il n’y ait que des contraintes d’inégalité ( $E = \phi$ ), pour simplifier. On remarque qu’alors les multiplicateurs ont un signe (condition (d)) et que la condition (e) s’écrit aussi  $\lambda_i^* c_i(x^*) = 0$  pour tout  $i \in I$  (on utilise (c) et (d)), c’est-à-dire que soit  $\lambda_i^* = 0$  soit  $c_i(x^*) = 0$ , ou encore pour  $i \in I$  :  $c_i(x^*) < 0 \Rightarrow \lambda_i^* = 0$ . Ceci montre que les contraintes inactives en  $x^*$  n’interviennent pas dans les conditions de KKT, ce que l’on avait déjà signalé. On comprend pourquoi (e) porte le nom de *conditions de complémentarité*. A présent la condition (a) s’écrit :  $\nabla J(x^*) = \sum_{i \in I_0} \lambda_i^* (-\nabla c_i(x^*))$ .



**Fig. 3.1** *Interprétation géométrique des conditions d’optimalité*

Géométriquement (cf. figure 3.1(b)), cette identité exprime que le gradient  $\nabla J(x^*)$ , est dans le cône engendré par l’opposée des gradients des contraintes actives en  $x^*$ .

Pour les problèmes d’optimisation convexe, les conditions de KKT sont suffisantes pour entraîner l’optimalité globale.

**Théorème 3.6 (CS : cas convexe) :** *Si le problème ( $P_{EI}$ ) est convexe et si  $(x^*, \lambda^*)$  vérifie les conditions de Karush, Kuhn et Tucker (3-4) ( $J$  et  $c_{E \cup I_0}$  sont supposées dérivables en  $x^*$ ), alors  $x^*$  est un minimum global de ( $P_{EI}$ ).*

### 3.3. Prise en compte des contraintes par pénalisation

La pénalisation est un concept simple qui permet de transformer un problème d’optimisation avec contraintes en un problème ou en une suite de problèmes d’optimisation sans contrainte. C’est un outil à la fois théorique et numérique.

En analyse, l’approche par pénalisation est parfois utilisée pour étudier un problème d’optimisation dont les contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence. Si on a de la chance, ou si la pénalisation est bien choisie, des passages à la limite parfois délicats permettent d’obtenir des propriétés du problème original (l’existence de solutions par exemple). D’autre part, comme nous allons le montrer ci-dessous, la pénalisation est un outil permettant d’étudier les problèmes d’optimisation avec et sans contrainte identiquement. D’un point de vue numérique, cette transformation en problèmes sans contrainte permet d’utiliser des algorithmes d’optimisation sans contrainte pour obtenir la solution de problèmes dont l’ensemble admissible peut avoir une structure complexe. Cette approche est très souvent utilisée et permet d’obtenir rapidement des solutions de qualité suffisante. Bien que très adaptée aux problèmes que nous traitons, il ne faut pas oublier que ce n’est pas la seule approche possible. Elle a ses propres inconvénients : choix et réglage de la fonction de pénalité, non différentiabilité ou nécessité de minimiser une suite de fonctions.

Il existe de nombreuses approches, et nous nous limiterons ici à parler de celles qui permettent, par modification du critère, de se ramener à un problème sans contraintes. Elles ont l’avantage d’être compatibles avec les techniques d’optimisations sans contraintes qui forment l’essentiel de ce chapitre. Plus loin, dans la section 4.3.1.2, une deuxième approche à base de changement de variables est présentée. Cette dernière ne considère que des contraintes de bornes.

Les différentes techniques de pénalisation relèvent souvent du principe suivant. On remplace le problème  $(P_x)$  où  $X$  est une partie d’un espace vectoriel  $E$  (cf. équation 3-1), par un ou des problème(s)

$$(P_r): \begin{cases} \min f_r(x) \\ x \in E \end{cases} \quad (3-6)$$

où  $f_r(x)$  est obtenu en ajoutant à  $J(x)$  le terme  $r p(x)$  :

$$f_r(x) = J(x) + r p(x) \quad (3-7)$$

Le but de ce terme additionnel est de pénaliser la violation des contraintes (on parle alors de *pénalisation extérieure*, ce que nous verrons à la section 3.3.1.1) ou l’abord de la frontière du domaine admissible (on parle dans ce cas de *pénalisation intérieure*, ce que nous étudierons à la section 3.3.1.2). Le scalaire  $r > 0$  est appelé *le facteur de pénalisation*. Les propriétés de  $f_r$  vont dépendre de sa grandeur. On peut alors résoudre  $(P_r)$  par une méthode d’optimisation sans contrainte.

La question qui se pose immédiatement est de savoir si en résolvant  $(P_r)$  on résout  $(P_x)$ . Autrement dit, on cherche à savoir quand les ensembles de solutions de  $(P_x)$  et  $(P_r)$  coïncident. Cela va dépendre du choix de la fonction  $p(x)$  et de  $r$ . Par exemple, on pourrait choisir  $p(x)$  comme suit :

$$p(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{si } x \notin X \end{cases} \quad (3-8)$$

Il est clair que dans ce cas, les problèmes  $(P_x)$  et  $(P_r)$  sont identiques : ils ont les mêmes ensembles de solutions. Cette fonction de pénalisation est parfois utilisée dans la théorie, car elle permet de traiter en même temps les problèmes avec ou sans contrainte [Bon97]. Cependant, ce choix de  $p(x)$  n’est pas très utile en pratique car les méthodes classiques d’optimisation ne peuvent pas être utilisées sur des fonctions qui prennent la valeur  $+\infty$  dans des régions visitées lors des itérations de l’algorithme d’optimisation. Dans la pratique, ce type de pénalisation est parfois utilisé en remplaçant la valeur infinie par une valeur de critère suffisamment grande.

Nous allons donc, dans ce chapitre, introduire divers termes de pénalisation  $r p(x)$  et en étudier les propriétés théoriques et algorithmiques. La question précédente conduit à la notion de *pénalisation exacte*, à laquelle on a déjà fait référence dans le chapitre précédent.

**Définition 3.1 (Pénalisations exacte et inexacte) :** On dit qu’une fonction de pénalisation  $f_r$  associée au problème  $(P_x)$  est *exacte* si toute solution de  $(P_x)$  minimise  $f_r$  et qu’elle est *inexacte* dans le cas contraire (il y a des solutions de  $(P_x)$  qui ne minimisent pas  $f_r$ ).

Le terme solution est pris ici dans son sens générique et il faudra chaque fois préciser s’il s’agit de point stationnaire, de minimum local ou de minimum global.

La structure du problème  $(P_r)$ , dont le critère est la somme pondérée de deux fonctions, permet d’énoncer d’emblée une propriété très générale sur le comportement de chaque terme en une solution lorsque le poids  $r$  varie. Intuitivement, si  $r$  augmente, on attache plus d’importance à  $p(x)$  et il semble normal que, si  $\bar{x}_r$  est une solution,  $p(\bar{x}_r)$  décroisse. La proposition suivante énonce cela de façon rigoureuse. Le résultat est très général puisqu’il ne fait aucune hypothèse de convexité ou de différentiabilité; seule la structure de  $(P_r)$  intervient [Bon97, Gil07].

**Théorème 3.7 (Effet du facteur de pénalisation  $r$ ) :** On note  $x_r^*$  une solution de  $(P_r)$ . Alors, lorsque  $r > 0$  croît,  $p(x_r^*)$  décroît,  $J(x_r^*)$  croît et, si de plus  $p(\cdot) > 0$ ,  $f_r(x_r^*)$  croît.

Le même raisonnement montre que l’on a une croissance ou décroissance stricte des suites si  $x_r^*$  est l’unique minimum de  $(P_r)$  et si  $x_r^*$  change avec  $r$ .

### 3.3.1. Types de pénalisations

Mise à part les méthodes à base du Lagrangien (plus adaptées aux problèmes convexes) qui ne représentent pas le sujet de notre étude, on distingue trois grandes catégories de pénalisation en optimisation non linéaire :

#### 3.3.1.1. Pénalisations extérieures

Soit l’exemple de problème d’optimisation à une variable et une contrainte donné par :

$$\begin{cases} \min (1 + x + x^3 / 3) \\ x \geq 0 \end{cases} \quad (3-9)$$

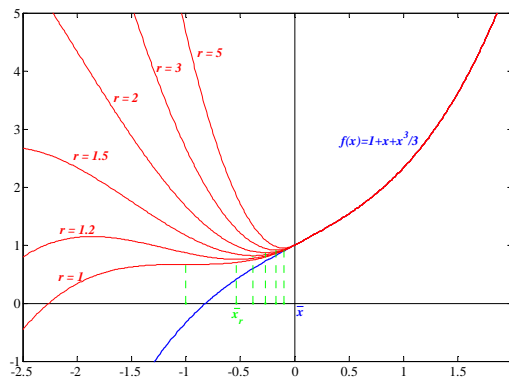


auquel on associe la fonction de pénalisation extérieure, dite quadratique :

$$f_r(x) = (1 + x + x^3 / 3) + r(|x|_-)^2 \tag{3-10}$$

où  $|\cdot|_-$  désigne la fonction partie négative définie par  $|y|_- = \max(-y, 0)$ .

L’effet de cette pénalisation peut s’observer à la figure 3.2 On voit que le terme  $r(|x|_-)^2$  ne joue un rôle qu’à l’extérieur du domaine admissible  $\mathfrak{R}^+$ . D’autre part, on observe que le minimum local de  $f(x)$  (il n’existe ici que si  $r > 1$ ) est extérieur au domaine admissible. Plus  $r$  est grand, plus le minimum se rapproche de la solution du problème qui est ici  $x^* = 0$ . Cependant, plus  $r$  est grand, plus le minimum local est accentué (la dérivée seconde de  $f_r(x)$  si elle existait serait élevée), ce qui pourra être une source de difficultés numériques.



**Fig. 3.2 Exemple de pénalisation quadratique extérieure**

Dans ce qui suit, nous allons montrer, de manière rigoureuse, que ce que nous venons d’observer sur cet exemple simple se produit pour une grande classe de fonctions de pénalisation.

On parle de pénalisation extérieure lorsque la fonction de pénalisation  $p(x)$  utilisée dans (3-7) vérifie les propriétés suivantes :

- (1)  $p(x)$  est continue sur  $\mathfrak{R}^n$
  - (2)  $p(x) \geq 0, \forall x \in \mathfrak{R}^n$
  - (3)  $p(x) = 0 \Leftrightarrow x \in X$
- (3-11)

Le qualificatif “extérieur” vient de la propriété (3), qui exprime que  $f_r(x)$  ne modifie  $J(x)$  qu’à l’extérieur de l’ensemble admissible. Le tableau 3.1 ci-dessous donne quelques exemples de fonctions  $p(x)$  satisfaisant (3-11).

Contraintes	Choix de $p(x)$
$c(x) = 0$	$p(x) = \ c(x)\ _2^2$
$x \geq 0$	$p(x) = \   x _- \ _2^2$
$c(x) \leq 0$	$p(x) = \   c(x) _+ \ _2^2$

**Tab. 3.1: Exemples de pénalisation extérieure.**

Ces fonctions  $p(x)$  font de  $f_r(x) = J(x) + r p(x)$  une fonction de pénalisation inexacte, puisque l’on trouve pour toute solution  $x^*$  de  $(P_r)$  :

$$\nabla f_r(x^*) = \nabla J(x^*) \tag{3-12}$$

Rien n’impose en effet, dans les conditions d’optimalité, que ce vecteur soit nul. Donc  $x^*$  n’est généralement pas solution de  $(P_r)$ .

Le théorème ci-dessous étudie le comportement des solutions  $x_r^*$  de  $(P_r)$ , lorsque  $r$  tend vers l’infini [Gil07]. Il donne des conditions pour que les solutions des problèmes pénalisés convergent vers une solution du problème original.

**Théorème 3.8 (Convergence de  $x_r^*$  vers  $x^*$ ) :** *Supposons que  $J$  soit continue et vérifie  $J(x) \rightarrow \infty$  quand  $\|x\| \rightarrow \infty$ . Soit  $X$  un fermé, non vide. On suppose que  $p(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}$  vérifie (3-11). Alors, on a :*

- (i)  $(P_r)$  a au moins une solution  $x_r^*$ ,
- (ii) la suite  $\{x_r^*\}_{r \rightarrow \infty}$  est bornée,
- (iii) tout point d’adhérence (ou de cumulation) de la suite  $\{x_r^*\}_{r \rightarrow \infty}$  est solution de  $(P_x)$ .

Les résultats de ce théorème ont une valeur indicative sur le comportement des minima globaux de  $f_r$ . Malheureusement, le même résultat ne tient plus pour les minima locaux. Par exemple, si  $c(x) = (x + 1)(2x^2 - 5x + 5)$ . Le problème

$$\begin{cases} \min J(x) = 0 \\ c(x) = 0 \end{cases} \tag{3-13}$$

consiste donc à chercher l’unique racine réelle  $x^* = -1$  de  $c(x)$ . En définissant le problème pénalisé par :

$$\min_{x \in \mathfrak{R}} (0 + c(x)^2) \tag{3-14}$$

Ce dernier présente un minimum local en  $x_r^* = 1$  quel que soit  $r > 0$ . On n’a donc pas la convergence de ces minima locaux vers  $x^* = -1$ .

Une autre limitation du théorème est de supposer que  $J$  est bornée inférieurement sur  $\mathfrak{R}^n$ . Si ce n’est pas le cas, il se peut que  $(P_x)$  ait une solution mais que  $(P_r)$  n’en ait pas. C’est le cas pour le problème suivant :

$$\begin{cases} \min (x^3) \\ x \geq 0 \end{cases} \tag{3-15}$$

Alors  $f_r(x) = x^3 + r(|x|_-)^2$  n’est pas bornée inférieurement. Dans de pareils cas, on peut rajouter un terme de pénalisation plus fort à l’infini ou introduire des bornes sur les variables.

Pour que la suite  $\{x_r^*\}$  s’approche d’une solution de  $(P_x)$ , il faut faire croître le facteur de pénalisation  $r$  (cf. théorème 3.8). Dans ces conditions, le théorème 3.7 nous apprend que la suite  $\{J(x_r^*)\}$  croît. Si la croissance de cette suite est stricte, les points  $x_r^*$  ne peuvent être qu’extérieurs à  $X$ , sinon les points d’adhérence de  $\{x_r^*\}$  ne pourraient pas être solutions de  $(P_x)$  (en effet, on aurait alors des points de  $X$

aussi proches que l’on veut d’une solution  $x^*$  en lesquels  $J$  prendrait une valeur strictement inférieure à  $J(x^*)$ , ce qui contredirait l’optimalité de  $x^*$ ).

La mise en œuvre de la pénalisation extérieure est tirée du théorème 3.8. Le schéma algorithmique suivant approche des solutions de  $(P_x)$  par pénalisation extérieure.

1. **initialisation** : choix de  $x^{(0)} \in \mathfrak{R}^n$ ,  $r_1 > 0$
2. **pour**  $k \geq 1$
3.     **trouver** un point stationnaire *approché*  $x^{(k)}$  de  $f_{r_k}$  (en partant de  $x^{(k-1)}$ ) vérifiant,  $\|\nabla f_{r_k}(x^{(k)})\| \leq \varepsilon_k$
4.     **si**  $x^{(k)}$  est satisfaisant (il vérifie approximativement les conditions de KKT), alors
5.         **arrêt**
6.     **sinon**
7.         **choisir**  $\varepsilon_{k+1} < \varepsilon_k$  et  $r_{k+1} > r_k$
8.     **finsi**
9. **finpour**

Le schéma algorithmique décrit ci-dessus est simple mais pose quelques problèmes de mise en œuvre. D’abord, il n’est pas aisé de donner un critère d’arrêt à l’étape 3 qui soit entièrement satisfaisant. On aimerait en effet ne pas passer trop de temps dans la minimisation de  $f_r$  si son minimum est éloigné de la solution du problème original, parce que  $r$  n’est pas assez grand. Dans le schéma ci-dessus, nous nous sommes contentés de donner un critère d’arrêt raisonnable, portant sur la condition d’optimalité du problème  $(P_r)$ , qui permet d’avoir un résultat de convergence (cf. théorème 3.9). Ensuite, il faut nécessairement augmenter le facteur de pénalisation  $r$  progressivement, car la minimisation de  $f_r$  pour  $r$  grand règle principalement le problème de l’admissibilité de  $x_r^*$ , sans ne plus voir  $f$ . Il faut donc que le point de départ de cette minimisation soit bon par rapport à la minimisation de  $f$ , ce qui ne peut être obtenu qu’en augmentant  $r$  progressivement. Ceci en fait un algorithme coûteux, puisqu’il faut nécessairement résoudre *une suite* de problèmes d’optimisation *non linéaires* (mais sans contrainte).

Enfin, lorsque  $r$  augmente le minimum est accentué, le problème  $(P_r)$  devient de plus en plus mal conditionné et donc de plus en plus difficile à résoudre numériquement. Ainsi, si la pénalisation extérieure conduit à un algorithme facile à implémenter, celui-ci est loin de n’avoir que des avantages. On ne peut pas gagner sur tous les plans en même temps.

Par la suite, nous serons souvent amenés à considérer le problème d’optimisation sous contraintes fonctionnelles  $(P_{El})$  (cf. équation 3-2).

Si  $v \in \mathfrak{R}^m$ , on note  $v^\# \in \mathfrak{R}^m$  le vecteur défini par :

$$v_i^\# = \begin{cases} v_i & \text{si } i \in E \\ |v_i|_+ & \text{si } i \in I \end{cases} \quad \text{avec } |v_i|_+ = \max(v_i, 0) \tag{3-16}$$

Les contraintes de  $(P_{El})$  s’écrivent alors simplement  $v^\#(x) = 0$ . L’application  $x \rightarrow v^\#(x)$  n’est pas différentiable en général ; on n’a donc fait que remplacer la difficulté liée à la présence de contraintes d’inégalité par une autre.

On associe au problème  $(P_{EI})$ , le problème pénalisé suivant :

$$\min_{x \in \mathfrak{R}^n} \left( f_r(x) = J(x) + \frac{r}{2} \|c^\#(x)\|^2 \right)$$

On rappelle que  $I_0(x^*) = \{i \in I \mid c_i(x^*) = 0\}$ . On notera aussi  $I_-(x^*) = \{i \in I \mid c_i(x^*) < 0\}$  et  $I_+(x^*) = \{i \in I \mid c_i(x^*) > 0\}$ .

Le théorème 3.9 ci-dessous donne les propriétés des points d’accumulation de la suite des points stationnaires approchés de  $f_r$  lorsque  $r \rightarrow +\infty$ . Il est donc complémentaire au théorème précédent qui ne s’intéresse qu’aux minima globaux [Gil07].

**Théorème 3.9 (Convergence de  $x_r^*$  vers  $x^*$ ) :** *On suppose que les fonctions  $J$  et  $c$  définissant  $(P_{EI})$  sont différentiables. Si  $x_r^*$  est un point stationnaire approché de la fonction de pénalisation  $f_r$ , définie par (3-7), dans le sens où  $\|\nabla f_r(x_r^*)\| \leq \varepsilon_r$ , si  $\varepsilon_r \rightarrow 0$  et  $x_r^* \rightarrow x^*$  lorsque  $r \rightarrow +\infty$ , où  $x^*$  est tel que les gradients  $\{\nabla c_i(x^*) : i \in E \cup I_0(x^*) \cup I_+(x^*)\}$  sont linéairement indépendants, alors*

- i)  $x^*$  est admissible pour  $(P_{EI})$ ,
- ii)  $\exists \lambda^* \in \mathfrak{R}^m$  tel que  $(x^*, \lambda^*)$  vérifie les conditions d’optimalité de KKT de  $(P_{EI})$ ,
- iii)  $r c^\#(x_r^*) \rightarrow \lambda^*$ , ( $\lambda^*$  multiplicateur de Lagrange optimal).

La qualification des contraintes au point optimal ( $\nabla c_i(x^*)$  linéairement indépendants) permet d’écartier la situation problématique rencontrée dans l’exemple (3-13)-(3-14). Dans ce cas, le théorème ci-dessus ne s’applique pas à une suite de minima de problèmes pénalisés qui convergerait vers 1, car  $c(1) > 0$  et  $c'(1) = 0$ .

### 3.3.1.2. Pénalisations intérieures

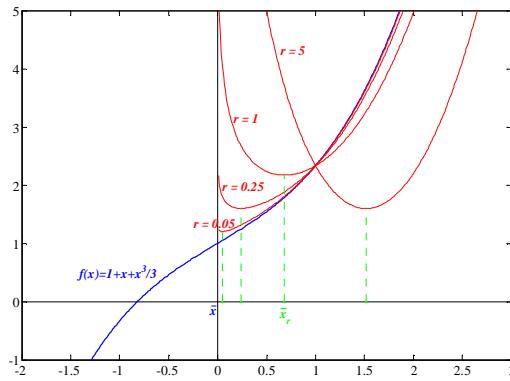
Dans certains problèmes, le fait que les itérés  $x_r$  générés par pénalisation extérieure ne soient pas admissibles peut être un inconvénient, par exemple, parce que  $J$  n’est pas définie à l’extérieur de  $X$ . On peut introduire des méthodes de pénalisation dans lesquelles les itérés  $x_r$  restent dans  $X$ . On parle alors de pénalisation intérieure. L’idée est d’utiliser un terme de pénalisation  $p$  qui tend vers l’infini lorsque  $x$  s’approche de la frontière  $\partial X$  de  $X$ .

Au problème simple (3-9), on pourra par exemple associer la fonction de pénalisation intérieure, dite logarithmique, suivante :

$$f_r(x) = 1 + x + x^3 / 3 - r \log(x) \tag{3-17}$$

L’effet de cette pénalisation peut s’observer à la figure 3.3.

On comprend que, dans cette section, il est nécessaire de supposer que l’intérieur  $X^0$  de  $X$  est non vide ( $X^0 \neq \emptyset$ ). Cette hypothèse exclut ainsi la possibilité de prendre en compte directement des contraintes d’égalité. Des artifices permettent toutefois de traiter de telles contraintes.



**Fig. 3.3 Exemple de pénalisation logarithmique intérieure**

Les termes de pénalisation  $p(x)$  considérés dans cette section satisfont les conditions suivantes :

- (1)  $p(x)$  est continue sur  $X^0$
  - (2)  $p(x) \geq 0, \forall x \in X^0$
  - (3)  $p(x) \rightarrow +\infty$  quand,  $x \in X^0 \rightarrow \partial X$
- (3-18)

On considère alors le problème de pénalisation  $(P_r)$  :

$$(P_r) \quad \min_{x \in X^0} f_r(x) \tag{3-19}$$

où  $f_r(x) = J(x) + r p(x)$ . La condition (3) de (3-18) crée une barrière au bord de l’ensemble admissible, si bien que  $f_r(x)$  porte parfois le nom de *fonction barrière*.

Le tableau 3.2 donne deux exemples de fonctions  $p(x)$  satisfaisant (3-18) lorsque l’ensemble admissible s’écrit  $X = \{x \in \mathfrak{R}^n : c(x) \leq 0\}$ , avec  $c(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ . On suppose que  $X^0 = \{x \in \mathfrak{R}^n : c(x) < 0\}$  n’est pas vide.

contraintes	choix de $p(x)$
$c(x) \leq 0$	$p(x) = -\sum_{i=1}^m 1/c_i(x)$
$c(x) \leq 0$	$p(x) = -\sum_{i=1}^m \log(-c_i(x))$

**Tab. 3.2: Exemples de pénalisation intérieure.**

Le premier exemple, celui de la fonction de *pénalisation intérieure inverse*, est dû à Carroll [Car61]. Le second exemple de terme pénalisant de ce tableau, dans lequel intervient le logarithme, porte aussi le nom de *pénalisation logarithmique*. Elle est due à Frisch [Fri55]. Cette pénalisation a connu une large utilisation et un renouveau avec *les algorithmes de points intérieurs*.

Le théorème suivant [Gil07] étudie la suite  $\{x_r^*\}$  des solutions des problèmes pénalisés. Contrairement à la pénalisation extérieure, il faut ici faire tendre  $r$  vers 0 (et non vers  $+\infty$ ), ce qui a pour effet de diminuer l’influence du terme pénalisant qui repousse les points vers l’intérieur de  $X$  et donc de permettre à  $x_r^*$  de se rapprocher de la frontière du domaine admissible, si cela est nécessaire.

**Théorème 3.10 (Convergence de  $x_r^*$  vers  $x^*$ ) :** *Supposons que  $J$  soit continue sur  $\mathfrak{R}^n$  et que l’ensemble admissible  $X$  soit d’intérieur non vide et vérifie  $X = \bar{X}^0$ . On suppose également que soit*

$X$  est borné, soit  $f(x) \rightarrow +\infty$  quand  $\|x\| \rightarrow +\infty$ . Alors, si la fonction de pénalisation  $p(x)$  vérifie (3-18), on a :

- (i)  $\forall r > 0$ ,  $(P_r)$  a au moins une solution  $x_r^*$ ,
- (ii) la suite  $\{x_r^*\}_{r \rightarrow 0}$  est bornée,
- (iii) tout point d’adhérence de  $\{x_r^*\}_{r \rightarrow 0}$  est solution de  $(P_X)$ .

Comme dans le cas de pénalisation extérieure, un schéma algorithmique de la mise en œuvre d’une pénalisation interne est proposé :

```

1.  initialisation : choix de  $x^{(0)} \in \mathfrak{R}^n$ ,  $r_1 > 0$ 
2.  pour  $k \geq 1$ 
3.      trouver un point stationnaire approché  $x^{(k)}$  de  $f_{r_k}$  (en partant de  $x^{(k-1)}$ ) vérifiant,  $\|\nabla f_{r_k}(x^{(k)})\| \leq \varepsilon_k$ 
4.      si  $x^{(k)}$  est satisfaisant (il vérifie approximativement les conditions de KKT), alors
5.          arrêt
6.      sinon
7.          choisir  $\varepsilon_{k+1} < \varepsilon_k$  et  $r_{k+1} < r_k$ 
8.      fin si
9.  fin pour
    
```

**3.3.1.3. Pénalisations exactes**

Si dans les termes de pénalisation de la table 3.1, on enlève les carrés, on obtient une fonction de pénalisation exacte (cf. définition 3.1) : si  $r$  est pris assez grand, les solutions locales de  $(P_x)$  sont solutions locales du problème pénalisé. Cela semble très intéressant, puisque l’on remplace un problème d’optimisation avec contraintes par un unique problème d’optimisation sans contrainte. On ne peut cependant pas gagner sur tous les plans : si la fonction de pénalisation est exacte, elle est aussi non différentiable et donc plus délicate à minimiser.

Le schéma algorithmique de la mise en œuvre d’une pénalisation exacte est le même que celui d’une pénalisation extérieure. D’ailleurs, c’est pour cette raison que ce type de pénalisation est parfois agencé comme une sous-classe des pénalisations extérieures.

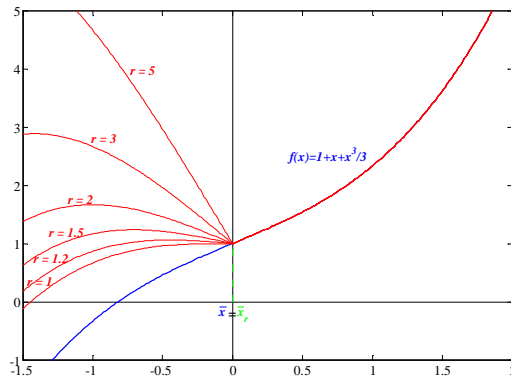
La fonction de pénalisation de Han est l’une des fonctions de pénalisation exactes les plus utilisées. Appliquée au problème d’optimisation avec contraintes explicites  $(P_{EI})$ , elle associe le critère pénalisé suivant :

$$f(x) = J(x) + r \|c^\#(x)\|_p \tag{3-20}$$

où  $\|\cdot\|_p$  représente l’opérateur norme d’ordre  $p$ .

La figure ci-dessous montre l’effet de ce type de pénalisation sur l’exemple (3-9).

On voit bien l’apparition d’un point d’adhésion optimal non différentiable qui coïncide avec le minimum global du problème.



**Fig. 3.4 Exemple de pénalisation exacte de Han**

C’est ce type de pénalisation qui a été déjà introduit dans le paragraphe 2.4.2 et qui sera reconduit dans la plus part de nos applications. D’une part, parce qu’il présente la propriété d’exactitude ce qui permet de considérer finement la faisabilité ou non des contraintes, et d’autre part, parce qu’il n’augmente guère la complexité du problème initial qui comporte souvent des fonctions non partout différentiables (cf. section 2.5).

Notons que les contraintes d’égalité étant peu fréquentes dans le problème d’évaluation du cahier des charges (cf. chapitre 1 et 2). Dans le cadre de notre étude, nous n’aborderons donc plus que des problèmes d’optimisation avec des contraintes d’inégalité qui permettent d’aller plus loin en analysant les limites de performance d’un système bouclé donné.

### 3.4. Méthodes globales versus méthodes locales

Dans tout ce manuscrit, le terme d’optimisation globale fait référence à la recherche des optima globaux de la fonction objectif, au sens défini dans le paragraphe 3.1.1. De ce point de vue, la méthode d’optimisation globale vise la détermination des optima globaux du problème, en évitant ses optima locaux. Cette dénomination présente néanmoins une certaine ambiguïté, car on rencontre souvent dans la littérature la dénomination de “méthode locale”, qui fait cette fois référence au mécanisme de recherche, lorsqu’il procède par voisins successifs. Ainsi, le recuit simulé est une méthode de recherche locale (la solution testée est une voisine de la solution courante), qui sera de notre point de vue une méthode d’optimisation globale (la méthode est en théorie capable de déterminer les optima globaux de la fonction objectif mais sans contrainte sur le temps de résolution).

Nous pouvons partager les méthodes en deux catégories. Celles qui permettent de déterminer un minimum local, ces méthodes sont appelées méthodes locales, et celles qui s’efforcent de déterminer un optimum global, ces méthodes sont appelées méthodes de recherche globale.

Les recherches locales partent usuellement d’un point initial  $x^{(0)}$  avec un pas initial  $\alpha^{(0)}$ . Ces paramètres vont conditionner la descente d’une des vallées de la fonction critère. De nombreuses méthodes locales existent. Les plus anciennes et les plus utilisées sont les méthodes où la direction de descente est déduite des dérivées de la fonction critère [Noc99] (méthode de la plus forte pente, méthode de Newton, méthodes de gradient conjugué, méthodes quasi-Newtoniennes, ...). Il existe d’autres méthodes déterministes locales qui ne nécessitent aucun calcul de dérivée. A titre d’exemples,

on site les méthodes directes, les méthodes par motifs, les méthodes de directions conjuguées (cf. section 3.5.4).

Les méthodes globales ont pour objectif d’atteindre un ou plusieurs optima globaux. Ils sont souvent liées à une stratégie de recherche qui peut être stochastique ou déterministe.

Ces méthodes ne s’excluent pas mutuellement. Afin d’améliorer les performances d’une recherche, plusieurs auteurs [Glo93] combinent les deux types d’algorithmes. Une recherche globale permet de bien explorer l’espace de recherche ; cette phase est appelée “diversification” ; et une recherche locale permet de bien exploiter une “zone prometteuse” (susceptible de contenir un minimum global), localisée lors de l’exploration du domaine de recherche ; cette phase est appelée “intensification”.

Cependant, il n’existe pas un algorithme optimal pour tous les problèmes (cf. “No Free Lunch Theorem (NFLT)” Wol97)), et la plupart des méthodes possèdent des paramètres à régler. Le choix de la méthode à utiliser et le réglage des paramètres restent liés au problème à optimiser. Le NFLT prouve que chaque méthode d’optimisation faisant des progrès sur une classe de fonctions régresse sur une autre classe (cf. figure 3.5).

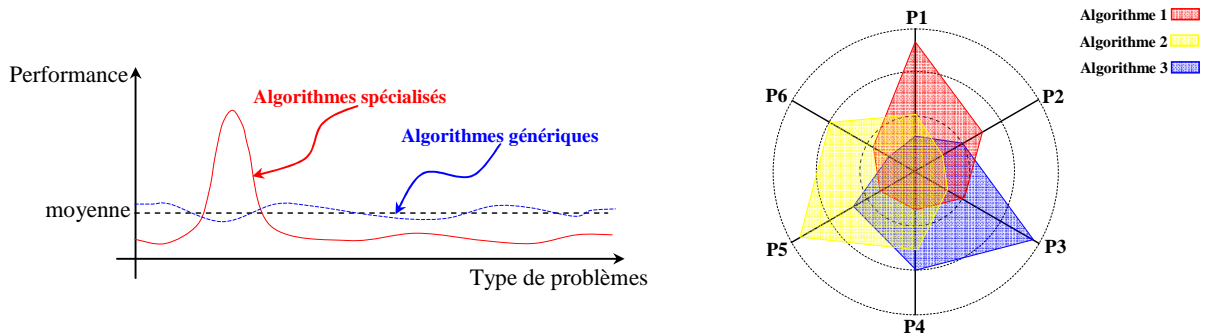


Fig. 3.5 Le principe du NFLT

### 3.5. Méthodes d’optimisation sans contraintes

Cette synthèse raisonnée décrit les principaux algorithmes de résolution des problèmes d’optimisation sans contrainte et en donne leur motivation.

Il s’agit de réaliser une revue sommaire des principales méthodes qui permettent de traiter le problème d’optimisation sans contraintes. Ce dernier est souvent donné sous la forme :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{3-21}$$

Cette étude aura pour objectif de choisir les méthodes d’optimisation qui peuvent être améliorées et adaptées à la problématique du cahier des charges ainsi que les techniques de calcul susceptibles à employer pour développer des algorithmes numériquement efficaces.

Dans le domaine de l’optimisation, on dispose a priori d’un grand nombre de méthodes. Une hiérarchie générale est impossible, l’efficacité d’une méthode donnée variant selon les problèmes



traités et les besoins de l'utilisateur. Cependant, en optimisation sans contraintes on peut distinguer quatre grandes classes.

### 3.5.1. Les méthodes stochastiques

Ce sont des méthodes où l'approche de l'optimum est en partie ou entièrement guidée par un processus stochastique. La plupart de ces algorithmes stochastiques sont itératifs et comportent trois éléments principaux : un mécanisme de perturbation, un critère d'acceptation et un critère d'arrêt. Ils sont appliqués à partir d'un ou plusieurs points de la fonction objectif, choisis aléatoirement.

Dans ce groupe de méthodes on peut avoir des recherches aléatoires pures, qui consistent à tirer un point, au hasard, à chaque itération. La fonction coût est évaluée en ce point, et s'il y a une amélioration, ce point et la fonction correspondante sont enregistrés, et le processus continue. Les recherches aléatoires peuvent aussi être associées aux recherches locales. Ainsi des points au hasard sont pris pour réinitialiser des recherches locales. Ces réinitialisations sont susceptibles de converger plusieurs fois vers les mêmes minima locaux. De plus, il n'y a pas de discrimination entre régions prometteuses ou non prometteuses. Ces procédures purement stochastiques, qui explorent et mémorisent le meilleur élément trouvé, ne sont pas efficaces et robustes [Sia03]. On leur préfère les méthodes dites pseudo aléatoires ; ces méthodes utilisent un choix aléatoire comme outil pour guider une exploration intelligente de l'espace des solutions. Dans cette catégorie on peut citer les méthodes énumératives [Per93], les méthodes de regroupement (clustering) [Tuw02] et les méthodes de descente généralisées [Tor89a].

Parmi les différentes méthodes stochastiques, nous allons uniquement nous intéresser aux heuristiques "modernes". Le mot "heuristique" vient du grec *heurein* (découvrir) et qualifie tout ce qui sert à la découverte, à l'invention et à la recherche. Pour l'algorithmique, les heuristiques sont des méthodes qui cherchent à approcher une solution optimale; on les appelle parfois méthodes approchées. Une heuristique peut être conçue pour résoudre un type de problème donné, ou bien être conçue comme une méthode générale, qui peut être adaptée à divers problèmes d'optimisation : dans le second cas, elle est désignée sous le terme de "métaheuristique".

#### 3.5.1.1. Métaheuristiques

Les métaheuristiques forment une famille d'algorithmes d'optimisation stochastiques visant à résoudre des problèmes d'optimisation difficile pour lesquels on ne connaît pas de méthode classique plus efficace. Ces méthodes stochastiques itératives progressent vers un optimum par échantillonnage de la fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents. En effet, ces algorithmes se veulent des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

Les métaheuristiques sont souvent inspirées par des systèmes naturels, qu’ils soient pris en physique (cas du recuit simulé), en biologie de l’évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l’optimisation par essais particuliers).

Les métaheuristiques ne nécessitent pas de connaissances particulières sur le problème optimisé pour fonctionner, le fait de pouvoir associer une (ou plusieurs) valeurs à une solution est la seule information nécessaire. En pratique, elles ne devraient être utilisées que sur des problèmes ne pouvant être optimisés par des méthodes mathématiques. Utilisées en lieu et place d’heuristiques spécialisées, elles montrent généralement de moins bonnes performances. De façon générale, on peut considérer que des problèmes présentant les caractéristiques suivantes sont assez propices à l’utilisation de métaheuristiques : nombreux optima locaux, discontinuités, contraintes fortes, non dérivabilité, temps de calcul de la fonction objectif prohibitif, solution approchée souhaitée...etc.

Les métaheuristiques sont souvent employées en optimisation combinatoire, mais on en rencontre également pour des problèmes continus ou mixtes (problèmes à variables discrètes et continues). Elles sont aussi d’un excellent recours pour les problèmes continus qui mettent en jeu une fonction critère présentant un nombre considérable de minima locaux. En effet, certaines métaheuristiques sont théoriquement convergentes sous certaines conditions. Il est alors garanti que l’optimum global sera trouvé en un temps fini, la probabilité de ce faire augmentant asymptotiquement avec le temps. Cette garantie revient à considérer que l’algorithme se comporte au pire comme une recherche aléatoire pure où la probabilité de tenter toutes les solutions tend vers 1. Cependant, les conditions nécessaires sont rarement vérifiées dans le cadre d’applications réelles. En pratique, la principale condition de convergence est de considérer que l’algorithme est ergodique : qu’il peut atteindre n’importe quelle solution à chaque itération, mais on se satisfait souvent d’une quasi-ergodicité : si la métaheuristique peut atteindre n’importe quelle solution en un nombre fini d’itérations.

### 3.5.1.2. Principe

D’une manière générale, les métaheuristiques s’articulent autour des notions suivantes : *le voisinage*, *la diversification* (exploration), *l’intensification* (exploitation), *la mémoire* et *l’apprentissage*.

- *Le voisinage* d’une solution est un sous-ensemble de solutions qu’il est possible d’atteindre par une série de transformations données.
- *La diversification* désigne les processus visant à récolter de l’information sur le problème optimisé.
- *L’intensification* vise à utiliser l’information déjà récoltée pour définir et parcourir les zones intéressantes de l’espace de recherche.
- *La mémoire* est le support de l’apprentissage, qui permet à l’algorithme de ne tenir compte que des zones où l’optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les notions d’intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. Les deux notions ne sont donc pas contradictoires, mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l’un et l’autre des aspects.

Les métaheuristiques progressent ainsi de façon itérative, en alternant des phases d’intensification, de diversification et d’apprentissage, ou en mêlant ces notions de façon plus étroites. L’état de départ est souvent choisi aléatoirement, l’algorithme se déroulant ensuite jusqu’à ce qu’un critère d’arrêt soit atteint.

### 3.5.1.3. Classification

Les métaheuristiques peuvent être classées selon plusieurs critères : approche à base de parcours ou à base de population, l’emploi ou non de la mémoire, fonction objectif statique ou dynamique, type d’échantillonnage, algorithme hybride ou non, algorithme évolutif ou non. Selon ce dernier critère, les métaheuristiques les plus connues sont classées en deux classes :

Les algorithmes évolutifs : les stratégies d’évolution, les algorithmes génétiques, les algorithmes à évolution différentielle, les algorithmes à estimation de distribution, les systèmes immunitaires artificiels, la recombinaison de chemin (Path relinking).

Les algorithmes non évolutifs : le recuit simulé, les algorithmes de colonies de fourmis, Les algorithmes d’optimisation par essais particuliers, la recherche avec tabous, la méthode GRASP (Greedy Randomized Adaptive Search Procedures)

La recherche dans le domaine étant très active, il est impossible de produire une liste exhaustive des différentes métaheuristiques d’optimisation. La littérature spécialisée montre un grand nombre de variantes et d’hybridations entre méthodes, particulièrement dans le cas des algorithmes évolutionnaires.

### 3.5.1.4. Exemples de métaheuristiques

On distingue principalement quatre grands types d’algorithmes relevant de ces méthodes.

Tout d’abord, les algorithmes génétiques tentent de simuler le processus d’évolution naturelle dans un environnement hostile. Chaque solution du problème, ou individu, est codée par une chaîne de bits finie à laquelle est associée une “fitness” égale au critère en cette solution. Ensuite, des populations d’individus sont générées itérativement en appliquant des processus de sélection, de croisement et de mutation qui se basent sur la “fitness” des individus [Gol89].

L’algorithme de recuit simulé (ou “simulated annealing”) est né d’une analogie thermodynamique avec le refroidissement d’un métal. Le refroidissement lent et régulier d’un métal permet aux atomes de se stabiliser peu à peu dans une position d’énergie minimale en dépit du nombre immense de configurations que peuvent prendre ces atomes. N. Metropolis a proposé d’introduire un paramètre ou température qui diminue au cours de l’optimisation [Met53]. Au début du processus, la température élevée autorise les transitions vers un état d’énergie plus élevée. Au cours du processus, la température diminuant, la transition vers un état d’énergie plus élevée devient de plus en plus improbable. Pratiquement, pour un problème de minimisation, la stratégie des algorithmes de recuit consiste, en effectuant une exploration aléatoire de l’espace d’état, à favoriser les descentes, mais sans interdire tout à fait les remontées. Plus précisément, on se donne une chaîne de Markov sur l’espace d’état, et on accepte ou on refuse une transition avec une probabilité 1 si la fonction coût  $f$  décroît, et une probabilité égale à  $\exp(-\Delta f/T)$  si le coût  $f$  croît. Ici  $T$  est un paramètre, appelé température par

analogie. Il est clair que plus la température est grande, plus seront facilitées les transitions ascendantes. En revanche, à la limite  $T = 0$ , on obtient un algorithme de descente. Tout au long de l’algorithme, on va donc faire décroître  $T$ , ni trop vite pour ne pas rester bloqué autour d’un minimum local, ni trop lentement si on veut avoir un résultat en un temps raisonnable.

La méthode Tabou est une procédure heuristique moins populaire que les deux précédentes. Elle suppose qu’on puisse définir un voisinage de solutions pour chaque solution. A chaque itération, la procédure se déplace vers la solution du voisinage  $N(x_k)$  de  $x_k$  qui diminue au mieux la fonction critère. Alors que la plupart des méthodes d’exploration ne gardent comme information que la valeur minimale de la fonction critère calculée jusqu’alors, cette procédure garde en mémoire les solutions rencontrées ou plus généralement l’itinéraire effectué lors des dernières itérations dans une liste dite tabou [Glo93]. La liste tabou a pour rôle d’interdire le choix des déplacements (respectivement des solutions) à ceux (respectivement celles) qui ramènent à une solution visitée précédemment. Ainsi, l’algorithme peut s’échapper des minima locaux. Au cours de la procédure, la liste tabou ainsi que les mouvements admissibles pour la solution sont régis par des processus d’intensification, de diversification et d’aspiration [Roc95].

La technique appelée GRASP (ou Greedy Randomized Adaptive Search Procedures) est une heuristique très simple utilisée surtout pour les problèmes combinatoires. Il s’agit d’un processus itératif, qui à chaque itération répète les mêmes phases. La première phase est une phase de construction pendant laquelle une solution réalisable est exhibée. La seconde consiste à réaliser une optimisation locale à partir de la solution construite. La meilleure solution est gardée en mémoire jusqu’à ce que le processus soit arrêté (stagnation, nombre maximal d’itérations atteint) [Res98].

Pour mieux tirer parti des avantages de chaque méthode, des variantes hybrides ont été développées ainsi que des algorithmes parallélisés [Par95]. Cependant, l’inconvénient majeur de ces méthodes stochastiques vient du fait qu’elles requièrent beaucoup de simulations. Dans le cas où l’évaluation de la fonction coût est coûteuse, de telles méthodes sont donc inutilisables.

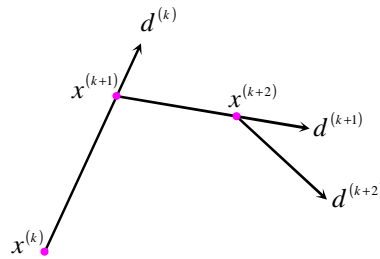
### 3.5.2. Les méthodes de descente

Dans cette section, nous rappelons les principales notions liées aux algorithmes à direction de descente. Ces algorithmes, qui exploitent l’information locale du critère afin d’atteindre son minimum, présentent des avantages et des inconvénients disparates que nous essayons d’analyser et de mettre en avant pour en tirer profits et les utiliser au mieux dans le cadre de l’analyse du cahier des charges.

On considère de nouveau le problème d’optimisation sans contraintes (3-21) et on s’intéresse à la classe d’algorithmes qui est fondée sur la notion de direction de descente.

**Définition 3.2 (Direction de descente)** On dit qu’un vecteur  $d$  de  $\mathfrak{R}^n$  est une direction de descente pour une fonction  $f$ , de  $n$  variables, au point  $x$  si :  $\forall s > 0, \exists \alpha \in ]0, s[ , f(x + \alpha d) < f(x)$ .

Ainsi  $f$  décroît strictement dans la direction  $d$ . De telles directions sont intéressantes en optimisation car, pour faire décroître  $f$ , il suffit de faire un déplacement le long de  $d$ . Les méthodes à directions de descente utilisent cette idée pour minimiser une fonction. Elles construisent la suite des itérés  $\{x^{(k)}\}_{k \geq 1}$  approchant une solution  $x^*$  de (3-21) par la récurrence  $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$  (cf. figure 3.6). Dans ce cas,  $\alpha^{(k)} > 0$  est appelé le pas et  $d^{(k)}$  est une direction de descente de  $f$  en  $x^{(k)}$ .



**Fig. 3.6** Enchaînement des itérés pour un algorithme de descente

Partant d'un point  $x^{(0)}$  qui lui sera initialement passé pour argument, un algorithme de descente actualise un point courant  $x$  de façon à réduire, à chaque étape, la valeur du critère à minimiser. Le schéma général est le suivant :

1. **poser**  $k = 0$
2. **choisir**  $x^{(0)} \in \text{dom } f$
3. **tant que** (le critère d'arrêt n'est pas satisfait) **faire** :
4.       calculer une direction de descente  $d^{(k)}$
5.       calculer le pas  $\alpha^{(k)} > 0$  tel que  $f(x^{(k)} + \alpha^{(k)}d^{(k)}) < f(x^{(k)})$  // recherche unidirectionnelle
6.       mise à jour  $x^{(k+1)} = x^{(k)} + \alpha^{(k)}d^{(k)}$ ,  $k = k + 1$
7. **fin tant que**

Pour définir une méthode à directions de descente, il faut donc spécifier trois arguments :

- La stratégie de choix des directions de descente successives  $d^{(k)}$  ; la manière de procéder donne le nom l'algorithme, la section qui suit est dédiée au détail de ces méthodes.
- La stratégie de choix du pas  $\alpha^{(k)}$  qui sera effectué, à chaque étape, dans la direction choisie ; c'est ce que l'on appelle la recherche linéaire ou unidirectionnelle; la section 3.5.2.2 sera consacrée à la description des principales techniques de recherche linéaire.
- Le choix de la condition d'arrêt de l'algorithme.

Dans les problèmes sans contrainte, le test d'arrêt de l'étape 3 porte sur la petitesse du gradient :  $\nabla f(x^{(k)}) \approx 0$ . C'est en effet ce que suggère la condition nécessaire d'optimalité du premier ordre  $\nabla f(x^*) = 0$ . Comme  $x^{(k)}$  n'est jamais exactement égal à  $x^*$ , ce test ne pourra marcher que si  $\nabla f(x^{(k)})$  est faible en norme pour  $x$  voisin de  $x^*$ , ce qui revient pratiquement à supposer que  $f$  est de classe  $C^1$ . Par ailleurs, un tel test d'arrêt suggère qu'un algorithme à directions de descente ne peut pas trouver mieux qu'un point stationnaire de  $f$ . C'est en effet souvent le cas, mais ce point faible est rarement rédhibitoire en pratique.

On est parfois tenté d'arrêter l'algorithme si le critère  $f$  ne décroît presque plus. Ceci n'est pas sans risque et il vaut mieux ne pas utiliser un tel test d'arrêt, car une faible variation du critère peut se produire loin d'une solution. En effet, au premier ordre,  $f(x^{(k+1)}) \approx f(x^{(k)}) + \alpha^{(k)} \cdot \nabla f(x^{(k)})^T d^{(k)} \approx f(x^{(k)})$ , ce qui peut arriver si le pas  $\alpha^{(k)}$  est petit (c'est en général très suspect) ou si la direction de descente fait avec l'opposé du gradient un angle proche de 90 degrés, une situation qui se rencontre fréquemment (si l'algorithme est bien conçu, cela traduit un mauvais conditionnement du problème).

Par ailleurs, un algorithme du type descente est dit convergent s’il existe un minimum local  $x^*$  du critère qui lui est passé pour argument pour lequel l’une des deux éventualités suivantes serait réalisée en choisissant :  $x = x^*$  pour test d’arrêt :

- L’algorithme s’arrête après un nombre fini  $k$  d’itérations.
- Il construit théoriquement (en supposant tous les calculs exacts et la capacité de calcul illimitée) une suite infinie  $x^{(1)}, \dots, x^{(k)}, \dots$  de points de  $\mathfrak{R}^n$  convergeant vers  $x^*$ .

En pratique, le test d’arrêt passé pour argument devra être choisi pour garantir que l’algorithme s’arrête toujours après un nombre fini d’itérations et que le dernier point calculé est suffisamment proche de  $x^*$ . Lorsque l’algorithme (suite d’itérations) converge, on dit que sa vitesse de convergence est d’ordre  $p$  s’il existe une constante  $\tau < 1$  telle que :

$$\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^p} \leq \tau \tag{3-22}$$

La constante  $\tau$  est le taux de convergence (ou l’erreur asymptotique) de l’algorithme. En particulier, la convergence est dite : linéaire lorsque  $p = 1$ , superlinéaire lorsque :  $p > 1$  et  $\tau \rightarrow 0$ , et quadratique si  $p = 2$ .

La convergence éventuelle d’un algorithme de descente dépendra toujours des propriétés du critère qui lui sera passé pour argument et, en général, du choix de l’initialisation  $x^{(0)}$ . Il n’existe aucun algorithme universel dont la convergence soit garantie quels que soient le critère ou l’initialisation qui lui seront passés pour argument [Bon97].

Dans la suite, on donne une description sommaire des principaux algorithmes correspondant à ces étapes, émaillée de nombreuses références bibliographiques. On s’efforcera de faire ressortir les avantages et les inconvénients génériques de chaque méthode, et on mentionnera si oui ou non nous l’avons sélectionnée et mise en œuvre pour les problèmes de l’Automatique (faisabilité de cahier des charges, calcul et retouche de correcteurs).

### 3.5.2.1. Calcul de la direction de descente

On peut distinguer deux grandes stratégies de choix de la direction de descente  $d^{(k)}$  au point  $x^{(k)}$  : la stratégie de Cauchy [Cau47] et la stratégie de Newton [Naz94, Ypm95].

#### 3.5.2.1.1. La stratégie de Cauchy

Dans cette stratégie la direction de descente est donnée par  $d^{(k)} = -\nabla f(x^{(k)})$  ce qui conduit aux algorithmes dits de gradient. Ces derniers interviennent lorsque la fonction à optimiser est différentiable (ou qu’on la postule comme telle...). Ils utilisent les informations données par les dérivées partielles de  $f$  pour calculer les itérés du processus, ce qui a pour objectif d’économiser sur le nombre total d’évaluations de la fonction.

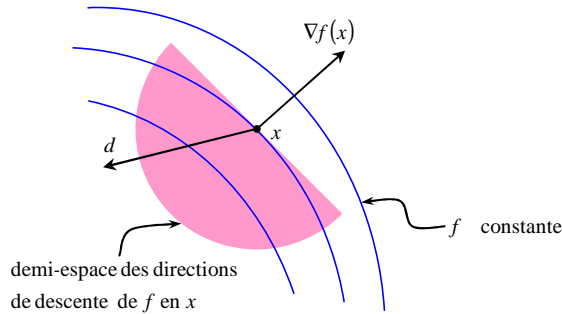
**Théorème 3.11 (Direction de descente à base de gradient)** Supposons  $f$  dérivable au point  $x$ . Si :  $\nabla f(x)^T d < 0$ ,  $d$  est une direction de descente pour  $f$  au point  $x$ .

**Preuve :** si  $\phi(\alpha) = f(x + \alpha d)$ , alors  $\phi'(0) = \nabla f(x)^T d < 0$ .

Par définition du gradient, il revient au même de dire que  $\langle \nabla f(x), d \rangle < 0$  ou encore que  $d$  fait avec l’opposé du gradient  $-\nabla f(x)$  un angle  $\theta$  strictement plus petit que  $90^\circ$ :

$$\theta = \arccos\left(\frac{\langle -\nabla f(x), d \rangle}{\|\nabla f(x)d\|}\right) \in [0, \pi/2[$$

L’ensemble des directions de descente de  $f$  en  $x$ ,  $\{d \in \mathfrak{R}^n \mid \langle \nabla f(x), d \rangle < 0\}$ , forme un demi-espace ouvert de  $\mathfrak{R}^n$  (cf. figure 3.7).



**Fig. 3.7** Demi-espace des directions de descente  $d$  de  $f$  en  $x$

La stratégie du Cauchy calcule la direction qui minimise, à norme constante, la dérivée  $\nabla f(x)^T d$  de  $\varphi(\alpha) = f(x + \alpha d)$ , et retourne la direction  $d = -\nabla f(x)$ , pour laquelle  $\nabla f(x)^T d = -\|\nabla f(x)\|^2$ . Elle définit donc une direction de descente en tout point non critique et correspond à la plus grande diminution de la fonction critère. C’est pour cela qu’elle est souvent appelée méthode du gradient ou méthode de la plus profonde descente.

De ce fait, l’algorithme du gradient semble séduisant; d’autant plus qu’il est *facile à mettre en œuvre* et *robuste* aux éventuelles erreurs de calcul de gradients vu son large domaine de convergence. En effet, tant que l’algorithme n’a pas trouvé un point critique, la valeur du critère décroît strictement à chaque itération.

Cependant, on notera que, pour minimiser une fonction quadratique strictement convexe de deux variables (ce qui correspond à résoudre un système linéaire de deux équations linéaires à deux inconnues), l’algorithme demande, en général, un nombre infini d’itérations, alors que la solution est évidente et aisément calculable analytiquement ou par d’autres algorithmes en un nombre fini d’opérations. En pratique, on observe souvent que  $-\nabla f(x)$  est une bonne direction de descente loin d’une solution mais qu’elle est à éviter dès que l’on entre dans le voisinage d’une solution  $x^*$ , là où les termes du second ordre d’un développement de Taylor de  $f$  autour de  $x^*$  jouent un grand rôle. En fait, le défaut de cet algorithme est d’ignorer la courbure de  $f$ , qui est décrite par son hessien, ce qui limite la vitesse de convergence d’un tel algorithme à être seulement *linéaire*.

Malgré ses piètres performances numériques, cette classe d’algorithmes mérite d’être étudiée. Les techniques utilisées pour l’analyser servent, en effet, souvent de guide dans l’étude d’algorithmes plus complexes (cf. section 4.2).

Une autre méthode de cette catégorie est celle de Hooke et Jeeves. Dans cette méthode, on effectue des minimisations par rapport à une seule variable à la fois. On peut employer pour chacune de ces optimisations un algorithme de plus profonde descente ou un algorithme de Newton (cf. section 3.5.2.1.2). Cette méthode se révèle très mauvaise dans la pratique [Bon97].

**Les méthodes du gradient conjugué**

Les méthodes de gradient ont l’avantage de nécessiter peu d’espace mémoire (pas de stockage de matrices), ce qui les rend incontournables en très grande dimension. C’est pourquoi la méthode du gradient conjugué non linéaire représente une alternative encore très répandue. Il s’agit d’une généralisation de l’algorithme du gradient conjugué linéaire dans laquelle on effectue une recherche linéaire à chaque itération.

$$\begin{cases} d^{(0)} = -\nabla f(x^{(0)}) \\ d^{(k+1)} = -\nabla f(x^{(k)}) + \beta^{(k)} d^{(k)} \\ x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)} \end{cases} \quad (3-23)$$

Dans le cas d’une fonction linéaire, la conjugaison des directions  $d^{(k)}$  et la recherche linéaire exacte induit un choix unique des paramètres  $\beta^{(k)}$  et  $\alpha^{(k)}$ . Dans le cas d’une fonction non linéaire, cela n’est plus possible. On a alors recours à des formules spécifiques pour  $\beta^{(k)}$  et à une recherche linéaire pour le calcul de  $\alpha^{(k)}$ . On dénombre trois choix classiques de  $\beta^{(k)}$ .

Méthode de la plus profonde descente,  $\beta^{(k)} = 0$

Méthode de Fletcher et Reeves, 
$$\beta_{FR}^{(k)} = \frac{\|\nabla f(x^{(k)})\|_2}{\|\nabla f(x^{(k-1)})\|_2}$$

Méthode de Polak et Ribière, 
$$\beta_{PR}^{(k)} = \frac{(\nabla f(x^{(k)}) - \nabla f(x^{(k-1)}))^T \nabla f(x^{(k)})}{\|\nabla f(x^{(k-1)})\|_2}$$

Pour assurer la convergence globale de l’algorithme de Polak et Ribière associé à une recherche linéaire vérifiant les conditions de Wolfe (cf. section 3.5.2.2.3), on a aussi testé  $\beta^{(k)} = \max(\beta_{PR}^{(k)}, 0)$  [Gil92]. On dispose aussi d’algorithmes de recherche linéaire particuliers pour obtenir la convergence globale de l’algorithme de Polak-Ribière [Gri97].

**3.5.2.1.2. La stratégie de Newton**

La seconde stratégie de descente est celle de Newton où la direction de descente est donnée par  $d^{(k)} = -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$  et mène aux algorithmes Newtoniens.

Le principe de ces algorithmes est de calculer, à chaque itération, la direction  $d$  qui minimise l’approximation quadratique :

$$\phi(0) + \alpha \phi'(0) + \frac{\alpha^2}{2} \phi''(0) = f(x) + \alpha \nabla f(x)^T d + \frac{\alpha^2}{2} d^T \nabla^2 f(x) d$$

de  $\phi(x + \alpha d)$  et retourne  $d = -\nabla^2 f(x)^{-1} \nabla f(x)$  pour laquelle  $\phi'(0) = \nabla f(x)^T d = -d^T \nabla^2 f(x) d$ . C’est une direction de descente dès que  $\nabla^2 f(x)$  est définie positive, ce qui sera toujours vérifié si  $x$  est suffisamment proche d’un minimum local non dégénéré de  $f$ .



Lorsqu’ils convergent, les algorithmes Newtoniens ont une vitesse de convergence au pire superlinéaire (si  $\nabla^2 f$  continue) et au mieux *quadratique* (si  $f \in C^3$ ) [Bon97]. Ils sont donc plus rapides que les algorithmes de gradient. Mais ils sont plus coûteux, et surtout moins robustes : loin d’un minimum local, la direction de Newton n’est plus nécessairement une direction de descente.

Les méthodes basiques consistent à choisir une factorisation pour résoudre exactement le système linéaire de Newton. Quand la matrice n’est pas définie positive, on utilise une factorisation de Cholesky modifiée (Gill & Murray [Gil81], Schnabel et Eskow [Sch91]) ou une factorisation de Bunch et Parlett [Bun71]. Dans la pratique, ces méthodes sont très peu utilisées car soit le hessien est trop coûteux, soit la dimension est trop importante pour mettre en œuvre ces factorisations. De plus, le calcul de la direction de Newton de façon précise ralentit considérablement l’algorithme. Comme les problèmes que nous traiterons mettent en jeu peu de paramètres de contrôle, le coût de ces décompositions est très faible et on peut simplement utiliser une décomposition *LU* quand on aura à inverser l’approximation du hessien.

### La méthode de Quasi-Newton

Lorsque le calcul du hessien est très lourd, il est possible de l’approximer à l’aide de matrices dites de quasi-Newton. On se donne une matrice initiale définie positive (généralement égale à l’identité) que l’on met à jour à chaque itération de l’algorithme afin d’approcher de mieux en mieux le hessien  $H = \nabla^2 f$  (ou son inverse  $B = H^{-1}$ ), ce qui est souhaitable vu les propriétés déjà mentionnées de ces deux algorithmes.

La mise à jour de l’approximation du hessien  $H^{(k)}$  (ou de son inverse  $B^{(k)}$ ) est généralement la solution du problème d’optimisation sous contraintes général

$$\begin{aligned} \min \quad & \omega(H^{(k+1)}, H^{(k)}, B^{(k)}, B^{(k+1)}) \\ \left\{ \begin{array}{l} H^{(k+1)} \text{ symétrique} \\ H^{(k+1)}(x^{(k+1)} - x^{(k)}) = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) \end{array} \right. \end{aligned} \quad (3-24)$$

La deuxième contrainte est appelée équation de la sécante vérifiée par la moyenne du hessien sur le morceau de droite compris entre les points  $x^{(k)}$  et  $x^{(k+1)}$ . La fonction  $\omega$  désigne une mesure sur l’espace des matrices. Cette mesure est très souvent liée aux valeurs propres de la matrice  $H^{(k+1)}$ .

Ainsi, on améliore le conditionnement de la matrice  $H^{(k+1)}$  qui doit être inversée pour la résolution du système de Newton. En optimisation sans contraintes, on a plutôt besoin de l’inverse du hessien. On utilise alors dans la majeure partie des cas les formules mettant à jour l’inverse du hessien  $B^{(k)}$ . On s’affranchit alors d’une résolution matricielle à chaque itération.

Par ailleurs, quand on travaille avec le hessien, on utilise le plus souvent des mises à jour qui préservent la définie positivité de la matrice de Quasi-Newton, afin de faciliter la résolution du système matriciel de Newton. C’est le cas de la plupart des mises à jour qui sont citées plus loin.

Plusieurs mesures  $\omega$  ont été étudiées [Wol95]. Ainsi, il a été défini quelques grandes classes de mise à jour dont la plus connue est la classe de Broyden. On dispose des formules générales suivantes pour les matrices  $H^{(k)}$  et  $B^{(k)}$  approximant le hessien et son inverse :

$$H^{(k+1)} = H^{(k)} + \frac{y \cdot y^T}{s^T \cdot y} - \frac{H^{(k)} s s^T H^{(k)}}{s^T H^{(k)} s} + \frac{\beta}{s^T H^{(k)} s} \left( \frac{s^T H^{(k)} s}{s^T \cdot y} y - H^{(k)} s \right) \left( \frac{s^T H^{(k)} s}{s^T \cdot y} y - H^{(k)} s \right)^T \quad (3-25)$$

$$B^{(k+1)} = B^{(k)} + \frac{s \cdot s^T}{s^T \cdot y} - \frac{B^{(k)} y \cdot y^T B^{(k)}}{y^T B^{(k)} y} + \frac{\eta}{y^T B^{(k)} y} \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right) \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right)^T \quad (3-26)$$

où

$$s = x^{(k+1)} - x^{(k)}, \quad y = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) \quad \text{et} \quad \eta \beta \left( \frac{y^T B^{(k)} y}{s^T \cdot y} \cdot \frac{s^T (H^{(k)})^{-1} s}{s^T \cdot y} - 1 \right) + \eta + \beta = 1.$$

La mise à jour de Davidon, Fletcher et Powell ou DFP, qui fut la première proposée, correspond à  $\eta = 0$  (ou  $\beta = 1$ ). Pour  $\eta = 1$  (ou  $\beta = 0$ ), on obtient la très répandue mise à jour de Broyden, Fletcher, Godfarb et Shanno ou BFGS. Cette mise à jour semble en moyenne la plus efficace et moins sensible que DFP aux erreurs commises lors des phases de recherche unidimensionnelle [Wal94].

Ces algorithmes se révèlent assez lents pour des problèmes mal conditionnés car la matrice initiale est très différente du hessien. C'est pourquoi, on peut introduire deux nouvelles formules générales pour les matrices  $H^{(k)}$  et  $B^{(k)}$  qui dépendent de deux paramètres supplémentaires  $\rho$  et  $\gamma$  :

$$H^{(k+1)} = \gamma^{-1} \left( H^{(k)} + \frac{\gamma}{\rho} \cdot \frac{y \cdot y^T}{s^T \cdot y} - \frac{H^{(k)} s s^T H^{(k)}}{s^T H^{(k)} s} + \frac{\beta}{s^T H^{(k)} s} \left( \frac{s^T H^{(k)} s}{s^T \cdot y} y - H^{(k)} s \right) \left( \frac{s^T H^{(k)} s}{s^T \cdot y} y - H^{(k)} s \right)^T \right) \quad (3-27)$$

$$B^{(k+1)} = \gamma \left( B^{(k)} + \frac{\rho}{\gamma} \cdot \frac{s \cdot s^T}{s^T \cdot y} - \frac{B^{(k)} y \cdot y^T B^{(k)}}{y^T B^{(k)} y} + \frac{\eta}{y^T B^{(k)} y} \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right) \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right)^T \right) \quad (3-28)$$

Le paramètre  $\rho$  est un paramètre de stabilisation introduit par Biggs [Big71] et  $\gamma$  est un paramètre d'échelle (scaling) introduit par Oren [Ore74]. Luksan a répertorié les différentes heuristiques pour le choix des paramètres  $\rho$  et  $\gamma$  dans les articles [Luk90, Luk95]. Il a procédé à une comparaison numérique exhaustive des différentes mises à jour [Luk94], de laquelle, il résulte que l'utilisation d'une matrice de Quasi-Newton semble incontournable pour la mise en œuvre d'une méthode Newtonienne.

### **La méthode de Gauss-Newton**

On s'intéresse ici à un problème d'optimisation sans contrainte particulier, celui de la minimisation de la norme  $\ell^2$  d'une fonction  $r: \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ , dont les composantes  $r_i$  sont appelées les résidus :

$$\min_{x \in \mathfrak{R}^n} \left( \frac{1}{2} \|r(x)\|_2^2 \right) \quad (3-29)$$

C'est ce qu'on appelle un problème de moindres carrés non linéaire.

On note  $\mathfrak{J}(x) = r'(x)$  la jacobienne de dimension  $m \times n$  de  $r$  en  $x$ . Alors le gradient et le hessien de  $f$  pour le produit scalaire euclidien s'écrivent :

$$\begin{cases} \nabla f(x) = \mathfrak{S}(x)^T r(x) \\ \nabla^2 f(x) = \mathfrak{S}(x)^T \mathfrak{S}(x) + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x) \end{cases} \quad (3-30)$$

Dans l'algorithme de Gauss-Newton, on détermine  $d^{(k)}$  comme solution particulière (il peut y en avoir plusieurs) du système linéaire

$$\left( \mathfrak{S}(x^{(k)})^T \mathfrak{S}(x^{(k)}) \right) \cdot d^{(k)} = -\mathfrak{S}(x^{(k)})^T r(x^{(k)}) \quad (3-31)$$

Si  $J(x^{(k)})$  est injective, on obtient :

$$d^{(k)} = -\left( \mathfrak{S}(x^{(k)})^T \mathfrak{S}(x^{(k)}) \right)^{-1} \mathfrak{S}(x^{(k)})^T r(x^{(k)}) \quad (3-32)$$

Comparée à la direction de Newton, cette direction n'utilise qu'une partie du hessien de  $f$ , de manière à éviter le calcul des dérivées secondes des résidus, qui sont souvent coûteuses à évaluer.

Lorsque  $x^{(k)}$  n'est pas stationnaire ( $\mathfrak{S}(x^{(k)})^T r(x^{(k)}) \neq 0$ ), la direction de Gauss-Newton  $d^{(k)}$  est de descente puisque

$$\nabla f(x^{(k)})^T d^{(k)} = -r(x^{(k)})^T \mathfrak{S}(x^{(k)}) d^{(k)} = -(d^{(k)})^T \left( \mathfrak{S}(x^{(k)})^T \mathfrak{S}(x^{(k)}) \right) d^{(k)} = -\|d^{(k)} \mathfrak{S}(x^{(k)})\|_2^2 < 0$$

La stricte négativité vient du fait que  $\mathfrak{S}(x^{(k)}) d^{(k)} = 0$  impliquerait par (3-31) que  $\mathfrak{S}(x^{(k)})^T r(x^{(k)}) = 0$ , ce que l'on a supposé ne pas avoir lieu.

Pour les fonctionnelles quadratiques faiblement non linéaires, cette méthode est meilleure que la méthode de Quasi-Newton. En contrepartie, on a besoin de calculer  $\mathfrak{S}(x)$ , ce qui s'avère très coûteux quand  $m$  est grand. A ce propos, il est préférable que  $m$  soit plus grand que le nombre de paramètres afin de travailler avec une matrice de rang maximal.

### 3.5.2.2. Calcul de la longueur de descente (recherche linéaire)

Dans cette section, nous allons décrire les différentes manières de déterminer un pas  $\alpha^{(k)} > 0$  le long d'une direction de descente  $d^{(k)}$ . C'est ce que l'on appelle faire de la recherche linéaire. Il s'agit de réaliser deux objectifs.

Le premier objectif est de faire décroître  $f$  suffisamment. Cela se traduit le plus souvent par la réalisation d'une inégalité de la forme

$$f(x^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)}) + \nu^{(k)} \quad \text{avec } \nu^{(k)} < 0 \quad (3-33)$$

Le terme négatif  $\nu^{(k)}$ , joue un rôle clé dans la convergence de l'algorithme utilisant cette recherche linéaire. L'argument est le suivant. Si  $f(x^{(k)})$  est minorée (il existe une constante  $C$  telle que  $f(x^{(k)}) \geq C$  pour tout  $k$ ), alors ce terme négatif  $\nu^{(k)}$  tend nécessairement vers zéro. C'est souvent à partir de la convergence vers zéro de cette suite que l'on parvient à montrer que le gradient lui-même doit tendre vers zéro. Le terme négatif  $\nu^{(k)}$  devra prendre une forme bien particulière si on veut pouvoir en tirer de l'information. En particulier, il ne suffit pas d'imposer  $f(x^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)})$ .

Le second objectif de la recherche linéaire est d’empêcher le pas  $\alpha^{(k)}$  d’être trop petit, trop proche de zéro. Le premier objectif n’est en effet pas suffisant car l’inégalité (3-33) est en général satisfaite par des pas  $\alpha^{(k)} > 0$  arbitrairement petit. Or ceci peut entraîner une fausse convergence, c’est-à-dire la convergence des itérés vers un point non stationnaire, comme le montre l’observation suivante. Si on prend :

$$0 < \alpha^{(k)} < \frac{\varepsilon}{2^k \|d^{(k)}\|}$$

la suite  $\{x^{(k)}\}_{k \geq 1}$  générée par la récurrence de descente est de Cauchy, puisque pour  $1 \leq l < k$  on a :

$$\|x^{(k)} - x^{(l)}\| = \left\| \sum_{i=l}^{k-1} \alpha^{(i)} d^{(i)} \right\| \leq \sum_{i=l}^{k-1} \frac{\varepsilon}{2^i} \rightarrow 0, \text{ lorsque } l \rightarrow \infty$$

Donc la suite  $\{x^{(k)}\}_{k \geq 1}$  converge, disons vers un point  $\bar{x}$ . En prenant  $l=1$  et  $k = \infty$  dans l’estimation ci-dessus, on voit que  $\bar{x} \in B(x^{(1)}, \varepsilon)$  et donc  $\bar{x}$  ne saurait être solution s’il n’y a pas de solution dans  $B(x^{(1)}, \varepsilon)$ . On a donc arbitrairement forcé la convergence de  $\{x^{(k)}\}_{k \geq 1}$  en prenant des pas très petits.

Pour simplifier les notations, on définit la restriction de  $f$  à la droite  $\{x^{(k)} + \alpha d^{(k)} : \alpha \in \mathfrak{R}\}$  comme la fonction  $h_k : \alpha \rightarrow h_k(\alpha) = f(x^{(k)} + \alpha d^{(k)})$ .

Le choix du pas  $\alpha^{(k)}$  obéit à deux objectifs souvent contradictoires ; trouver le meilleur pas possible en effectuant le moins de calculs possibles. Ces objectifs conduisent séparément à deux stratégies dominantes :

Les algorithmes à pas optimal minimisent, à chaque étape, la fonction  $h_k(\alpha)$  en utilisant une procédure unidirectionnelle pour rechercher le meilleur pas possible.

Les algorithmes à pas fixe au contraire se satisfont d’un pas constant, passé pour paramètre à la procédure. Le choix du pas, effectué une fois pour toutes, dépend alors, en général, d’une analyse de convergence de l’algorithme utilisé et des propriétés du critère à minimiser.

L’expérience montre que ces stratégies radicales sont le plus souvent mauvaises ; il est inefficace d’utiliser un pas constant et il est inutile de calculer à chaque étape le pas optimal. En pratique, on se contentera d’un pas permettant de faire décroître raisonnablement le critère à minimiser.

### 3.5.2.2.1. Recherches linéaires exactes

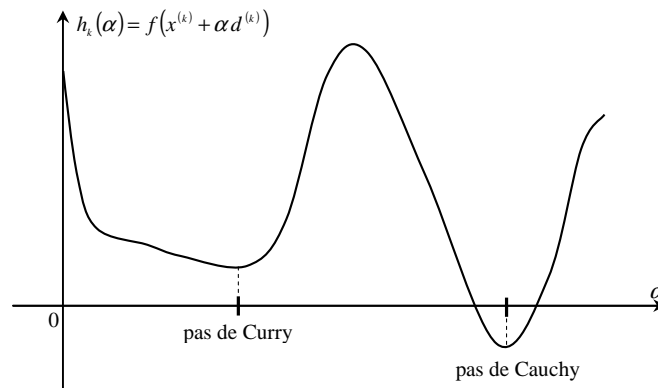
Comme on cherche à minimiser  $f$ , il semble naturel de chercher à minimiser le critère le long de  $d^{(k)}$  et donc de déterminer le pas  $\alpha^{(k)}$  comme solution du problème :

$$\min_{\alpha \geq 0} h_k(\alpha) \tag{3-34}$$

C’est ce que l’on appelle la règle de Cauchy et le pas déterminé par cette règle est appelé *pas de Cauchy* ou pas optimal (cf. figure 3.8). Dans certains cas, on préférera le plus petit point stationnaire de  $h_k(\alpha)$  qui fait décroître cette fonction :

$$\alpha^{(k)} = \inf \{ \alpha \geq 0 : h'_k(\alpha) = 0 \text{ et } h_k(\alpha) < h_k(0) \} \tag{3-35}$$

On parle alors de règle de Curry et le pas déterminé par cette règle est appelé *pas de Curry* (cf. figure 3.8). De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte.



**Fig. 3.8 Règles de Cauchy et Curry**

Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque  $h_k(\alpha)$  est quadratique. En effet, pour une fonction non linéaire arbitraire,

- il peut ne pas exister de pas de Cauchy ou de Curry,
- la détermination de ces pas demande en général beaucoup de temps de calcul et ne peut de toute façon pas être faite avec une précision infinie,
- l’efficacité supplémentaire éventuellement apportée à un algorithme par une recherche linéaire exacte ne permet pas, en général, de compenser le temps perdu à déterminer un tel pas,
- les résultats de convergence autorisent d’autres types de règles, moins gourmandes en temps de calcul.

Au lieu de demander que  $\alpha^{(k)}$  minimise  $h_k(\alpha)$ , on préfère imposer des conditions moins restrictives, plus facilement vérifiées, qui permettent toutefois de contribuer à la convergence des algorithmes. En particulier, il n’y aura plus un unique pas (ou quelques pas) vérifiant ces conditions mais tout un intervalle de pas (ou plusieurs intervalles), ce qui rendra d’ailleurs leur recherche plus aisée. C’est ce que l’on fait avec les règles d’Armijo, de Goldstein et Price et de Wolfe décrites ci-dessous.

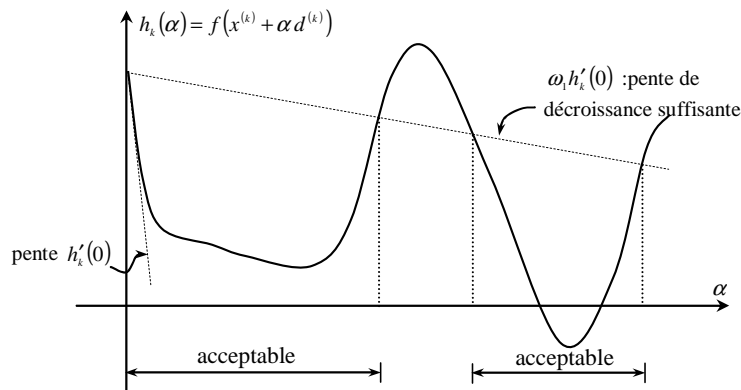
Les méthodes les plus simples consistent à effectuer une recherche itérative en subdivisant l’intervalle initial sur  $\alpha^{(k)}$  par la méthode de la section dorée, la méthode de Fibonacci ou par une recherche dichotomique. Ces méthodes simples sont généralement très gourmandes en temps de calcul et donc peu utilisées quand l’évaluation de la fonction est coûteuse.

#### 3.5.2.2.2. Condition d’Armijo

Une condition naturelle est de demander que  $f$  décroisse autant qu’une portion  $\omega_1$  de ce que ferait le modèle linéaire de  $f$  en  $x^{(k)}$ . Cela conduit à l’inégalité suivante, parfois appelée *condition d’Armijo* ou *condition de décroissance linéaire* :

$$f(x^{(k)} + \alpha^{(k)} d^{(k)}) \leq f(x^{(k)}) + \omega_1 \alpha^{(k)} \nabla f(x^{(k)})^T d^{(k)} \quad (3-36)$$

Elle est de la forme (3-33), car  $\omega_1$  devra être choisi dans  $]0,1[$ . On voit bien à la figure 3.9 ce que signifie cette condition.

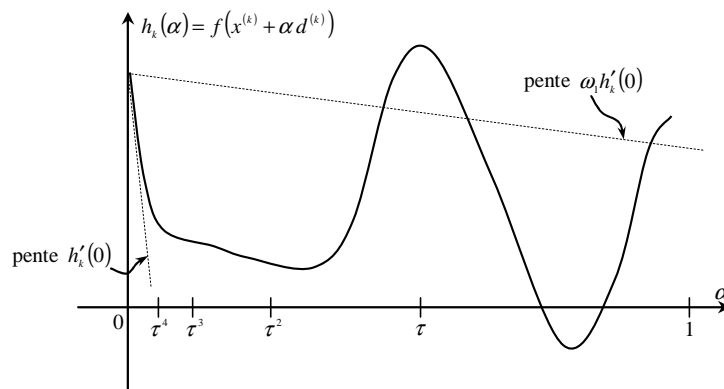


**Fig. 3.9** Les intervalles de pas suggérés par la règle d’Armijo

Il faut qu’en  $\alpha^{(k)}$ , la fonction  $h_k(\alpha)$  prenne une valeur plus petite que celle prise par la fonction affine  $f(x^{(k)}) + \omega_1 \alpha \nabla f(x^{(k)})^T d^{(k)}$ .

En pratique, la constante  $\omega_1$  est prise très petite, de manière à satisfaire (3-36) le plus facilement possible. Typiquement,  $\omega_1 = 10^{-4}$ . Notons que cette constante ne doit pas être adaptée aux données du problème et donc que l’on ne se trouve pas devant un choix de valeur délicat. On montrera toutefois que, dans certains algorithmes, il est important de prendre  $\omega_1 < 1/2$  pour que le pas unité ( $\alpha^{(k)} = 1$ ) soit accepté lorsque  $x^{(k)}$  est proche d’une solution.

Il est clair d’après la figure 3.9 que l’inégalité (3-36) est toujours vérifiée si  $\alpha^{(k)} > 0$  est suffisamment petit. D’autre part, on a vu qu’il était dangereux d’accepter des pas trop petits, cela pouvait conduire à une fausse convergence. Il faut donc un mécanisme supplémentaire qui empêche le pas d’être trop petit. On utilise souvent la technique de rebroussement “backtracking” due à Armijo [Arm66] ou celle de Goldstein [Gol65].



**Fig. 3.10** Règle d’Armijo avec rebroussement

Dans sa version la plus simple, la technique de rebroussement consiste à prendre  $\alpha^{(k)} = \tau^{i_k}$ , où  $\tau \in ]0,1[$  est une constante et  $i_k$  est le plus petit entier naturel ( $i_k \in \mathbb{N}$ ) tel que l’on ait (3-36) (cf. figure 3.10). C’est le fait de prendre pour  $\alpha^{(k)}$  le plus grand réel dans  $\{1, \tau, \tau^2, \dots\}$  permettant de vérifier (3-36) qui

garantit que ce pas ne sera pas trop petit. On voit bien pourquoi cette technique porte le nom de rebroussement : on essaie d’abord  $\alpha^{(k)} = 1$  et si ce pas n’est pas acceptable, on rebrousse chemin en essayant des pas plus petits  $\tau$ ,  $\tau^2$  etc.

La règle d’Armijo avec rebroussement se formule sous le problème d’optimisation suivant :

$$\alpha^{(k)} = \max_{i \in \{0,1,2,\dots\}} (\tau^i) \text{ tel que } f(x^{(k)} + \tau^i d^{(k)}) < f(x^{(k)}) + \omega_1 \tau^i \nabla f(x^{(k)})^T d^{(k)} \quad (3-37)$$

Le code de cette règle sera de la forme :

1. **choisir**  $\tau \in ]0,1[$
2. **poser**  $\alpha^{(k)} = 1$
3. **tant que**  $f(x^{(k)} + \alpha^{(k)} d^{(k)}) \geq f(x^{(k)}) + \omega_1 \alpha^{(k)} \nabla f(x^{(k)})^T d^{(k)}$  **faire** :
4.          $\alpha^{(k)} = \tau \cdot \alpha^{(k)}$
5. **fin tant que**

L’avantage majeur de cette technique de recherche linéaire est de ne pas entraîner de surcoût lié à un calcul de gradient qui nécessite un temps de calcul non négligeable.

### 3.5.2.2.3. Règle de Wolfe

Les recherches linéaires les plus évoluées sont celles qui utilisent le gradient de la fonction [Fle87]. Une des plus puissantes consiste à satisfaire les conditions dites de Wolfe. On exige alors :

$$\begin{cases} f(x^{(k)} + \alpha^{(k)} d^{(k)}) \leq f(x^{(k)}) + \omega_1 \alpha^{(k)} \nabla f(x^{(k)})^T d^{(k)} \\ \nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})^T d^{(k)} \geq \omega_2 \nabla f(x^{(k)})^T d^{(k)} \end{cases} \quad (3-38)$$

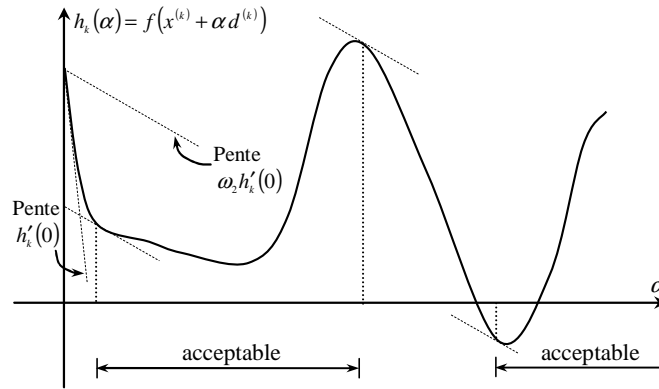
pour un couple  $\omega_1$  et  $\omega_2$  appartenant à  $[0,1]$  fixé à l’avance et vérifiant  $\omega_1 < \omega_2$ .

Il arrive souvent qu’on préfère les conditions de Wolfe dites fortes. La deuxième condition dans (3-38) est alors remplacée par :

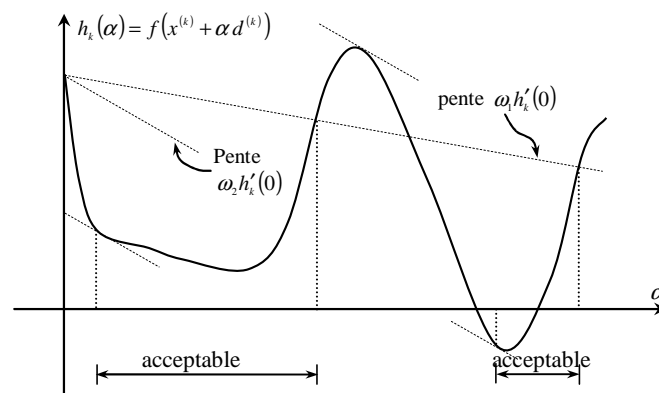
$$\left| \nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})^T d^{(k)} \right| \leq \omega_2 \left| \nabla f(x^{(k)})^T d^{(k)} \right| \quad (3-39)$$

La première condition dans (3-38) n’est autre que la condition de décroissance linéaire d’Armijo. La seconde condition est dite *condition de courbure*. Elle interdit le choix de pas trop petits pouvant entraîner une convergence lente. Le choix  $\omega_1 = 10^{-4}$  et  $\omega_2 = 0,99$  est un choix standard pour ces paramètres [Bon97].

Le pas déterminé par cette règle est appelé pas de Wolfe. Les schémas ci-dessous interprètent géométriquement les deux conditions (3-38).



**Fig. 3.11** La condition de courbure dans la règle de Wolfe



**Fig. 3.12** Règles de Wolfe

Cette règle de recherche linéaire est bien adaptée aux algorithmes de quasi-Newton [Bon97]. Les conditions de la règle de Wolfe ne sont pas constructives. En pratique, on utilise des algorithmes spécifiques pour trouver un pas de Wolfe, on peut citer les algorithmes de Fletcher et Lemaréchal [Fle87] et de Moré et Thuente [Mor94]. Dans notre étude, l’algorithme communément utilisé de Lemaréchal [Lem89] est implémenté et testé (cf. section 4.2.6.2).

**3.5.2.2.4. Recherche de Goldstein et Price**

La recherche de Goldstein et Price représente un compromis entre la recherche d’Armijo et la recherche de Wolfe (forte). Comme le gradient est souvent coûteux et que d’autre part il est important de satisfaire la condition de courbure dans (3-38), au moins en un sens faible, on approxime le gradient intervenant dans la condition de courbure par une approximation aux différences finies selon :

$$\nabla f(x^{(k)} + \alpha^{(k)} d^{(k)}) \approx \frac{f(x^{(k)} + \alpha^{(k)} d^{(k)}) - f(x^{(k)})}{\alpha^{(k)}} \tag{3-40}$$

Il n’est alors pas plus coûteux d’utiliser cette méthode que la recherche d’Armijo. En revanche, on rappelle qu’une recherche de Wolfe coûte plus cher car elle nécessite un calcul de gradient supplémentaire à chaque itération de la recherche linéaire.



Le choix de paramètres  $\omega_1$  et  $\omega_2$  varie d’une référence à une autre. Nocedal [Noc99], suggère de choisir  $\omega_2 = 1 - \omega_1$  avec  $0 < \omega_1 < 1$ . Selon [Bon97], les valeurs typiques de  $\omega_1$  et  $\omega_2$  pour cette règle de recherche sont respectivement 0,1 et 0,7

Les trois recherches linéaires précédentes ont été implémentées et testées sur les algorithmes développés dans la section 4.2.

En ce qui concerne le choix de la longueur de descente initiale, on peut choisir durant les trois premières itérations, par exemple, de l’algorithme d’optimisation de prendre une longueur plus grande que la valeur standard. En effet, on se situe loin de l’optimum, on a donc intérêt à prendre des pas grands pour converger plus vite. Pour les itérations suivantes, on adopte la valeur 1.

#### 3.5.2.2.5. *Interpolation cubique et parabolique*

Ces heuristiques jouent un rôle important dans l’algorithme de recherche linéaire lors du calcul de l’itéré  $\alpha^{(k+1)}$  car elles réduisent le nombre d’itérations du processus d’optimisation (soit le nombre d’évaluations de fonction et de gradient) par rapport à une recherche itérative par pure dichotomie par exemple. Avec les valeurs de la fonction et de son gradient en certains points, on peut calculer les paramètres d’une cubique (ou d’une parabole) qui approxime de façon plus ou moins satisfaisante la fonction. On se sert alors du minimum de la cubique, facile à calculer, pour passer de  $\alpha_i^{(k+1)}$  à  $\alpha_{i+1}^{(k+1)}$ .

Pour nos problèmes, nous n’avons pas sélectionné ce type de recherche. Comme nous sommes limités par le nombre d’itérations, nous pensons en effet ne pas pouvoir réellement profiter de la non monotonie pour converger vers de meilleures valeurs.

### 3.5.3. Les méthodes mixtes

On a vu précédemment que les algorithmes stochastiques peuvent converger vers le minimum global très lentement et que les algorithmes de descente assurent une convergence vers un minimum local rapidement. Dans le cas de critères différentiables fortement non linéaires, des algorithmes hybrides sont développés afin de combiner les avantages des algorithmes précédents.

La combinaison la plus simple consiste à faire une optimisation locale sur la solution (recuit simulé) ou les solutions (algorithmes génétiques) données par l’algorithme stochastique. Cette démarche permet d’augmenter la précision des algorithmes stochastiques avec peu de travail supplémentaire.

La démarche que suivent la plupart de ces algorithmes est celle qui consiste à remplacer dans un algorithme stochastique la valeur de la fonction critère en un point par la valeur de la fonction critère obtenue après optimisation locale à partir de ce point. Ces algorithmes sont appelés plus généralement algorithmes mimétiques [Rad94]. La méthode SALO (Simulated Annealing and Local Optimization) utilise le recuit simulé [Des96], et la méthode GLS (Guided Local Search) utilise la méthode Tabou [Vou95, Tsa97]. Dans le même esprit, on note aussi le développement de deux algorithmes qui améliorent notablement les algorithmes génétiques. L’algorithme génétique parallèle et l’algorithme BGA (Breeder Genetic Algorithm) sont des extensions des algorithmes mimétiques qui exploitent efficacement le parallélisme des machines [Muh91, Muh93]. Ces algorithmes semblent donner de très bons résultats en optimisation combinatoire, l’optimisation locale s’effectuant bien sûr de différentes manières.

Pour tous les algorithmes précédents, l’algorithme principal est issu des méthodes stochastiques. A l’opposé, on peut citer deux autres méthodes qui résultent d’une adaptation des méthodes de gradient. La première est la méthode du gradient stochastique. Dans le calcul de la direction de descente, des bruits sont introduits afin d’éviter les minima locaux. Ceci se traduit par un terme supplémentaire dans le passage  $x^{(k)}$  à  $x^{(k+1)}$ ,

$$x^{(k+1)} = x^{(k)} - \tau_k \nabla f(x^{(k)}) + \sqrt{\frac{\tau_k}{v_k}} \varepsilon_k \quad (3-41)$$

où  $\varepsilon_i$  sont typiquement des variables identiquement distribuées suivant la loi normale  $\mathfrak{N}(0,1)$ , et les suites  $(\tau_k)_{k \in \mathbb{N}}$  et  $(v_k)_{k \in \mathbb{N}}$  jouent le même rôle que la température guidant l’algorithme de recuit simulé. Il faut que  $\tau_k \rightarrow 0$ ,  $v_k \rightarrow +\infty$  et  $\sum_k \tau_k \rightarrow +\infty$ .

L’intérêt du gradient stochastique est qu’on ne peut pas, comme dans le gradient déterministe, converger numériquement vers un point selle. De plus, on est assuré, en théorie, de converger vers le minimum global pour des évolutions des paramètres suffisamment lentes.

Quand le temps de calcul de la fonction critère et du gradient est dérisoire, ces algorithmes améliorent de façon notable les algorithmes stochastiques. Par rapport aux algorithmes de gradient, ils sont beaucoup plus sûrs (plus de chance de convergence vers le minimum global) même si le temps de calcul est encore trop grand. Comme les algorithmes stochastiques, ces algorithmes sont difficilement utilisables lorsque la fonction critère et son gradient sont coûteux, ce qui est le cas pour les problèmes que nous abordons.

### 3.5.4. Les méthodes de recherche directe

Cette branche de l’optimisation déterministe s’intéresse aux problèmes mettant en jeu une fonction critère non différentiable, une fonction critère bruitée, ou une fonction critère présentant un grand nombre de minima locaux.

Les algorithmes d’optimisation directe sont ceux qui ne requièrent que l’évaluation de la fonction critère, mais pas celle des dérivées. Bien que les algorithmes évolutionnaires ou les recherches aléatoires pures, par exemple, soient des algorithmes d’ordre directes, cette section n’abordera que les algorithmes locaux. Les méthodes directes sont importantes car, en pratique, un très grand nombre de fonctions à optimiser ne sont pas dérivables et parfois même pas continues.

Les méthodes locales directes les plus répandues sont les recherches directes (direct search). Pour préciser le terme “recherche directe”, la définition considérée dans ce travail est celle utilisée dans [Wri96] : *un algorithme de recherche directe utilise seulement les valeurs de la fonction et n’approxime pas de gradient*. Le deuxième critère dans cette définition exclue, par exemple, les méthodes qui utilisent les différences finies pour évaluer le gradient. Ainsi, les méthodes de recherche directe peuvent accepter des nouvelles itérations qui conduisent à une simple amélioration de la fonction coût, ce qui contraste avec les conditions d’Armijo, Goldstein et Wolfe pour les méthodes de descente qui requièrent qu’une condition de descente suffisante soit satisfaite [Lew00].

Les méthodes de recherche directe ont été proposées pendant les années 50 et 60. Des exemples des premières méthodes sont les algorithmes de Box (1957), Hooke et Jeeves (Pattern Search Method, 1961), Spendley, Hext et Himsworth (1962), Powell (1964), et l’algorithme de Nelder-Mead (1965)

[Wri96]. Depuis leurs publications, les algorithmes de recherche directe sont utilisés avec succès dans plusieurs domaines, et sont populaires particulièrement en chimie, ingénierie des procédés et médecine. Ils sont réputés pour être des algorithmes simples, robustes, et efficaces pour les problèmes d’optimisation en variables réelles, sans contraintes, où les fonctions sont bruitées [Mck98]. Malgré l’efficacité et la popularité auprès des praticiens, pendant plusieurs années la communauté scientifique d’optimisation ne s’est pas intéressée à ces méthodes car, à part quelques exceptions comme l’algorithme de Powell, des propriétés théoriques de convergence n’ont pas été prouvées. L’intérêt a ressurgi avec les publications [Tor91] et [Den91], qui présentent une méthode appelée recherche multidirectionnelle, adaptée pour être efficace en calcul parallèle et qui possède des propriétés de convergence [Tor91].

Les méthodes de recherche directe peuvent être divisées en trois groupes : les méthodes de recherches par motifs généralisés (Generalized Pattern Search methods, GPS), les méthodes des directions conjuguées (algorithme de Powell et ses variantes), et les méthodes basées sur la figure géométrique d’un simplexe (méthode de Nelder-Mead et ses variantes).

Les méthodes de recherches par motifs généralisés et de Powell sont exposées ci-dessous. La méthode de Nelder-Mead est discutée dans le chapitre suivant.

#### **3.5.4.1. Méthodes de recherches par motifs généralisés**

Les generalized pattern search methods (GPS) [Tor97] sont une généralisation de la méthode de Hooke et Jeeves (1961). La spécificité de ces méthodes est que les directions de recherche ne changent pas avec les itérations. Les GPS sont caractérisées par une série de déplacements exploratoires autour du point courant. Ces déplacements forment des motifs qui présentent une disposition invariable (patterns). A chaque itération la fonction objectif est évaluée sur les points du motif. Si une amélioration est trouvée, le point associé est accepté comme nouveau point courant, et la taille du prochain motif est conservée ou augmentée. Sinon, la taille du nouveau motif, généré autour de l’ancien point courant, est réduite. Les GPS présentent des propriétés de convergence robustes pour des fonctions continues, différentiables et bornées [Tor97, Dol03].

Dans [Lew99, Lew02] les méthodes GPS sont appliquées à des problèmes avec bornes et avec des contraintes non-linéaires, respectivement. La fonction “patternsearch” de Matlab est une variante de ces algorithmes. Par la suite, cette dernière est choisie comme une méthode référence afin qu’elle soit évaluée et comparée à des algorithmes que nous développerons dans le chapitre 4 et ceci pour une série de problèmes d’optimisation analytiques.

#### **3.5.4.2. Directions conjuguées (algorithme de Powell)**

L’algorithme de Powell [Pow64, Pre92] réalise des minimisations unidimensionnelles suivant des directions conjuguées. Deux vecteurs (ou directions)  $s_1, s_2 \in \mathbb{R}^n$  sont conjugués vis à vis d’une matrice définie positive et symétrique  $A$  si  $s_1^T A s_2 = 0$ .

L’algorithme est motivé par la propriété selon laquelle le minimum d’une fonction quadratique  $f$  à  $n$  variables est trouvé en  $n$  minimisations unidirectionnelles successives suivant  $n$  directions conjuguées, et par le théorème suivant (cf. preuve en [Bre73]). De plus, des directions conjuguées peuvent être construites au moyen de ce théorème.

**Théorème 3.12 (Directions conjuguées)** Soit  $f$  une fonction quadratique,  $s_1, s_2, \dots, s_p$ ,  $p$  directions conjuguées,  $x^p$  et  $y^p$  les résultats de  $p$  minimisations unidirectionnelles suivant les directions  $s_i$  en partant de  $x^0$  et  $y^0$ , respectivement. Alors la direction  $s_{p+1} = y^p - x^p$  est conjuguée par rapport aux autres directions  $s_i$  pour  $i = 1, \dots, p$ .

Le détail de la méthode de construction des directions conjuguées n’est pas donné ici. Le lecteur intéressé consultera les références [Pow64], [Min86] ou [Bre73].

Les minimisations unidimensionnelles peuvent être accomplies, sans calcul de dérivées, par dichotomie ou par section dorée [Cul94].

Ces méthodes de recherche directe sont des méthodes locales, ainsi elles peuvent s’arrêter quand un minimum local est trouvé. Afin de les utiliser dans l’optimisation globale de fonctions multimodales, il est nécessaire de les associer à une stratégie d’exploration de l’espace de recherche (cf. section 3.5.3).

### 3.6. Conclusions

Dans ce chapitre, nous avons exposé les principales techniques d’optimisation non linéaire dont les concepts peuvent être utilisés pour créer ou améliorer les méthodes génériques que nous avons mises au point.

Initialement, le concept de pénalisation est introduit. Il permet de simplifier un problème d’optimisation avec contraintes (comme celui d’un cahier des charges multi-objectif) en le transformant en un problème ou une série de problèmes d’optimisation sans contraintes. Néanmoins, l’utilisation de cet artifice mathématique n’est pas gratuite, elle occasionne l’apparition de difficultés telles que les nouveaux extremums et les zones non différentiables. Malgré tout, l’utilisation de la pénalisation dans les problèmes de faisabilité des cahiers des charges ne peut qu’être avantageuse car elle permet de reformuler un problème d’optimisation multi-objectif difficile en un problème mono-objectif de complexité similaire.

Dans un deuxième temps, un état de l’art des techniques d’optimisation non linéaire capables de résoudre le nouveau problème d’optimisation sans contraintes est dressé. Une classification de ces techniques d’optimisation est proposée. Elle montre quels sont les principes de base et les propriétés des principaux algorithmes de chaque classe.

Les méthodes déterministes qui ont été avancées présentent des propriétés de robustesse vis-à-vis des erreurs de calcul du critère et de la direction de descente. Dans cette catégorie, les méthodes de descente à base de gradient et les méthodes de recherche directes à base de simplexe sont très performantes. C’est pourquoi elles sont sélectionnées pour faire l’objet d’une étude plus profonde et de généralisation à des problèmes d’optimisation différentiables presque partout. Ces études seront présentées au chapitre 4.



# Chapitre 4

## Algorithmes d'optimisation développés

Dans un premier temps, ce chapitre est dédié à l'exposition des algorithmes développés au cours de notre travail pour la résolution des problèmes d'optimisation non linéaires formulés dans le deuxième chapitre.

Le premier est un algorithme de recherche directe (ne nécessitant pas un calcul de variations) basé sur l'algorithme du simplexe (algorithme de Nelder-Mead). Nous commençons par une analyse détaillée de cet algorithme qui est une composante à part entière de l'algorithme proposé. Les limites de l'algorithme du simplexe sont mis en avant afin de l'améliorer et de l'adapter aux problèmes d'optimisation liés au cahier des charges. Enfin, deux variantes de l'algorithme, désignées ASM (algorithme du simplexe modifié), sont exposées. Leurs résultats numériques pour des problèmes d'optimisation analytiques typiques sont donnés dans la section 4.3.

Une deuxième classe d'algorithmes d'optimisation est introduite dans la section 4.2. Ses méthodes sont basées sur la technique de plus profonde descente et sur une généralisation de la notion du gradient. Le but est, principalement, de développer une méthode présentant à la fois une qualité de robustesse d'un algorithme du type gradient et une extension aux problèmes d'optimisation non différentiables. Pour ce faire, nous nous intéressons à la notion du sous-différentiel et ses propriétés. Particulièrement, le problème de calcul des sous-différentiels est posé et une approximation tirée de la notion du  $\varepsilon$ -sous-différentiel de Clarke est présentée. En effet, cette notion permet d'approcher assez efficacement un sous-différentiel par un ensemble convexe construit à base d'un nombre fini de gradients autour d'un point non différentiable donné. Un premier algorithme dit AESD (algorithme du  $\varepsilon$ -sous-différentiel) est développé et soumis à la même série de tests. Ses résultats seront ensuite discutés et comparés avec ceux de l'algorithme ASM.

Dans un deuxième temps, nous proposons de nouvelles versions de cet algorithme où les principales modifications visent à améliorer les performances en termes de nombre d'itérations et de taux de convergence. Une première amélioration est réalisée en réduisant le nombre de gradients nécessaire pour définir l'ensemble convexe approximant le  $\varepsilon$ -sous-différentiel. L'algorithme résultant est nommé AESDM (algorithme du  $\varepsilon$ -sous-différentiel modifié). La deuxième modification suggérée consiste à coupler l'algorithme AESD avec un algorithme de type quasi-Newton afin de réduire le nombre d'itérations et d'accélérer la convergence lors des phases différentiables du critère. L'algorithme final développé est baptisé AGU (algorithme de gradient universel). Les performances de chaque algorithme sont évaluées et comparées en utilisant les mêmes problèmes tests. L'apport des améliorations proposées est illustré par une étude comparative globale avec les différents algorithmes.

## 4.1. Algorithme du simplexe modifiée

L’algorithme que nous proposons est basé sur l’algorithme classique de Nelder-Mead (cf. section 4.1.1). Cet algorithme, qui est parfois appelé méthode du simplexe, ne doit pas être confondu avec la méthode du simplexe mise au point par Dantzig et utilisée en programmation linéaire.

Le principe de cet algorithme est le suivant : on part d’un simplexe dont les sommets sont définis par  $N+1$  points (pour un espace de paramètres de dimension  $N$ ). Suivant les valeurs des fonctions au sommet, on effectue des transformations sur ce simplexe (réflexion, expansion et contraction) jusqu’au moment où le simplexe est quasiment réduit à un point.

Un attribut de l’algorithme de Nelder-Mead est ne pas requérir ni la différentiabilité du critère ni le calcul de gradient. Or le calcul de dérivées ou différentiels est souvent une étape laborieuse et délicate dans les modèles de systèmes physiques (cf. chapitre 5). Le second avantage de Nelder-Mead est d’être une méthode rapide et précise par rapport aux recherches stochastiques. Par exemple, les algorithmes génétiques sont numériquement coûteux quand on a besoin de résultats précis [Ren94]. Ces avantages de Nelder-Mead sont ceux, plus généralement, des algorithmes de recherche directe [Lew00, Tor97].

Cependant, la méthode classique de Nelder-Mead présente des inconvénients : elle s’applique à des variables sans bornes et à problèmes sans contraintes et la recherche peut échouer par stagnation sur un point non-stationnaire (cf. section 4.1.2). C’est pourquoi des améliorations de la méthode de Nelder-Mead sont proposées. L’algorithme ASM développé est donc pragmatique. Il résout des problèmes contraints en variables réelles et bornées et il ne nécessite pas de calcul de variations car il s’appuie sur la méthode de Nelder-Mead. En outre, il peut devenir globale via une série de réinitialisations si le coût des analyses le permet [Lue04].

### 4.1.1. Algorithme de Nelder-Mead (Simplexe)

La méthode de Nelder-Mead [Nel65] ou méthode du “polytope mouvant”, est fondée sur l’algorithme de Spendley, Hext et Himsforth [Spe62], et utilise un arrangement de  $(n+1)$  points  $x_i$  où la fonction coût est évaluée,  $n$  étant la dimension du domaine de recherche. Cet arrangement peut être vu comme les sommets d’un simplexe. Un simplexe régulier de taille initiale  $a$  est initialisé en  $x_0$ , par exemple au moyen de la règle suivante [Haf93],

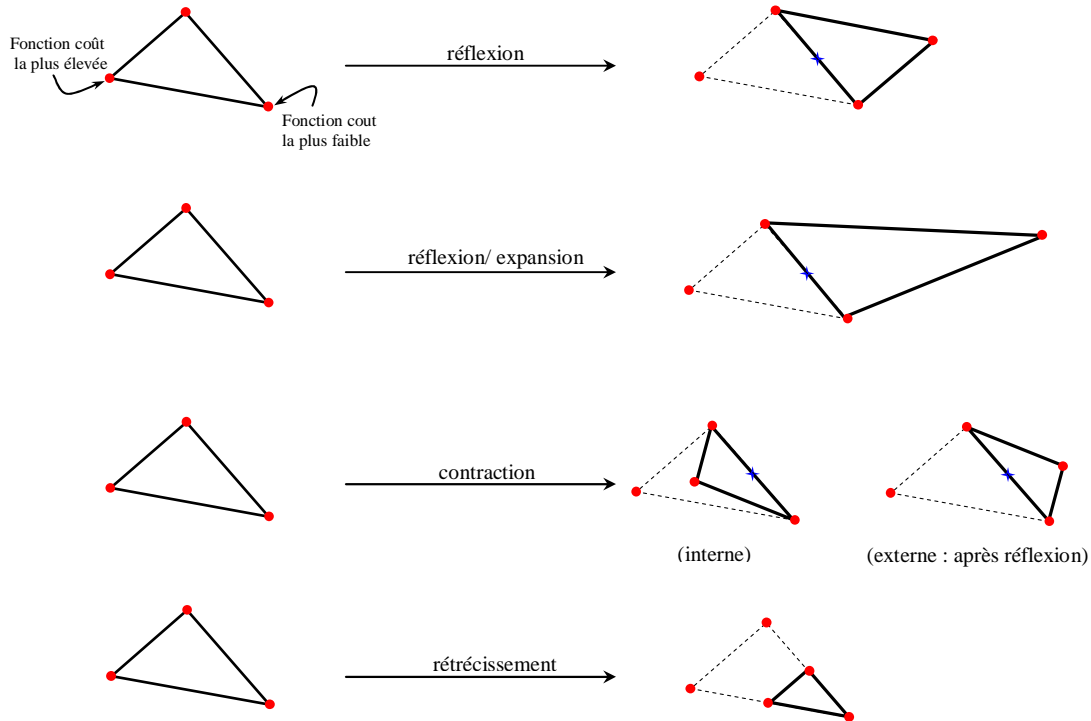
$$x_i = x_0 + pu_i + \sum_{\substack{k=1 \\ k \neq i}}^n qu_k, \quad i = 1, \dots, n \quad (4-1)$$

où  $u_i$  sont les vecteurs unitaires de la base et

$$\begin{cases} p = \frac{a}{n\sqrt{2}}(\sqrt{n+1} + n - 1) \\ q = \frac{a}{n\sqrt{2}}(\sqrt{n+1} - 1) \end{cases} \quad (4-2)$$

Chaque itération de la méthode commence avec les sommets d’un simplexe et les valeurs correspondantes de la fonction coût. Le simplexe est modifié à travers les opérations de réflexion,

d’expansion, de contraction, ou par un rétrécissement, et un point est accepté ou rejeté en fonction de sa valeur de fonction coût. La figure 4.1 Montre les effets de ces opérations pour un simplexe dans un domaine bidimensionnel (un triangle).



**Fig. 4.1** Les opérations dans la méthode de Nelder-Mead. Le simplexe original est représenté par des traits discontinus et + représente le barycentre des sommets hormis le plus mauvais

Une itération générique présente deux possibilités :

- Un nouveau sommet au moins meilleur que le plus mauvais sommet lui est substitué,
- Un rétrécissement est effectué où un ensemble de  $n$  nouveaux points plus le meilleur des anciens points constituent le simplexe de la prochaine itération.

L’organigramme de la méthode de Nelder-Mead est présenté dans la figure 4.2. Les valeurs recommandées dans [Nel65] pour les coefficients de réflexion  $r$ , contraction  $\beta$  et expansion  $\gamma$  sont 1, 1/2 et 2, respectivement. Ces valeurs seront utilisées dans ce mémoire.

Une interprétation intuitive de cet algorithme est qu’une direction de recherche est définie par le plus mauvais point (celui dont la fonction coût est la plus élevée) et le barycentre des sommets hormis le plus mauvais. Le simplexe peut accélérer (expansion) ou décélérer (contraction) dans cette direction pour localiser une région optimale et zoomer (rétrécissement) vers l’optimum. La recherche s’achève quand les valeurs des fonctions aux sommets sont proches :

$$\sqrt{\sum_{i=1}^{n+1} (f_i - \bar{f})^2} / n < \varepsilon_f \quad \text{avec} \quad \bar{f} = \frac{1}{n+1} \sum_{i=1}^{n+1} f_i \quad (4-3)$$



où  $f_i$  est la valeur de la fonction coût en  $x_i$ , et  $\varepsilon_f$  est un petit scalaire positif représentant la tolérance sur le critère  $f$ .

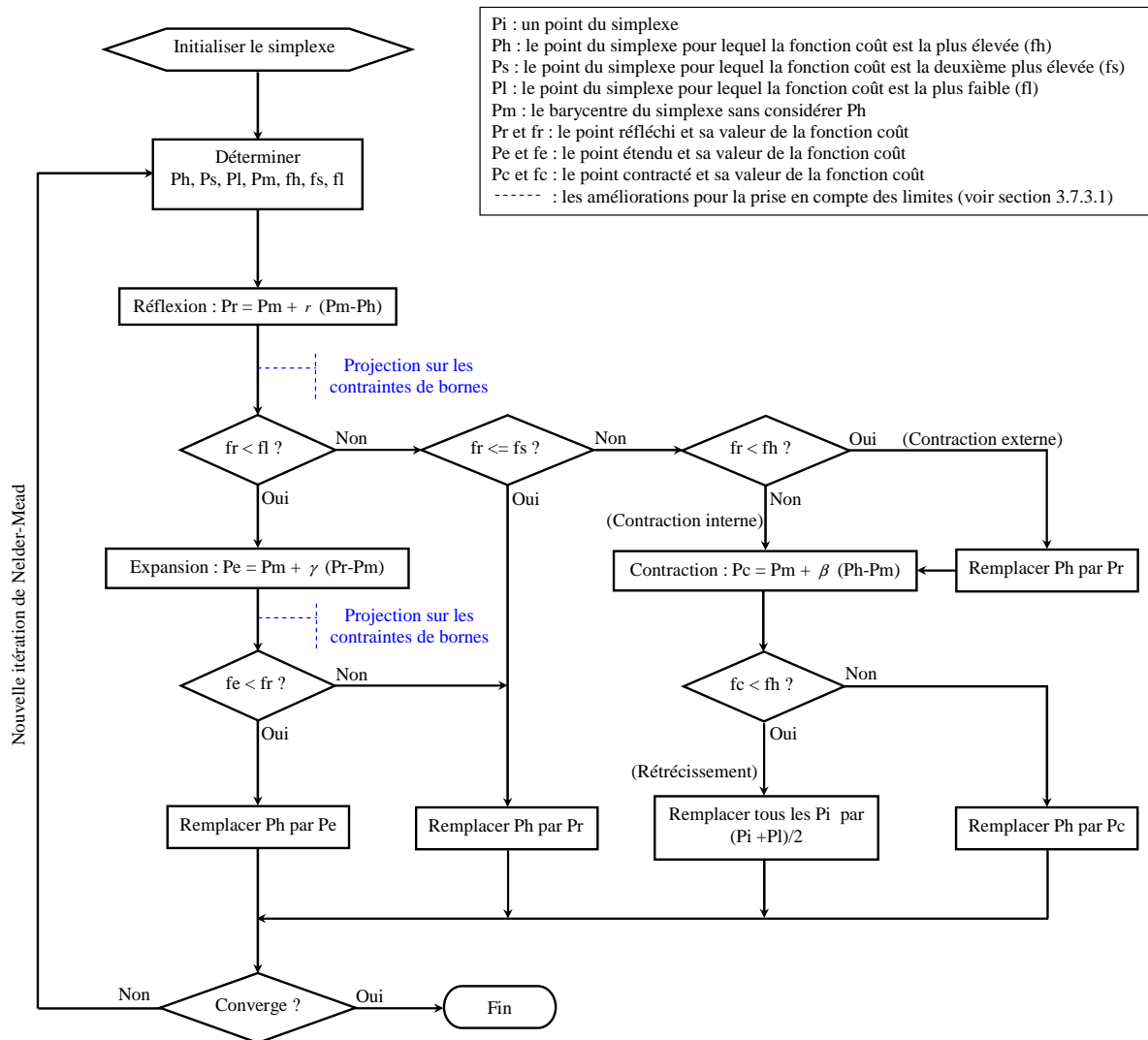


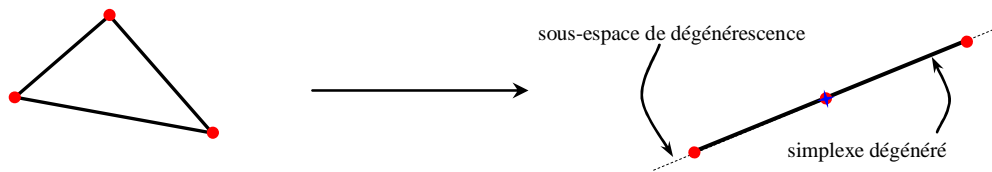
Fig. 4.2 Organigramme de l’algorithme de Nelder-Mead (Simplexe)

La convergence vers un point minimum n’est pas garantie pour la méthode de Nelder-Mead [Wri96, Kel99, Kol03]. Des preuves de convergence pour des fonctions strictement convexes unidimensionnelles et des résultats restreints ont été obtenus pour un ensemble de fonctions bidimensionnelles. Ils sont donnés dans [Lag98, Mck98]. Dans ces références, le comportement de la méthode appliquée à une famille de fonctions bidimensionnelles strictement convexes et continues jusqu’à la troisième dérivée, où la convergence (ou stagnation) se produit sur un point non-stationnaire est analysé. L’explication de l’échec est la capacité du simplexe à se déformer pendant la recherche, où des déformations répétées peuvent amener à une dégénérescence du simplexe sur un sous-espace du domaine de dimension inférieure. Nonobstant ces inconvénients, l’algorithme de Nelder-Mead est probablement la méthode de recherche directe la plus utilisée et référencée [Lag98] et est, en général, très efficace et rapide par rapport aux autres méthodes de recherche directe. Elle peut requérir, par exemple, moins d’évaluations de la fonction coût que la recherche multidirectionnelle [Wri96]. Par rapport aux méthodes de motifs (GPS), l’algorithme de Nelder-Mead est capable de distordre le

simplexe pour mieux s’adapter à la topologie de la fonction. Ceci lui confère à la fois une force et une faiblesse : l’algorithme gagne en vitesse de convergence, mais perd en robustesse (et en preuve formelle de convergence).

Dans [Nel65] les méthodes de Nelder-Mead et de Powell sont comparées pour la minimisation d’une fonction à deux variables (fonction test de Rosenbrock [Ros60]), d’une fonction à trois variables (fonction test de Fletcher et Powell), et d’une fonction à quatre variables (fonction test quadratique de Powell). Dans ces exemples, pour atteindre le minimum avec la même précision finale, la méthode de Nelder-Mead nécessite moins d’évaluations que la méthode de Powell. Comme les autres méthodes de recherche directe, elle est robuste pour les problèmes discontinus ou bruités [Mck98]. L’algorithme de Nelder-Mead est aussi la méthode de recherche directe la plus utilisée dans les codes de calcul numérique. Par exemple, elle fait partie du Matlab Optimization Toolbox. C’est la fonction “fminsearch” pour l’optimisation sans contraintes et sans limites sur les variables.

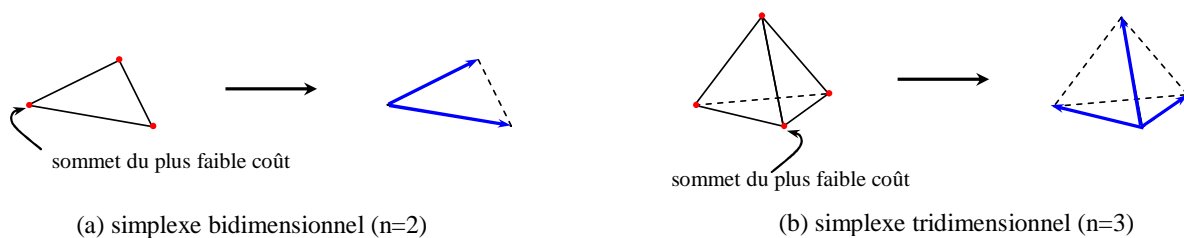
Une propriété importante de l’algorithme doit être mentionnée. Comme évoquée auparavant, la méthode peut ne pas converger vers un optimum si le simplexe dégénère dans un sous-espace du domaine, c’est-à-dire si toutes les arêtes émanant d’un même sommet deviennent linéairement dépendantes. Le simplexe ne peut plus alors sortir du sous-espace couvert par ses arêtes (cf. section 4.1.2 et figure 4.3).



*Fig. 4.3 Dégénérescence d’un simplexe dans un espace bidimensionnel*

### 4.1.2. Détection et traitement des dégénérescences

La dégénérescence du simplexe est le symptôme le plus courant d’un échec de la recherche avec l’algorithme de Nelder-Mead [Wri96]. Il est possible que lors des opérations sur le simplexe certaines arêtes attachées à un même sommet deviennent linéairement dépendantes. Dans ce cas, la méthode de Nelder-Mead n’est capable de rechercher un minimum que dans le sous-espace décrit par les arêtes. Afin d’éviter ce problème, une vérification de la dégénérescence a été mise en œuvre à chaque itération. Pour cette vérification, deux tests sont faits sur les arêtes du simplexe qui partent du sommet de plus faible fonction coût.



*Fig. 4.4 Les arêtes du simplexe utilisées dans les tests de dégénérescence pour  $n=2$  et  $n=3$*

La figure 4.4 montre comment sont définies les arêtes dans les espaces bi et tridimensionnel. Le simplexe est considéré dégénéré s’il n’est pas petit (la définition de simplexe petit, utilisée comme critère de convergence, est donnée en section 4.1.3.1 par l’équation (4-6)), s’il ne touche pas les bornes (cf. section 4.1.3), et si l’une des conditions suivantes est satisfaite :

$$\frac{\min_{k=1,\dots,n} \|e_k\|}{\max_{k=1,\dots,n} \|e_k\|} < \varepsilon_{\text{deg } 1} \quad \text{ou} \quad \frac{\det(e)}{\prod_{k=1,\dots,n} \|e_k\|} < \varepsilon_{\text{deg } 2} \quad (4-4)$$

où  $e_k$  est la  $k^{\text{ième}}$  arête,  $e$  est la matrice formée par les composantes des arêtes, et  $\varepsilon_{\text{deg } 1}$  et  $\varepsilon_{\text{deg } 2}$  sont des petites constantes positives. Le premier test vérifie s’il y a une arête très petite par rapport aux autres, et le deuxième test analyse la dépendance linéaire des arêtes.

Si la dégénérescence est détectée pour un simplexe intérieur au domaine de définition des variables, il est réinitialisé en utilisant comme point initial son meilleur sommet (celui qui possède la fonction coût la plus basse). Les autres sommets du simplexe sont définis suivant les équations (4-1) et (4-2). Le test de dégénérescence n’est pas exécuté lorsque des sommets touchent les bornes des variables car la dégénérescence peut être légitime, c’est-à-dire, le simplexe dégénéré peut continuer à être un simplexe non-dégénéré mais dans l’hyperplan des bornes actives (un “sous-simplexe” dans un “sous-espace”) et la recherche peut continuer sur ces bornes. Pour un simplexe complètement sur les bornes, si le nombre d’arêtes linéairement indépendantes du simplexe  $\text{rang}(e)$  est égal à la dimension du sous-espace ( $n$  - nombre de bornes actives), cette dégénérescence est légitime.

Notons pourtant que si le simplexe perd plus de dimensions qu’il y a de bornes actives, il n’a plus la capacité d’explorer tout cet hyperplan : le “sous-simplexe” est dégénéré dans le “sous-espace”. D’autres configurations de dégénérescence peuvent encore être envisagées quand une partie du simplexe est sur les bornes et l’autre est dans le domaine. Pour garder une mise en œuvre suffisamment simple, les tests de dégénérescence de sous-simplexes dans les sous-espaces ne sont pas réalisés. On se restreint au test présenté en équation (4-2) (avec les conditions attenantes), pour que les vérifications de dégénérescence soient simples et que le simplexe ait la possibilité d’explorer les bornes. Enfin, si une convergence se produit sur les bornes, le test d’optimalité présenté en section 4.1.1 est réalisé.

### 4.1.3. Prise en compte des bornes

L’algorithme original de Nelder-Mead a été proposé pour des problèmes sans contraintes de bornes. Or une grande partie des problèmes d’optimisation, et en particulier la plupart des problèmes de commande, possède ce type de contraintes.

Afin d’adapter la méthode de Nelder-Mead à la résolution de cette catégorie de problèmes d’optimisation, deux stratégies de prise en compte des bornes sont mises en œuvre :

#### 4.1.3.1. Prise en compte des bornes par projection

Dans un premier temps, on propose de vérifier les contraintes de bornes, a posteriori, par projection :

$$x^j = \begin{cases} x^{j,\min} & \text{si } x^j < x^{j,\min} \\ x^{j,\max} & \text{si } x^{j,\max} < x^j \end{cases} \quad (4-5)$$

où  $x^j$  est la  $j^{\text{ième}}$  coordonnée du sommet  $x$  à analyser, et  $x^{j,\min}$  et  $x^{j,\max}$  sont, respectivement, les bornes inférieure et supérieure dans la direction  $j$ .

Dans le cas d’un algorithme de simplexe, la projection peut intervenir après les étapes de réflexion ou d’expansion (cf. figure 4.2). Une conséquence de la projection sur les bornes est que le simplexe peut dégénérer dans l’hyperplan des bornes actives (cf. section précédente). Si le simplexe a convergé avec des bornes actives, il peut soit avoir convergé vers un minimum local, soit avoir convergé vers un minimum dégénéré.

Dans le cas où les fonctions et contraintes sont considérées comme différentiables, l’optimalité locale est vérifiée sous la forme des conditions de Karush, Kuhn et Tucker. Ici, cependant, des fonctions pas nécessairement différentiables sont considérées. En guise de test d’optimalité le long des bornes, un redémarrage est réalisé à partir d’un “petit” simplexe au point de convergence. Si la recherche retourne au même point, il s’agit d’un minimum local sur les bornes. Remarquons que la taille du petit simplexe de redémarrage doit être supérieure à celle du critère de convergence (dans ce cas, le critère de convergence utilisé par ASM est présenté dans la suite de cette section), et suffisamment petite pour rester dans le même bassin d’attraction. Si le point de convergence avec bornes actives est un minimum dégénéré, la recherche continue vers un autre minimum. Dans ce cas, on remarque que cette recherche peut être coûteuse, puisque la taille initiale du simplexe est très petite. Un exemple de dégénérescence sur les bornes est présenté en section 4.3.

En conséquence, la mise en œuvre des améliorations de l’algorithme de Nelder-Mead, qui composent l’ASM, est basée sur un jeu d’options de réinitialisations dont l’organigramme est représenté par la figure 4.5. Les réinitialisations ont pour objectif l’amélioration et la validation des convergences locales de l’algorithme. Les deux schémas associés à la convergence initialisent un nouveau simplexe en utilisant comme point initial le meilleur point du simplexe courant. Les réinitialisations nommées locale et large utilisent un simplexe “petit” et “grand”, de tailles  $a_s$  et  $a_l$ , respectivement. Ces réinitialisations sont associées au test d’optimalité sur les bornes (cf. section 4.1.3) et à la dégénérescence dans le domaine (cf. section 4.1.2), respectivement.

La convergence des recherches locales de l’algorithme de Nelder-Mead avec variables bornées est estimée à travers trois critères : simplexe petit, plat, ou dégénéré.

Le simplexe est considéré petit par rapport aux bornes si

$$\max_{k=1,\dots,C_s+1} \left( \sum_{j=1}^n \left| \frac{e_k^j}{x^{j,\max} - x^{j,\min}} \right| \right) < \varepsilon_x \quad (4-6)$$

où  $e_k^j$  est la  $j^{\text{ième}}$  coordonnée de la  $k^{\text{ième}}$  arête,  $x^{j,\min}$  et  $x^{j,\max}$  sont les bornes inférieure et supérieure dans la  $j^{\text{ième}}$  direction, et  $\varepsilon_x$  est une valeur de tolérance. Ici, contrairement au calcul de la dégénérescence, toutes les arêtes du simplexe sont prises en compte. On considère ainsi qu’un minimum local est trouvé si le simplexe courant est petit (et pas sur les bornes).

Le simplexe est plat si :

$$|f_H - f_L| < \varepsilon_{\text{plat}} \quad (4-7)$$

où  $f_H$  et  $f_L$  sont, respectivement, la valeur la plus haute et la plus faible de la fonction coût parmi les sommets du simplexe courant et  $\varepsilon_{plat}$  est une valeur de tolérance. Ce test permet d’échapper, par un redémarrage aléatoire local (T<sub>3</sub>), aux éventuelles régions plates de la fonction coût.

Le critère pour déterminer si un simplexe est dégénéré dans le domaine a été présenté dans la section 4.1.2 (cf. équation (4-4)).

L’enchaînement des trois réinitialisations (aléatoire, locale et large) et des trois critères de convergence (simplexe petit, plat et dégénéré) est montré en figure 4.5 Une mémoire des points de convergence des recherches passées est conservée, ainsi une série d’analyses permet de décider de la nature du résultat obtenu. Un simplexe qui est dégénéré entraîne une réinitialisation large (T<sub>7</sub>).

Quand l’optimalité des points de convergence est incertaine, comme dans le cas d’une convergence sur les bornes lorsque le simplexe est dégénéré (T<sub>5</sub>), une réinitialisation locale est exécutée en guise d’un test d’optimalité. Si le petit simplexe retourne au même point de convergence, ceci est considéré comme un minimum local.

Les tolérances pour détecter si un simplexe est petit ou dégénéré,  $\varepsilon_x$  et  $\{\varepsilon_{deg1}, \varepsilon_{deg2}\}$ , respectivement, peuvent être difficiles à régler. Par exemple, un simplexe qui devient petit pourrait avant être jugé dégénéré. Ainsi, si une dégénérescence est détectée consécutivement deux fois au même point, ce point est gardé, puisqu’il est un possible optimum. Similairement, si une dégénérescence est détectée après le test d’optimalité (réinitialisation locale), ce point est aussi gardé comme un possible optimum, et une nouvelle recherche avec redémarrage au meilleur point du simplexe courant (réinitialisation large) est lancée. Une fois que le nombre d’évaluations atteint  $N_{max}$  ou le nombre d’itération atteint  $C_{max}$ , l’exécution se termine, et le résultat de la recherche est un optimum local (éventuellement global).

Toutefois, nous notons bien que cette technique de prise en compte des contraintes de bornes est universelle et peut être transposée à un n’importe quelle méthode d’optimisation itérative. Elle sera d’ailleurs utilisée dans un algorithme de descente à base de gradient à la fin de ce chapitre.

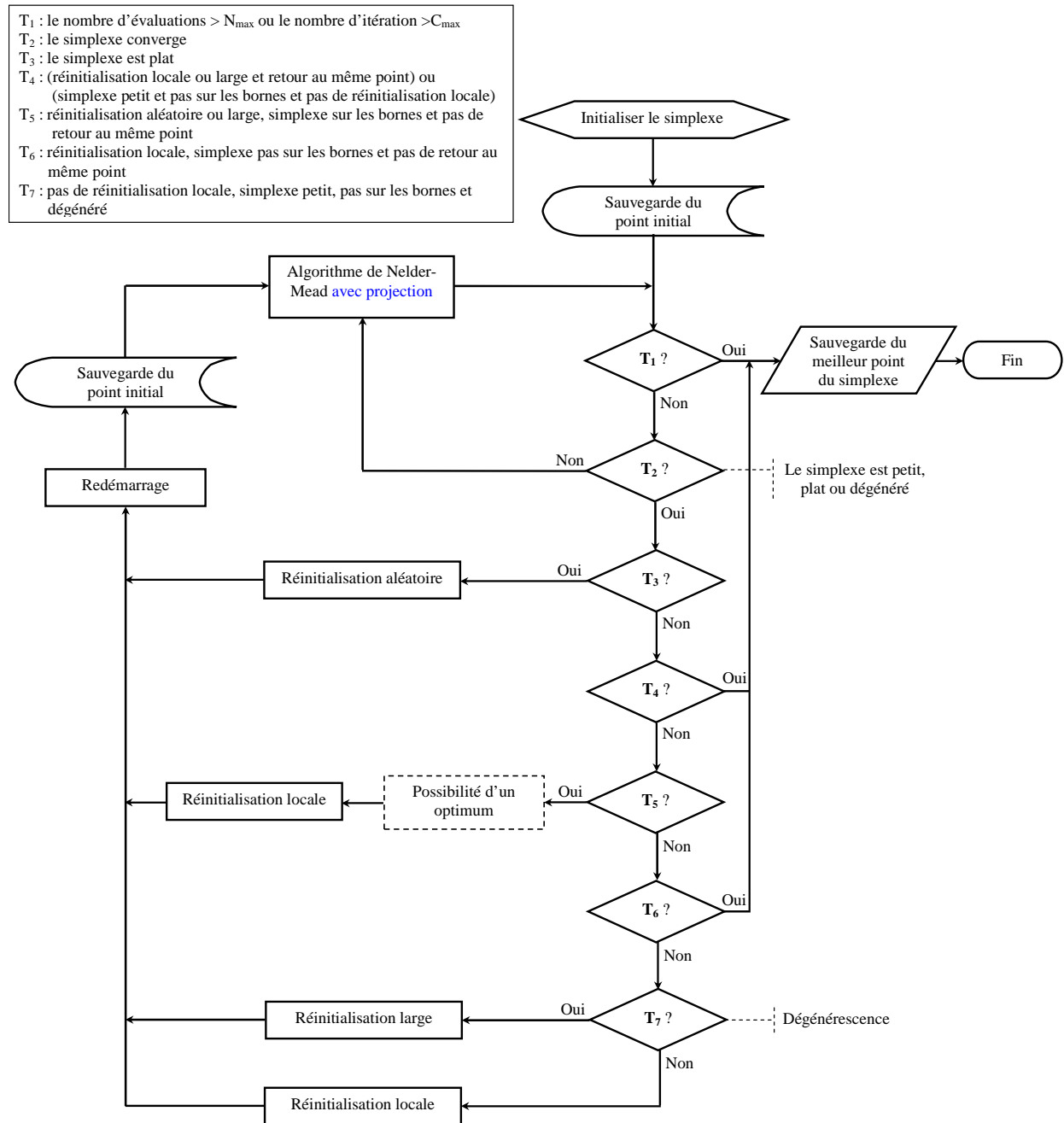
#### 4.1.3.2. *Prise en compte des bornes par reparamétrisation*

Il existe une deuxième approche qui permet de considérer, a priori cette fois, des contraintes de bornes dans un problème d’optimisation. Cette technique dite de changement de variables permet, par une modification du critère, de se ramener à un problème sans contraintes. Elle possède l’avantage d’être compatible avec les techniques d’optimisations sans contraintes qui forment l’essentiel de ce chapitre.

On trouve dans la littérature plusieurs types de changement de variables. Pour imposer, par exemple, la contrainte  $x^j \geq 0$ , on peut remplacer  $x^j$  par  $\exp(y^j)$  avec  $y^j \in \mathfrak{R}$ , ce qui garantit l’inviolabilité de la contrainte. De même, remplacer  $x^j$  par  $\tanh(y^j)(x^{j,max} - x^{j,min})/2 + (x^{j,max} + x^{j,min})/2$ , avec  $y^j \in \mathfrak{R}$  garantit  $x^{j,min} \leq x^j \leq x^{j,max}$ .

Ces reparamétrisations non linéaires vérifient tous la condition de monotonie croissante et ceci pour éviter l’apparition d’extrémums supplémentaires non désirées.

Dans ce qui suit, on propose d’utiliser d’autres types de changements de variables potentiels. Les variables bornées sont transformées tel que l’algorithme de Nelder-Mead voit un problème entièrement sans contrainte.



**Fig. 4.5 Organigramme de l’ASM avec prise en compte des contraintes de bornes par projection**

Par exemple, dans le cas d’une variable  $x^j$  limitée inférieurement par  $x^{j,\min}$ , on propose d’employer la transformation :

$$x^j = x^{j,\min} + (y^j)^2 \quad \text{avec} \quad y^j \in \mathfrak{R} \tag{4-8}$$

La nouvelle variable  $y^j$  est entièrement sans contrainte, et comme  $(y^j)^2$  est toujours non négative (pour  $y^j$  réel), alors  $x^j$  doit nécessairement être toujours supérieur ou égal à  $x^{j,\min}$ . De même, une contrainte de limite supérieure ( $x^j \leq x^{j,\max}$ ) est implémentée comme :

$$x^j = x^{j,\max} - (y^j)^2 \quad \text{avec} \quad y^j \in \mathfrak{R} \quad (4-9)$$

Dans ce cas-ci, il est clair que  $x^j$  ne peut jamais dépasser  $x^{j,\max}$ . Et finalement, la variable dualement bornée est manipulée par la transformation trigonométrique suivante :

$$x^j = x^{j,\min} + (x^{j,\max} - x^{j,\min})(\sin(y^j) + 1)/2 \quad \text{avec} \quad y^j \in \mathfrak{R} \quad (4-10)$$

Dans ce dernier cas, on vérifie la contrainte d’encadrement  $x^{j,\min} \leq x^j \leq x^{j,\max}$ . Numériquement, cette contrainte doit être imposée car les caprices de l’arithmétique à virgule flottante pourraient parfois causer le dépassement subtil de ses limites.

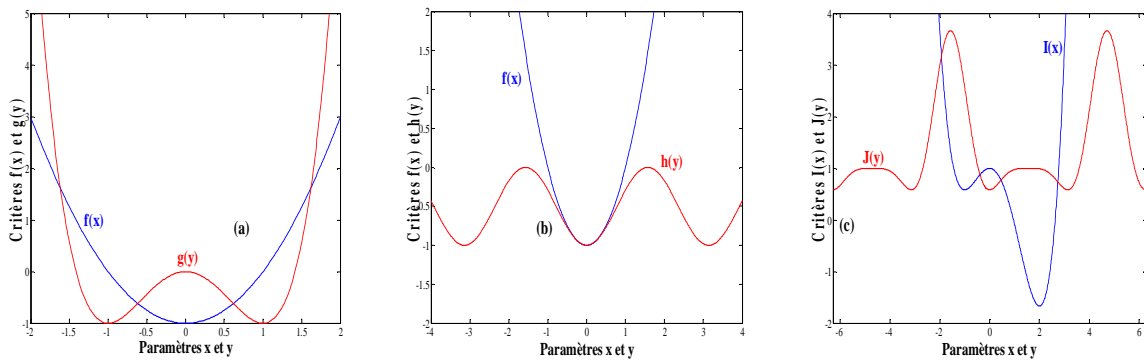
Un artefact des transformations non monotones utilisées est la création de solutions multiples à un problème qui, initialement, ne possède qu’une solution unique. Alors que la présence d’optima locaux multiples est souvent vue comme une grande complexité pour un optimiseur, chacun de ces optima procréés est entièrement équivalent (effet de la puissance carré dans les expressions (4-8) et (4-9) et de la périodicité de la fonction “sin” dans (4-10)). Ainsi, ces transformations n’importent que ce qu’est déjà dans la fonction critère. Cette propriété est illustrée par les exemples suivants :

**Exemple 4.1** Considérons les problèmes suivants :

**P1** : On désire minimiser la fonction coût convexe  $f(x) = x^2 - 1$  sous la contrainte  $x \geq a$ . On utilise le changement de variables (4-8) afin de traduire ce problème en un problème sans contraintes en la variable  $y$ . Le problème résultant est donné par :

$$\min_{y \in \mathfrak{R}} (g(y)) = \min_{y \in \mathfrak{R}} ((a + y^2)^2 - 1) \quad (4-11)$$

La figure 4.6(a) représente les critères du problème avec et sans contrainte pour  $a = -1$ . On remarque que contrairement au critère initial  $f(x)$  qui a un seul minimum global en zéro, le nouveau critère  $g(y)$  possède deux minima et un maximum. Dans les deux cas, la valeur des minima est identique et coïncide avec le minimum global défini par  $x_{opt} = 0$  (ou  $y_{opt} = \pm 1$ ) et  $f_{opt} = g_{opt} = -1$ .



**Fig. 4.6** Effet de la reparamétrisation en un problème sans contraintes

**P2** : On désire de nouveau minimiser la fonction critère  $f(x) = x^2 - 1$  sous la contrainte duale  $a \leq x \leq b$ . En utilisant la paramétrisation (4-10), on transforme le problème sous contraintes **P2** en un problème sans contraintes défini par :

$$\min_{y \in \mathfrak{R}} (h(y)) = \min_{y \in \mathfrak{R}} ((a + (b - a)(\sin(y) + 1)/2)^2 - 1) \quad (4-12)$$

La figure 4.6(b) représente le critère du problème avant et après la reparamétrisation pour les bornes  $a = -b = -1$ . Le nouveau critère  $h(y)$  est périodique et de période  $\pi$  et présente ainsi une infinité de minima équivalents définis par  $y_{opt} = k\pi$  avec  $k \in \mathbb{Z}$  et  $h_{opt} = f_{opt} = -1$ .

**P3**: Le troisième exemple concerne la minimisation du critère polynomial non convexe  $I(x) = x^4/4 - x^3/3 - x^2 + 1$  à l’intérieur de l’intervalle  $[-2,0]$ .

Il est clair que l’optimum global de ce problème est atteint pour  $x_{opt} = -1$  et  $I_{opt} = 7/12$ . L’utilisation de la reparamétrisation de type (4-10) permet de reformuler le problème en un problème sans contraintes dont le critère  $J(y)$  est tracé en rouge dans la figure 4.6(c). Le minimum de ce nouveau critère est atteint pour une infinité de points  $y_{opt}$  qui ont tous pour images, par le changement de variable inverse associé à (4-10), l’optimum global  $x_{opt}$ .

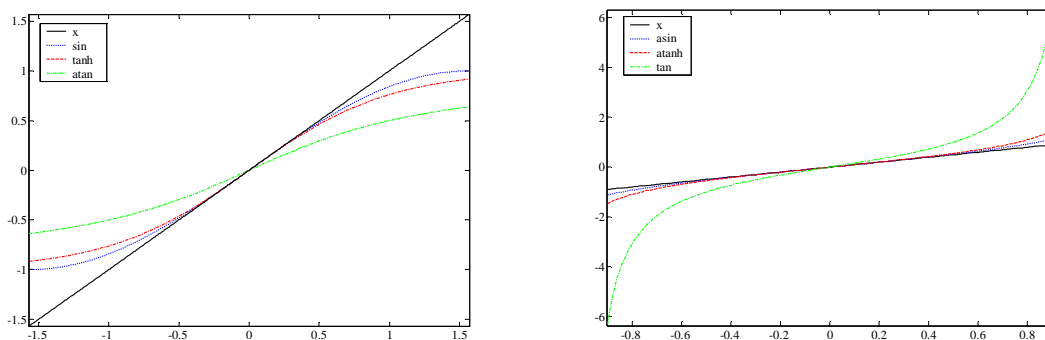
Sous l’hypothèse que l’optimiseur converge toujours vers un minimum, l’initialisation, dans les trois problèmes d’optimisation sans contraintes formulés, n’as aucun effet sur le résultat final.

Les tests analytiques montrent qu’une deuxième transformation trigonométrique pourrait être employée pour compenser l’effet des bornes. Cette dernière est donnée par :

$$x^j = x^{j,\min} + (x^{j,\max} - x^{j,\min}) \sin(y^j)^2 \quad \text{avec} \quad y^j \in \mathfrak{R} \quad (4-13)$$

Néanmoins, les mêmes tests exhibent que cette reformulation est légèrement plus non-linéaire, posant subtilement plus de problèmes en terme d’arithmétique à virgule flottante.

De même, d’autres types de transformations pour des bornes simples et duales ont été étudiés et comparés avec la reformulation proposée. La figure 4.7 montre le tracé des courbes associées aux changements de variables, à base des fonctions “sin”, “tanh” et “arctan”, et leurs inverses. La comparaison de ces courbes révèle une propriété forte de la reparamétrisation à base de la fonction sinus qui possède la dynamique la plus proche de celle de la première bissectrice et qui ne pose pas de problèmes de “limite à l’infini” que l’on peut trouver avec d’autres transformations.



**Fig. 4.7** Comparaison des changements de variables

Cependant, les transformations à base des fonctions “arctan” et “tanh” présentent un grand intérêt. Même si les essais prouvent qu’elles sont souvent plus lentes à converger près des bornes, elles demeurent un bon choix pour la mise en œuvre de limites exclusives (contraintes strictes). En effet, la reformulation des contraintes de bornes proposée concerne seulement les inégalités larges (contraintes inclusives), où les valeurs limites sont permises et prennent, par les changements de variables (4-8, 4-9



et 4-10), une valeur finie atteignable si ces contraintes sont actives. Cette vision reste surtout théorique et pose de nombreux problèmes de mise en œuvre. Si la fonction objective compte, par exemple, une évaluation de  $1/x$  ou  $\log(x)$ , où  $x$  est contraint d’être supérieur ou égal à zéro, alors le programme risquera de produire une singularité. Une inégalité stricte à zéro aurait empêché une telle erreur.

Dans cette deuxième version de l’algorithme ASM, l’utilisation des reformulations à base de fonctions “arctan” et “tanh” est écartée pour des raisons de commodité résumées, principalement, en deux causes : la convergence très lente près des bornes et la complexité algorithmique dans le cas où on indique le type de frontières pour chaque contrainte. Toutefois, une solution plus pratique peut être adoptée et s’avère très efficace. Elle consiste à remplacer les éventuelles inégalités strictes par des inégalités larges légèrement durcies. On applique alors la condition suffisante suivante :

$$(x^{j,\min} + \varepsilon \leq x^j \leq x^{j,\max} - \varepsilon \text{ avec } \varepsilon \in \mathfrak{R}_+^* \text{ et } \varepsilon \text{ petit}) \Rightarrow (x^{j,\min} < x^j < x^{j,\max}) \quad (4-14)$$

Les transformations choisies (4-8, 4-9 et 4-10) sont inversibles. Ceci permet d’assurer le passage entre l’espace contraint et non contraint pour le calcul des conditions initiales au début de l’algorithme et des optima obtenus à sa fin. Dans le cas de conditions initiales infaisables, ces dernières sont tout simplement projetées sur les bornes dès le début.

Pour l’algorithme du simplexe modifié, un critère de convergence plus adapté à l’approche par reparamétrisation serait :

$$|f_H - f_L| < \varepsilon_f \text{ et } \max_{k=1,\dots,C_2^{+1}} (\|e_k\|_\infty) < \varepsilon_x \quad (4-15)$$

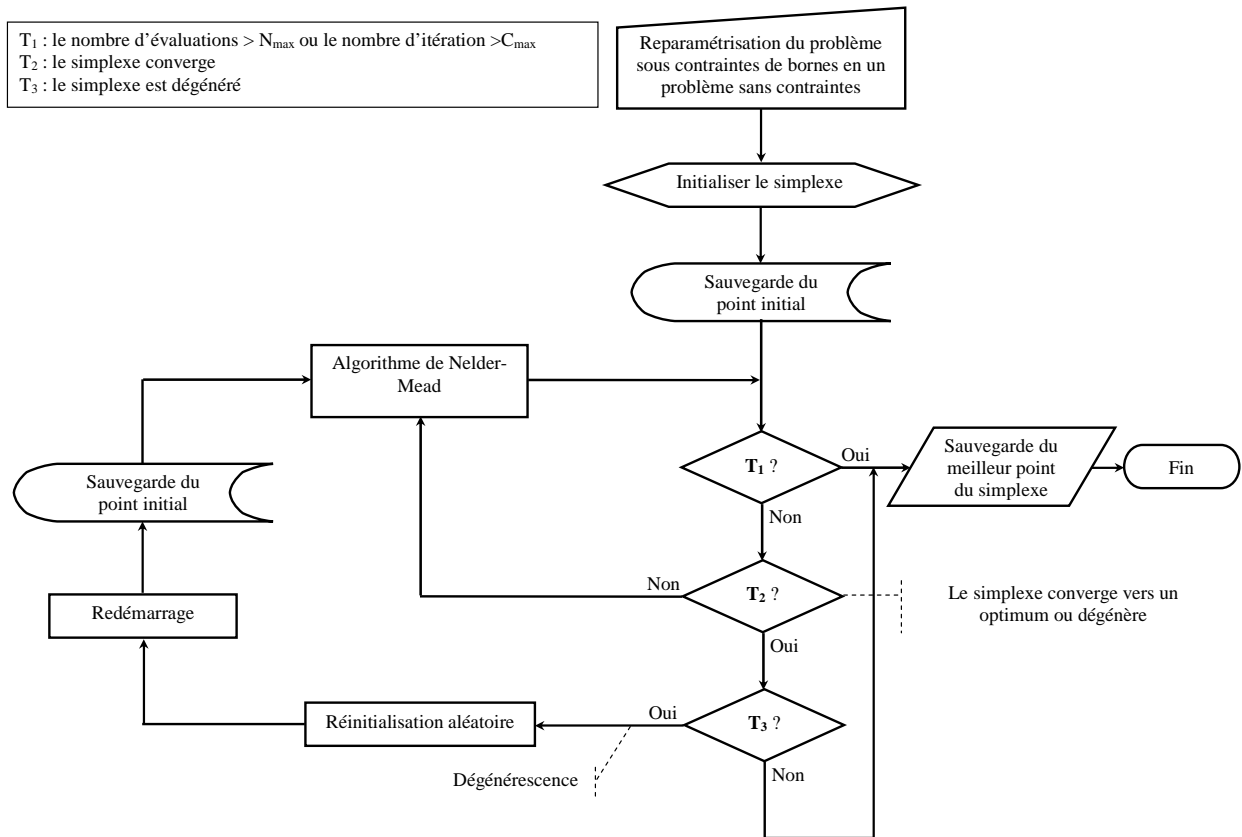
où  $f_H$  et  $f_L$  sont, respectivement, la valeur la plus haute et la plus faible de la fonction coût parmi les sommets du simplexe,  $e_k$  est la  $k^{\text{ième}}$  arrête et  $\varepsilon_f$  et  $\varepsilon_x$  sont les tolérances sur la fonction critère  $f$  et sur les paramètres  $x$  respectivement.

Un autre critère basé, cette fois, sur la taille relative du simplexe est souvent utilisé [Tor91] :

$$\frac{\max_{i=1,\dots,n} (\|x_i - x_0\|_2)}{\max(1, \|x_0\|_2)} < \varepsilon_x \quad (4-16)$$

où  $x_0$  est le meilleur sommet du simplexe  $S = \{x_0, x_1, \dots, x_n\}$  et  $\varepsilon_x$  la tolérance sur les paramètres.

Ainsi l’algorithme du simplexe classique n’est modifié que pour éviter les dégénérescences loin des bornes. Quant à la dégénérescence sur les bornes, elle disparaît car les contraintes sont prises en compte a priori lors de la reformulation du problème. Le diagramme de l’algorithme est présenté par la figure suivante :



**Fig. 4.8 Organigramme de l’algorithme ASM avec prise en compte des bornes par reparamétrisation**

Comme pour la technique de projection sur les bornes, l’approche par reparamétrisation est valable pour tout algorithme itératif. Toutefois, elle présente un inconvénient relatif à la définition de la tolérance sur les paramètres  $\epsilon_x$ . En effet, l’algorithme du simplexe classique (cf. section 4.1.1) qui est le noyau du nouvel algorithme proposé, ne manipule que des paramètres transformés dans un nouvel espace, et en conséquence, il n’y a aucune manière simple d’offrir le contrôle explicite de cette tolérance sur les paramètres sans la réécriture complète de l’algorithme, ce qui n’est pas l’objectif ici.

Les deux algorithmes ASM développés ont été implémentés et testés sous Matlab. La section 4.3 présente quelques résultats de ces algorithmes pour des problèmes de test numériques.

## 4.2. Méthodes du sous-différentiel

L’objectif de cette section est de développer une méthode d’optimisation à directions de descente qui exploite au mieux l’information locale fournie par la fonction objectif. Dans le contexte de notre travail, cette dernière étant non partout différentiable et sujette à des erreurs de calcul numériques, nous proposons de développer une méthode locale qui étend la méthode de plus profonde descente aux cas non différentiables tout en conservant sa robustesse due à la largeur de son domaine de convergence (cf. section 3.5.2.1.1). Pour ce faire, nous utiliserons des extensions de la notion de gradient au cas de critères non différentiables.

Dans ce qui suit, nous faisons un bref rappel des notions de base du calcul sous-différentiel. Pour les preuves et détails, nous renvoyons souvent le lecteur à [Hir93], notamment le chapitre VI. Les livres de R.T.Rockafellar, [Roc70], et de F.H.Clarke, [Cla83], sont des références classiques dans le domaine pour le cas convexe et non convexe respectivement.

### 4.2.1. Généralisation du gradient

Considérons une fonction  $f$  convexe à valeurs finies. Alors,  $f$  est continue et localement Lipschitzienne et la dérivée directionnelle  $f'(x, d) = \lim_{t \rightarrow 0} (f(x + td) - f(x))/t$  existe pour chaque  $x$  et  $d$  fixés dans  $\mathfrak{R}^n$ . Ceci entraîne la différentiabilité de  $f$  en presque tout  $x$  ; nous appellerons *coins* (kink en anglais) tout point  $x$  pour lequel le gradient  $\nabla f(x)$  n’existe pas. Bien que les coins forment un ensemble de mesure nulle, souvent dans la pratique, le résultat d’une optimisation en est un, d’où l’intérêt de l’étude détaillée de ces zones non différentiables.

#### 4.2.1.1. Définitions

Pour décrire le comportement d’une fonction autour d’un coin (points non différentiable), il est nécessaire de généraliser la notion de gradient. Ainsi, on associera à  $x$  tout un ensemble  $\partial f(x)$  appelé le *sous-différentiel* de  $f$  en  $x$  :

$$\partial f(x) = \{g \in \mathfrak{R}^n : f(y) \geq f(x) + g^T(y - x), \quad \forall y \in \mathfrak{R}^n\} \quad (4-17)$$

D’autres définitions équivalentes au sous-différentiel sont :

$$\partial f(x) = \{g \in \mathfrak{R}^n : f'(x, d) \geq g^T d, \quad \forall d \in \mathfrak{R}^n\} \quad (4-18)$$

ou encore :

$$\partial f(x) = \text{conv}\{ \lim_{x_i \rightarrow x} \nabla f(x_i); \text{ pour toute } \{x_i\} : \nabla f(x_i) \text{ et la limite existent} \} \quad (4-19)$$

où  $\text{conv}(S)$  est l’enveloppe convexe de l’ensemble  $S$  définie par :

$$\text{conv}(S) = \left\{ \sum_i \alpha_i s_i : s_i \in S, \sum_i \alpha_i = 1, \alpha_i \in [0, 1] \right\} \quad (4-20)$$

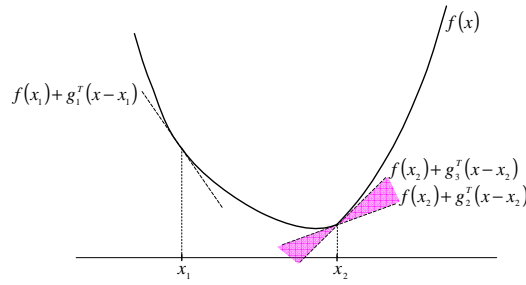
C’est cette dernière définition qui permet une généralisation de la notion du sous-différentiel à des fonctions non convexes. Elle est connue sous le nom de *sous-différentiel de Clarke* [cla83].

Chaque élément  $g$  de l’ensemble  $\partial f(x)$  est appelé *sous-gradient* de  $f$  à  $x$ . Cette notion reste valable pour le cas de fonction  $f$  non convexe.

La définition (4-17) est plus géométrique :  $\partial f(x)$  est formé par les pentes des hyperplans d’appui de l’épigraphe de  $f$  en  $(x, f(x)) \in \mathfrak{R}^n \times \mathfrak{R}$ . Cette caractérisation s’appelle *inégalité du sous-gradient* :

$$g \in \partial f(x) \text{ si et seulement si } f(y) \geq f(x) + g^T(y - x) \text{ pour tout } y \in \mathfrak{R}^n$$

La figure 4.9 illustre cette définition pour une fonction à une dimension. Le point  $x_1$  est différentiable et présente un seul sous-gradient égal à la dérivée de la fonction  $f$  en ce point  $g_1 = \partial f(x_1) = \nabla f(x_1)$ . Le sous-différentiel en ce point se réduit donc au singleton  $\partial f(x_1) = \{\nabla f(x_1)\}$ .



**Fig. 4.9** Sous-gradients

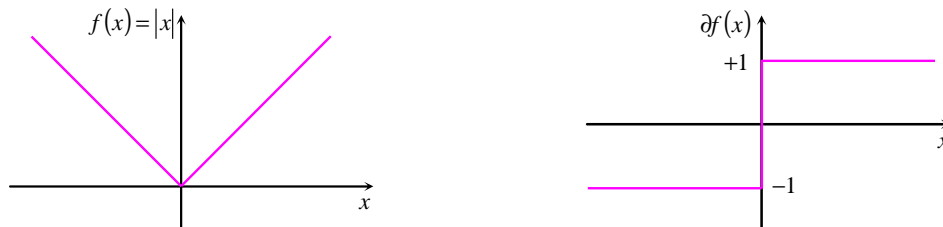
A l’opposé, le point  $x_2$  est un coin. Il possède deux droites tangentes à l’épigraphe de  $f$  de pentes  $g_2$  et  $g_3$  respectivement. Ces pentes ne sont que les dérivées à droite et à gauche du point  $x_2$  et représentent les sous-gradients de la fonction  $f$  en  $x_2$  qui délimitent le sous-différentiel  $\partial f(x_2)$  donné par l’ensemble :

$$\partial f(x_2) = [g_2, g_3] \tag{4-21}$$

**Exemple 4.2** Soit la fonction  $f(x) = |x|$ . Cette fonction est différentiable partout sauf en 0. Son sous-différentiel est défini comme suit :

$$\partial f(x) = \begin{cases} -1 & \text{si } x < 0 \\ [-1, +1] & \text{si } x = 0 \\ +1 & \text{si } x > 0 \end{cases}$$

La figure ci-dessous représente la fonction  $f$  ainsi que son sous-différentiel  $\partial f(x)$ .



**Fig. 4.10** Le sous-différentiel d’une valeur absolue

#### 4.2.1.2. Propriétés

Même en ayant élargi la notion du gradient à tout un ensemble, une grande partie du calcul différentiel reste valable pour des fonctions non différentiable.

Nous rappelons, dans ce paragraphe, les principales propriétés qui découlent des définitions du sous-gradient et du sous-différentiel d’une fonction (le lecteur est renvoyé à [Roc70, Boy04] pour plus de détails) :

$g$  est sous-gradient de  $f$  en  $x$  si et seulement si  $(g, -1)$  est le vecteur d’appui de l’épigraphe de  $f$  à  $(x, f(x))$ .

$g$  est sous-gradient de  $f$  en  $x$  si et seulement si  $f(x) + g^T(y - x)$  est la meilleure approximation affine de  $f$  en  $x$ .

Si  $f(y) \leq f(x) + g^T(y - x)$  pour tout  $y \in \mathfrak{R}^n$  alors  $g$  est dit un *super-gradient*.

Le sous-différentiel  $\partial f(x)$  est un ensemble convexe fermé qui peut être vide. Par exemple, la fonction  $f : [0,1] \rightarrow [0,1]$  définie par  $f(x) = x^{1/2}$  est concave et son sous-différentiel  $\partial f(0) = \emptyset$ .

$\partial f(x) = \{\nabla f(x)\}$  si  $f$  est différentiable en  $x$ .

Dans le cas d’une fonction  $f$  convexe,

- $\partial f(x)$  est non vide pour tout  $x \in \text{dom}(f)$ .
- $\partial f(x) = \{\nabla f(x)\}$ , si  $f$  est différentiable en  $x$ .
- Si  $\partial f(x) = \{g\}$ , alors  $f$  est différentiable en  $x$  et  $g = \nabla f(x)$ .

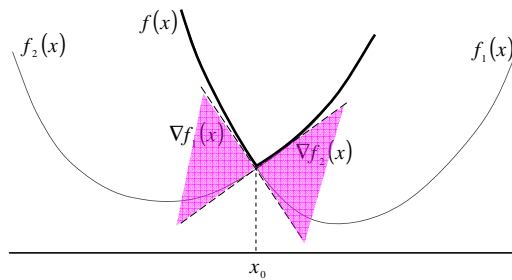
$\partial f(x)$  est homogène : si  $\alpha > 0$  alors  $\partial f(\alpha \cdot x) = \alpha \cdot \partial f(x)$ .

$\partial f(x)$  est additif :  $\partial(f_1 + f_2)(x) = \partial f_1(x) + \partial f_2(x)$  (le second membre est une addition d’ensembles).

Composition avec une fonction affine : si  $g(x) = f(Ax + b)$  alors  $\partial g(x) = A^T \partial f(Ax + b)$ .

Si  $f = \max_{i=1..m}(f_i)$  alors  $\partial f(x) = \text{Conv}\{\partial f_i(x) \mid f_i(x) = f(x)\}$  ; c’est-à-dire, le sous-différentiel  $\partial f(x)$  est l’ensemble convexe des sous-différentiels de fonctions  $f_i$  actives en  $x$ .

**Exemple 4.3** Soit la fonction  $f$  définie par  $f(x) = \max(f_1(x), f_2(x))$ , avec  $f_1$  et  $f_2$  convexes et différentiables (cf. figure 4.11)



**Fig. 4.11** Le sous-différentiel d’une fonction max

Au point coin  $x_0$  les deux fonctions  $f_1$  et  $f_2$  sont actives, le sous-différentiel en ce point est défini par :  $\partial f(x_0) = \text{Conv}\{\nabla f_1(x_0), \nabla f_2(x_0)\} = [\nabla f_1(x_0), \nabla f_2(x_0)]$ . Ainsi, le sous-différentiel de la fonction  $f$  est donné par :

$$\partial f(x) = \begin{cases} \nabla f_1(x) & \text{si } f_1(x) > f_2(x) \\ [\nabla f_1(x), \nabla f_2(x)] & \text{si } f_1(x) = f_2(x) \\ \nabla f_2(x) & \text{si } f_1(x) < f_2(x) \end{cases}$$

**Exemple 4.4** Soit la fonction  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  définie par  $f(x) = \|x\|_1 = \max\{s^T x \mid s_i \in \{-1, 1\}\}$ . La figure 4.12 présente les différents cas de figures de l’ensemble  $\partial f(x)$  pour  $n = 2$ .

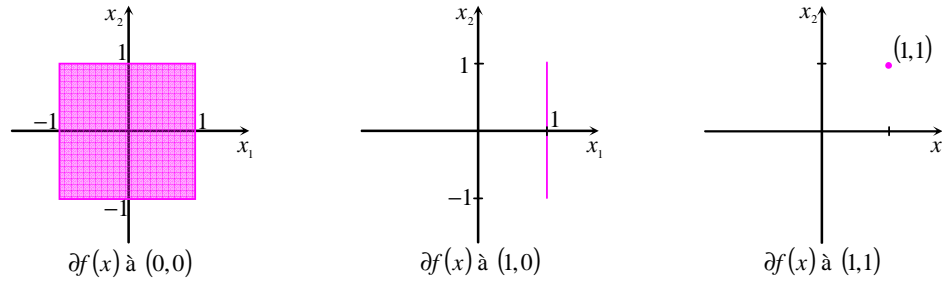


Fig. 4.12 Le sous-différentiel de la fonction  $\|x\|_1$

Même s’il existe des propriétés qui permettent de simplifier la détermination du sous-différentiel, son calcul reste subtil même pour des cas apparemment simples.

### 4.2.1.3. Stationnarité

Également, on peut caractériser un point optimal  $x^*$  par une généralisation de la condition d’Euler-Fermat du cas différentiable,  $\nabla f(x^*) = 0$ . En effet, une application directe des différentes définitions (4-17) et (4-18) donne :

**Théorème 4.1 (Stationnarité : cas non différentiable)** pour  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  convexe non partout différentiable, les trois propriétés suivantes sont équivalentes :

- (i)  $f$  est minimisé en  $x^*$ , i.e. :  $f(y) \geq f(x^*)$  pour tout  $y \in \mathfrak{R}^n$ ,
- (ii)  $0 \in \partial f(x^*)$ ,
- (iii)  $f'(x^*, d) \geq 0$  pour tout  $d \in \mathfrak{R}^n$ .

Par conséquent, un minimum  $x^*$  est caractérisé par  $0 \in \partial f(x^*)$ , ou  $f'(x^*, d) \geq 0$  pour tout  $d \in \mathfrak{R}^n$ . Dans le cas général d’une fonction  $f$  non convexe,  $x^*$  est dit stationnaire si  $0 \in \partial f(x^*)$ . Le point  $(0,0)$  dans le dernier exemple est stationnaire, ceci est facilement vérifiable sur la figure 4.12.

Réciproquement, un point  $x$  non optimal est caractérisé par l’existence d’au moins une direction  $d$  telle que  $f'(x, d) < 0$ . Dans ce cas,  $d$  est une direction de descente. Cette notion est fondamentale pour l’optimisation numérique qu’elle soit différentiable ou non.

## 4.2.2. Méthodes de descente non différentiables

Une fois caractérisé un optimum  $x^*$ , nous nous intéressons au problème de calcul de cet optimum. Comme règle générale, et dans un souci de cohérence, les méthodes non différentiables seront formulées pour approcher le plus possible les algorithmes développés pour les fonctions différentiables. Cela donne des algorithmes plus clairs et met bien en évidence les apports de l’approche “non différentiable partout”. Cela permet également de faciliter une implantation hybride différentiable/non différentiable très efficace lors des mises en œuvre.

Dans les sections suivantes, nous étudierons les algorithmes qui trouvent une direction de descente  $d^{(k)}$  (ou des candidats à directions de descente) et un pas de recherche linéaire  $\alpha^{(k)}$  pour actualiser l’itérée courante,  $x^{(k)}$  (cf. section 3.5.2). A première vue, il pourrait sembler inutile d’analyser séparément le cas non différentiable. Cependant, l’optimisation non différentiable présente quelques pièges dans lesquels un utilisateur non averti pourrait tomber facilement :

Problème du test d’arrêt : l’utilisation du test d’arrêt formel ( $0 \in \partial f(x)$ ) n’est pas implémentable dans le cas général. Ce problème est de nature extrêmement délicate, car la condition  $g(x^{(k)}) \in \partial f(x^{(k)})$  avec  $|g(x^{(k)})| < \varepsilon$ , translatée directement de  $|\nabla f(x^{(k)})| < \varepsilon$ , peut ne jamais être vérifiée. Cette situation arrive même dans des cas très simple : pour la fonction valeur absolue de  $x$ , par exemple,  $|g(x^{(k)})| = 1$  pour tout  $x^{(k)}$  différent de l’optimum 0.

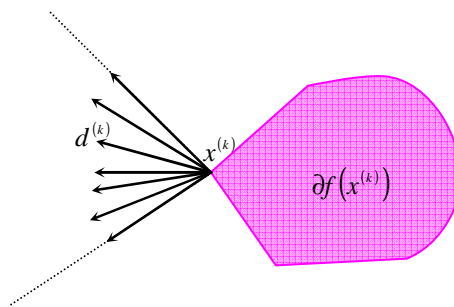
Problème des sous-gradients approchés : parfois le gradient (et même la fonction) n’est pas calculé exactement ; par exemple, il est obtenu par différence finie, à partir des valeurs de la fonction  $f$ . Cette approche n’est plus valable quand le sous-gradient n’est pas continu. Considérons la fonction  $f(x) = x^2$  si  $x \geq 0$  et  $f(x) = -x$  si  $x \leq 0$  : une différence finie autour de 0 résulte en  $(f(h) - f(0))/h = h > 0 \notin \partial f(0) = [-1, 0]$ .

Problème de non différentiabilité : nous savons déjà que l’application  $x \rightarrow \partial f(x)$  n’étant pas continue, une petite variation en  $x^{(k)}$  peut donner des grandes variations en  $\partial f(x^{(k)})$ . Basé sur l’information disponible du sous-différentiel, le calcul de  $d^{(k)}$  peut varier énormément et produire des  $x^{(k+1)}$  très différents. Ce phénomène arrive aussi en exécutant le même programme sur différents processeurs : les erreurs arrondies font que les suites obtenues sur différentes machines ne sont plus comparables ! Bien entendu, ce problème incontournable rend très difficile toute comparaison numérique.

D’un point de vue géométrique, trouver une direction de descente correspond à trouver un hyperplan séparant strictement les ensembles (convexes fermés)  $\partial f(x)$  et  $\{0\}$ . En effet, une conséquence directe de (4-17) et (4-18) permet de trouver le résultat suivant :

**Théorème 4.2 (Direction de descente)** *une direction de descente  $d$  est telle que, pour  $\alpha \in [f'(x, d), 0[$  l’hyperplan  $\{z \in \mathbb{R}^n : z^T d = \alpha\}$  sépare strictement  $\partial f(x)$  et  $\{0\}$  :  $s^T d \leq \alpha < 0$  pour tout  $s \in \partial f(x)$ .*

La figure 4.13 montre les directions  $d^{(k)}$  de descente pour  $f$  en  $x^{(k)}$  ; on observera que chacune d’elles forme un angle obtus avec tout les éléments de l’ensemble  $\partial f(x^{(k)})$ .



**Fig. 4.13 Direction de descente**

Une première idée pour trouver une direction de descente  $d^{(k)}$  consiste à choisir à chaque itération la direction qui nous donne la plus grande des descentes possibles (cf. section 3.5.2) qui vérifie :

$$d^{(k)} = \arg\left(\min_{\|d\|=1} f'(x^{(k)}, d)\right) = \arg\left(\min_{\|d\|=1} \left(\max_{s \in \partial f(x^{(k)})} (s^T d)\right)\right) \tag{4-22}$$

Cet algorithme “steepest descent algorithm” présente deux inconvénients majeurs qui le rendent très peu efficace :

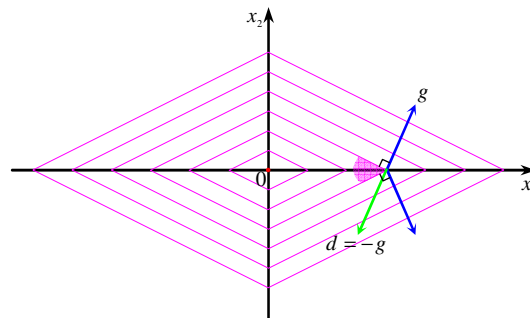
La suite  $\{x^{(k)}\}$  peut osciller indéfiniment et sans converger. Ce phénomène de *zigzag* est évident pour les problèmes mal conditionnés où la sensibilité par rapport aux paramètres est disparate. La non différentiabilité ne fait que l’amplifier.

Le calcul de direction de descente requiert la connaissance de tout le sous-différentiel, et ceci pour chaque itération. Cette exigence est excessive, voire impossible, dans la plus part des cas d’application pratiques.

Afin de contourner cette deuxième difficulté, certains algorithmes de descente se contentent d’une exigence bien plus faible : ils nécessitent de fournir, pour chaque paramètre  $x$ , la valeur  $f(x)$  et d’un sous-gradient arbitraire  $g(x) \in \partial f(x)$ .

L’algorithme du sous-gradient proposé par Shor [Sho85] est l’un des algorithmes les plus connus dans cette famille d’algorithmes. Il consiste à choisir une direction d’arrêt et une direction de descente normalisée de type gradient, c’est-à-dire  $\|g(x^{(k)})\| < \varepsilon$  et  $d^{(k)} = -g(x^{(k)})/\|g(x^{(k)})\|$ .

Or, dans le cas non différentiable, une direction opposée à un sous-gradient n’est pas forcément une direction de descente. En effet, on a vu sur la figure 4.13 pour qu’une direction soit de descente, elle doit former un angle obtus avec tout le sous-différentiel, et non avec le seul élément  $g(x^{(k)})$  choisi. La figure 4.14 montre des courbes de niveau pour une fonction minimisée en 0. La région ombrée indique les directions de descente.



**Fig. 4.14** Direction suggérée par un sous-gradient

On remarque que pour cet exemple simple,  $f(x_1, x_2) = |x_1| + 2|x_2|$ , la direction opposée à  $g^T = (1, 2) \in \partial f((\cdot, 0))$  n’est pas une direction de descente. Ainsi, le choix du sous-gradient paraît primordial pour cet algorithme.

Toutefois, on montre que le schéma itératif basé sur de telles directions converge vers la solution  $x^*$  du problème, à condition toutefois d’utiliser un pas adéquat [Sho85]. Les conditions de convergence de cet algorithme imposent une convergence lente, si bien que cette méthode n’est pas recommandée en pratique [Pol87].

Dans la section suivante, nous proposons d’utiliser une extension de la notion du sous-différentiel, proposée par F.H. Clarke [Cla83], afin de contourner les problèmes de calcul du sous-différentiel et du critère d’arrêt pour les méthodes de descente non différentiables.

### 4.2.3. Le sous-différentiel de Clarke est ses propriétés



L’objectif de cette extension est de développer un algorithme de descente pour les problèmes non différentiables, qui permet de générer une suite d’itérées  $\{x^{(k)}\}$  tout en assurant une décroissance de la suite correspondante  $\{f(x^{(k)})\}$ , et qui possède un point d’accumulation  $\bar{x}$  stationnaire.

Soit une fonction  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  est localement Lipschitz continue et de dérivée continue presque partout sur un sous-ensemble dense et ouvert  $D \in \mathfrak{R}^n$ . En plus, on suppose qu’il existe un point  $x^* \in \mathfrak{R}^n$  pour lequel l’ensemble  $\mathfrak{K} = \{x \mid f(x) \leq f(x^*)\}$  est compact. Sous ses hypothèses, on introduit l’approximation du sous-différentiel de Clarke [Cla83] comme suit :

**Définition 4.1 (Le sous-différentiel de Clarke) :** pour chaque  $\varepsilon > 0$ , on définit le sous-différentiel au sens de Clarke qu’on note  $\bar{\partial}f(x)$  par :

$$\begin{cases} \bar{\partial}f(x) = \bigcap_{\varepsilon > 0} G_\varepsilon(x) & \text{avec} \\ G_\varepsilon : \mathfrak{R}^n \rightarrow \mathfrak{R}^n \setminus \{ \} & G_\varepsilon(x) = clconv(\nabla f(x + \varepsilon B) \cap D) \end{cases} \quad (4-23)$$

où  $B = \{x \mid \|x\|_2 \leq 1\}$  est l’hyperboule unitaire fermée,  $clconv(\cdot)$  est l’opérateur enveloppe convexe fermée et  $G_\varepsilon(x)$  une multifonction définissant un ensemble d’approximation du sous-différentiel de Clarke  $\bar{\partial}f(x)$ .

Cette définition montre que le sous-différentiel de Clarke  $\bar{\partial}f(x)$  peut être défini comme étant l’intersection de tous les ensembles  $G_\varepsilon(x)$  définis pour tout  $\varepsilon > 0$ . Autrement dit,  $\bar{\partial}f(x)$  s’interprète comme la limite de l’ensemble  $G_\varepsilon(x)$  lorsque  $\varepsilon$  tend vers 0.

Nous introduisons également la notion de  $\varepsilon$ -sous-différentiel introduite par Goldstein [Gol77]

**Définition 4.2 (Le  $\varepsilon$ -sous-différentiel de Clarke) :** pour chaque  $\varepsilon > 0$ , on définit le  $\varepsilon$ -sous-différentiel de Clarke qu’on note  $\bar{\partial}_\varepsilon f(x)$  par :

$$\bar{\partial}_\varepsilon f(x) = clconv(\bar{\partial}f(x + \varepsilon B)) \quad (4-24)$$

De ces deux définitions, il est clair que  $G_\varepsilon(x) \subset \bar{\partial}_\varepsilon f(x)$  car le  $\varepsilon$ -sous-différentiel de Clarke est défini à base des sous-différentiels de Clarke qui prennent en compte les zones non différentiables de  $f$ .

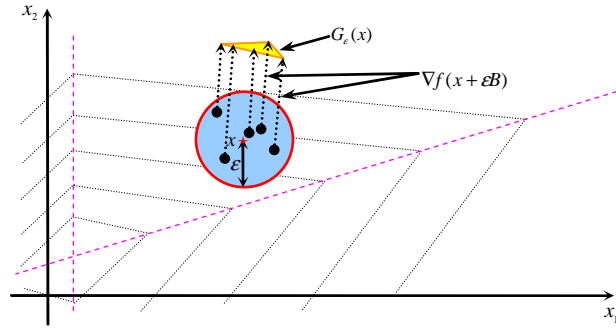
Pour tout  $0 < \varepsilon_1 < \varepsilon_2$ , on vérifie la propriété d’emboîtement suivante :

$$\bar{\partial}_{\varepsilon_1} f(x) \subset G_{\varepsilon_2}(x) \subset \bar{\partial}_{\varepsilon_2} f(x) \quad (4-25)$$

Par conséquent,  $\bar{\partial}_\varepsilon f(x)$  peut être approximé inférieurement par  $G_\varepsilon(x)$  et sous l’hypothèse de différentiabilité presque partout de  $f$ , l’approximation du  $\varepsilon$ -sous-différentiel de Clarke  $G_\varepsilon(x)$  peut être faite à partir d’un nombre fini  $m$  de gradients [Bur02, Bur03, Bur04]. On écrit alors :

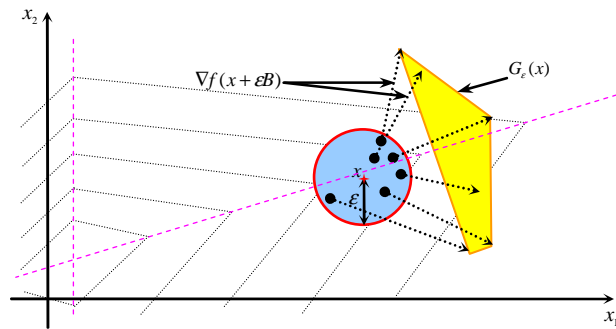
$$G_\varepsilon(x) \approx clconv\left(\bigcup_{i=1}^m (\nabla f(x + b_i) \cap D)\right) \text{ avec } \|b_i\|_2 \leq \varepsilon \quad (4-26)$$

Loin des zones non différentiables et pour un petit rayon  $\varepsilon$ , l’ensemble  $G_\varepsilon(x)$  devient très petit car les différents points échantillonnés possèdent approximativement le même gradient (cf. figure 4.15)



**Fig. 4.15** Approximation d'un  $\varepsilon$ -sous-différentiel de Clarke (cas différentiable)

La figure ci-dessous illustre un exemple de construction de l'ensemble  $G_\varepsilon(x)$  autour d'une zone non différentiable de la fonction  $f$ . Pour un rayon  $\varepsilon$  petit donné, les gradients échantillonnés sont de part et d'autres de la zone non différentiable. Ceci entraîne une dispersion de la direction des gradients et donc un large ensemble  $G_\varepsilon(x)$ .



**Fig. 4.16** Approximation d'un  $\varepsilon$ -sous-différentiel de Clarke (cas non différentiable)

Afin d'assurer une sélection homogène des gradients à l'intérieur de l'hyper-boule  $B(x, \varepsilon)$  de centre  $x$  et de rayon  $\varepsilon$ , une solution simple serait de choisir un échantillonnage qui suit une loi de distribution uniforme  $U_{[0,1]}$ . Ce point sera étudié en détail dans le chapitre 5.

**Définition 4.3 (L' $\varepsilon$ -stationnarité au sens de Clarke) :** On dit qu'un point  $x$  est Clarke  $\varepsilon$ -stationnaire (ou  $\varepsilon$ -stationnaire au sens de Clarke) si  $0 \in \bar{\partial}_\varepsilon f(x)$ .

Cette notion de  $\varepsilon$ -stationnarité de Clarke représente un point clé de notre approche car elle permet moyennant l'approximation du sous-différentiel d'évaluer la stationnarité d'une itérée et de déjouer ainsi la difficulté du test d'arrêt. Pour ce faire, nous introduisons un critère pour mesurer la proximité de l' $\varepsilon$ -stationnarité de Clarke. Il s'agit de la distance entre le point  $x$  et son  $\varepsilon$ -sous-différentiel de Clarke approché  $G_\varepsilon(x)$  :

$$\rho_\varepsilon(x) = \text{dist}(0 / G_\varepsilon(x)) \geq \text{dist}(0 / \bar{\partial}_\varepsilon f(x)) \quad (4-27)$$

Le  $\varepsilon$ -sous-gradient de Clarke (élément du  $\varepsilon$ -sous-différentiel de Clarke), qu'on note  $g$ , correspondant à cette distance  $\rho_\varepsilon(x)$  est un bon candidat pour un algorithme de descente (cf. figure 4.17(a)). En effet, la direction opposée à ce  $\varepsilon$ -sous-gradient de Clarke garantit la condition de

descente du théorème 4.2 et elle forme ainsi un angle obtus avec tous les éléments de l’ensemble  $G_\varepsilon(x)$  approximant  $\bar{\partial}_\varepsilon f(x)$ .

L’ensemble  $G_\varepsilon(x)$  étant convexe, le problème de détermination de cette distance  $\rho_\varepsilon(x)$  et du  $\varepsilon$ -sous-gradient de Clarke  $g_\varepsilon$  correspondant est un problème d’optimisation quadratique qui peut être résolu efficacement et rapidement :

$$g = \arg(\min_{s \in G_\varepsilon} \|s\|_2) \quad (4-28)$$

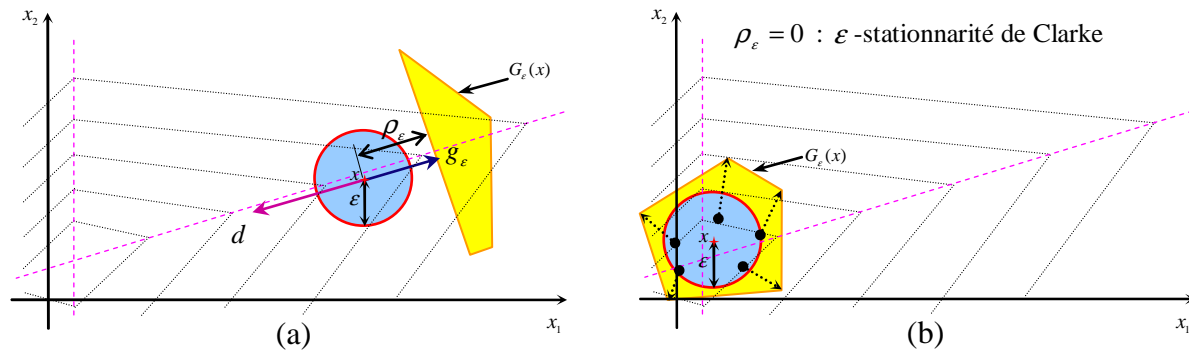


Fig. 4.17 L’  $\varepsilon$ -stationnarité au sens de Clarke

La figure 4.17(b) décrit le cas où la distance  $\rho_\varepsilon(x)=0$ , ceci implique que l’  $\varepsilon$ -stationnarité de Clarke est atteinte pour une résolution défini par le rayon  $\varepsilon$ . Plus le rayon  $\varepsilon$  est petit, plus la précision est meilleure. Il serait donc judicieux de décroître ce rayon à chaque fois que l’algorithme soit près de l’  $\varepsilon$ -stationnarité de Clarke et ceci pour accélérer la convergence et se rapprocher avec plus de précision de la solution optimale.

#### Remarque :

Il est important de préciser l’attribut de la définition du  $\varepsilon$ -sous-différentiel car on en rencontre plusieurs dans la littérature. A titre d’exemple, dans [Hir93] et [bon97], une définition différente au  $\varepsilon$ -sous-différentiel est introduite, elle est donnée par :

$$\partial_\varepsilon f(x) = \{s \in \mathfrak{R}^n : f(y) \geq f(x) + s^T(y-x) - \varepsilon, \forall y \in \mathfrak{R}^n\} \quad (4-29)$$

Ainsi, nous précisons que toutes les notions qui ont été introduites dans la section 4.2.3 sont au sens de Clarke [Cla83] et que dorénavant, le qualificatif “de Clarke” sera omis afin d’alléger le texte.

#### 4.2.4. Algorithme du epsilon-sous-différentiel (AESD)

Il est maintenant possible de construire un algorithme de descente en se basant sur une approximation  $G_\varepsilon(x)$  du  $\varepsilon$ -sous-différentiel et sur l’  $\varepsilon$ -stationnarité.

L’évolution de cet algorithme de descente présentera certainement deux phases qui seront mises plus en avant lors des tests numériques :

- La première phase est similaire à celle d’un algorithme de plus profonde descente, elle se produit quand le point courant est différentiable et loin d’une zone non différentiable. Dans ce cas, l’ensemble  $G_\varepsilon(x)$  est petit et le  $\varepsilon$ -sous-gradient calculé ne diffère pas vraiment du gradient.
- La deuxième phase se produit près des zones non différentiables qui sont souvent attractrices car elles présentent des points stationnaires presque dans toutes les directions. Dans ce cas, la direction suggérée par le  $\varepsilon$ -sous-gradient  $g$  est presque parallèle à celle de la zone non différentiable. Ceci permet d’accélérer la convergence dans cette direction, chose impossible avec des approches ponctuelles simples de type gradient ou sous-gradient (cf. section 4.2.2).

Par la suite, nous exposons un premier algorithme de descente qui permet d’exploiter l’ensemble des propriétés décrites dans la section 4.2.3. On l’appellera algorithme du  $\varepsilon$ -sous-différentiel (AESD). Ce dernier est une variante de l’algorithme proposé par Burke et al. [Bur04] où plusieurs modifications ont été apportées afin de l’adapter aux problèmes de cahier des charges et à améliorer ses performances. Ces évolutions qui concernent surtout l’aspect implémentation et mise en œuvre numérique seront précisées tout au long de la description de cet algorithme.

D’abord, notons que nous avons préféré appeler cet algorithme, **Algorithme du Epsilon-Sous-Différentiel** car nous estimons que ce nom est plus approprié (par rapport à Sampling Gradient Algorithm) et permet d’éviter toute confusion avec les méthodes d’optimisation stochastiques à base d’échantillonnage.

#### 4.2.4.1. Notations

Afin de faciliter la lecture et l’analyse des algorithmes, nous précisons dans ce glossaire les principales notations utilisées :

- $k$  : L’indice des itérations.
- $n$  : La dimension du vecteur paramètre.
- $x^{(k)}$  : L’itération courante.
- $\tau$  : Le facteur de réduction pour le rebroussement dans une recherche linéaire de type Armijo.
- $\mathfrak{S} : \{x \mid f(x) \leq f(x^*)\}$ .
- $u_j^{(k)}$  : Le  $j^{\text{ième}}$  échantillon à l’intérieur de l’hyper-boule unitaire  $B$  à l’itération  $k$ .
- $\omega_1$  : Le facteur d’Armijo.
- $B$  : L’hyper-boule unitaire fermée.
- $\varepsilon_k$  : Le rayon d’échantillonnage.
- $v_k$  : La tolérance d’optimalité.
- $\mu$  : Le facteur de réduction pour le rayon d’échantillonnage.
- $\theta$  : Le facteur de réduction de la tolérance d’optimalité  $v_k$ .
- $m$  : Le nombre d’échantillons.
- $D$  : L’ensemble des points différentiables.
- $x_j^{(k)}$  : Les points échantillonnés.
- $g^{(k)}$  : Le plus petit  $\varepsilon$ -sous-gradient de Clarke à l’itération  $k$  pour le rayon  $\varepsilon_k$ .
- $G^{(k)}$  : L’approximation du  $\varepsilon$ -sous-différentiel de Clarke à l’itération  $k$  pour le rayon  $\varepsilon_k$ .
- $d^{(k)}$  : La  $k^{\text{ième}}$  direction de recherche.

$\alpha^{(k)}$  : La  $k^{\text{ième}}$  longueur de pas.

$N_{\max}$  : Le nombre maximal de bisections lors de la recherche linéaire.

$r$  : Le compteur de la recherche linéaire.

#### 4.2.4.2. AESD

Dans le souci de faciliter la lecture et l’analyse de la stratégie d’optimisation, nous avons préféré de la décrire sous une forme algorithmique émaillée de quelques commentaires tout en soulignant les différentes étapes de cette méthode de descente.

**Algorithme AESD**

**Etape 0 :** (Initialisation)

Soit  $x^{(0)} \in \mathcal{S} \cap D$ ,  $\tau \in ]0,1[$ ,  $\omega_1 \in ]0,1[$ ,  $\varepsilon_0 > 0$ ,  $v_0 \geq 0$ ,  $\mu \in ]0,1[$ ,  $\theta \in ]0,1[$ ,  $k=0$ ,  $r=0$  et  $m \in \{n+1, n+2, \dots\}$ .

**Etape 1 :** (Approximation du  $\varepsilon$ -sous-différentiel de Clarke  $G^{(k)}$ )

Soient  $u_1^{(k)}, u_2^{(k)}, \dots, u_m^{(k)}$  des échantillons uniformes de  $B$ , // Échantillonnage  
 Faire  $x_0^{(k)} = x^{(k)}$  et  $x_j^{(k)} = x^{(k)} + \varepsilon_k u_j^{(k)}$  pour  $j=1, \dots, m$ ,  
 Si l’un des échantillons ( $j=1, \dots, m$ ) vérifie  $x_j^{(k)} \notin D$  // Test de différentiabilité  
     Aller à l’étape 1.  
 Sinon  
     Faire  $G^{(k)} = clconv\{\nabla f(x_0^{(k)}), \nabla f(x_1^{(k)}), \dots, \nabla f(x_m^{(k)})\}$ , // Approximation du  $\varepsilon$ -sous-différentiel  
 Fin Si

**Etape 2 :** (Calcul de la direction de descente  $d^{(k)}$ )

Soit  $g^{(k)} \in G_k$  la solution du problème quadratique positif :  $g^{(k)} = \arg(\min_{g \in G_k} \|g\|_2)$ ,  
 Si  $\|g^{(k)}\|_2 = 0$   
     Fin de l’algorithme ( $\varepsilon_k$ -stationnarité de Clarke).  
 Sinon  
     Si  $\|g^{(k)}\|_2 \leq v_k$   
         Faire  $\alpha^{(k)} = 0$ ,  $v_{k+1} = \theta v_k$  et  $\varepsilon_{k+1} = \mu \varepsilon_k$ ,  
         Aller à l’étape 4.  
     Sinon  
          $v_{k+1} = v_k$ ,  $\varepsilon_{k+1} = \varepsilon_k$  et  $d^{(k)} = -g^{(k)} / \|g^{(k)}\|_2$ , // Direction de descente  
     Fin Si  
 Fin Si

**Etape 3 :** (Calcul du pas d’optimisation  $\alpha^{(k)}$ )

Faire  $\alpha^{(k)} = 1$ , // Règle d’Armijo avec rebroussement  
 Tant que  $f(x^{(k)} + \alpha^{(k)} d^{(k)}) \geq f(x^{(k)}) - \omega_1 \alpha^{(k)} \|g^{(k)}\|_2$   
     Faire  $\alpha^{(k)} = \tau \cdot \alpha^{(k)}$  et  $r = r + 1$ ,  
     Si  $r = N_{\max}$  // Test du nombre maximal de réduction  
         Faire  $\alpha^{(k)} = 0$ ,  
         Arrêt Tant que,  
     Fin Si  
 Fin Tant que

**Etape 4 :** (Actualisation)

Si  $x^{(k)} + \alpha^{(k)}d^{(k)} \in D$  // Test de différentiabilité  
 Faire  $x^{(k+1)} = x^{(k)} + \alpha^{(k)}d^{(k)}$  et  $k = k + 1$ ,  
 Aller à l’étape 1.  
 Sinon  
 Choisir  $\hat{x}^{(k)} \in \hat{x}^{(k)} + \varepsilon_k B$  tel que :  $\hat{x}^{(k)} + \alpha^{(k)}d^{(k)} \in D$  et  $f(\hat{x}^{(k)} + \alpha^{(k)}d^{(k)}) < f(x^{(k)}) - \omega_1 \alpha^{(k)} \|g^{(k)}\|_2$ ,  
 Faire  $x^{(k+1)} = \hat{x}^{(k)} + \alpha^{(k)}d^{(k)}$  et  $k = k + 1$ ,  
 Aller à l’étape 1.  
 Fin Si

Inspiré des travaux de Burke, et al. [Bur04], cet algorithme sera étudié en détail. Il représentera notre noyau autour duquel une librairie contenant plusieurs stratégies d’optimisation sera développée.

D’un point de vu structurel, l’algorithme AESD comprend une étape supplémentaire par rapport à un algorithme de descente classique ; c’est l’étape 1. Dans cette dernière, on approxime le  $\varepsilon$ -sous-différentiel par le calcul d’un ensemble convexe fermée  $G^{(k)}$  en se basant sur un nombre  $m$  fini de gradients. Nous revenons à ce point dans la section 4.3 pour étudier son influence sur la progression de l’algorithme via des applications numériques et avec un peu plus de détail dans le chapitre 5 où la qualité de l’approximation du  $\varepsilon$ -sous-différentiel est étudiée. Pour le moment, nous choisissons  $m > n$  et nous calculons les gradients soit analytiquement (pour des problèmes simples), soit par différences finies. Ce calcul sera le principal objectif du chapitre 5.

Même si les zones non différentiables sont de mesure nulle, une précaution est prise afin d’éviter l’échantillonnage d’un point  $x_j^{(k)}$  non différentiable. Ce risque devient petit à petit non négligeable près d’une “faille” de la fonction critère. C’est pourquoi une modification est apportée à l’algorithme de base de Burke et al. en exploitant son caractère non répétitif afin de relancer l’étape 1 et ré-estimer l’ $\varepsilon$ -sous-différentiel de Clarke  $G^{(k)}$ .

Numériquement, dans cette première étape, la structure de l’algorithme est choisie afin d’éviter le recours à une boucle “For” ; au lieu d’échantillonner et de tester la dérivabilité pour chaque point  $x_j^{(k)}$ , nous générons  $m$  échantillons  $u_j^{(k)}$  et nous testons la dérivabilité du vecteur  $x^{(k)}$  une seule fois en exploitant la propriété du calcul vectoriel sur Matlab. Cette technique permet de réduire considérablement le temps d’exécution de l’algorithme.

La seconde étape consiste à calculer la direction de descente  $d^{(k)}$ . Le choix est fait sur le plus petit  $\varepsilon$ -sous-gradient en module. Ce choix est motivé par deux raisons :

- La direction opposée à  $g^{(k)}$  est une bonne direction de descente ; elle permet une descente sans zigzag autour de la zone non différentiable (cf. section 4.3).
- Le module  $\|g^{(k)}\|_2$  est égal à la distance minimale entre le point courant et son  $\varepsilon$ -sous-différentiel approximé,  $\|g^{(k)}\|_2 = \rho_\varepsilon(x^{(k)})$ , ce qui représente une bonne mesure de la stationnarité.

Le problème quadratique qui permet de calculer  $g^{(k)}$  se déduit en exprimant sa contrainte en utilisant l’équivalence suivante :

$$g \in G_k \Leftrightarrow \{\exists \lambda \in \mathbb{R}^m : g = S \cdot \lambda, e^T \lambda = 1 \text{ et } \lambda \geq 0 \text{ avec } S = [\nabla f(x_0^{(k)}), \nabla f(x_1^{(k)}), \dots, \nabla f(x_m^{(k)})]\}$$

où  $e^T = [1, \dots, 1]$  et  $\lambda \geq 0 \Leftrightarrow \{\lambda_i \geq 0 \text{ pour } i = 1, \dots, m\}$ .

Ce qui revient à chercher le vecteur de coefficients positifs  $\lambda$  qui solutionne le problème quadratique suivant :

$$\begin{cases} \min \lambda^T (S^T S) \lambda \\ e^T \lambda = 1 \text{ et } \lambda \geq 0 \end{cases} \quad (4-30)$$

Finalement, l’optimum obtenu  $\lambda^*$  est global et le plus petit  $\varepsilon$ -sous gradient est donné par  $g^{(k)} = S \cdot \lambda^*$ .

Dans sa version implémentée sous Matlab l’algorithme AESD, l’utilisateur dispose de 3 fonctions de 3 boites à outils différentes pour la résolution de ce sous-problème d’optimisation quadratique : “quadprog” de la boite à outils Optimisation de Matlab, “sedumi” de la boite à outils SeDuMi et “mosekopt” de la boite à outils MOSEK. Les tests numériques montrent que cette dernière est plus efficace et stable numériquement. En l’occurrence, elle est recommandée dans la version par défaut du programme que nous avons développé.

Ce calcul est suivi par le test d’arrêt (de  $\varepsilon$ -stationnarité),  $\|g^{(k)}\|_2 = 0$ . Il est clair que cette condition d’arrêt théorique est traduite numériquement par une limite très petite. Dans notre programme elle est égale à la plus petite variation possible en virgule flottante donnée par la fonction “eps” sous Matlab.

D’autres tests d’arrêt peuvent être ajoutés afin de limiter le nombre d’itérations ou le nombre d’évaluations de la fonction critère  $f$  si la convergence devient lente.

La direction de descente  $d^{(k)}$  choisie est normée afin de faciliter la recherche linéaire près des zones non différentiables.

Pour le rayon d’échantillonnage  $\varepsilon_k$ , le facteur de réduction  $\mu$  est choisi a priori afin d’adapter la convergence vers un optimum. Son effet ressemble à un zoom autour d’un point  $\varepsilon$ -stationnaire qui permet d’échantillonner plus près et de raffiner ainsi le résultat d’optimisation. Par défaut, l’algorithme AESD prend,  $\mu = 0,1$  et  $\varepsilon_0 = 0,1$ .

Le facteur de réduction  $\theta$  est introduit afin de réduire la tolérance d’optimalité  $v_k$  à chaque itération  $k$ . Sa valeur conditionne la vitesse de convergence de l’algorithme. Son choix dépend du choix de la tolérance initial  $v_0$ . Dans notre cas, le choix  $(v_0, \mu) = (10^{-6}, 1)$  est fait.

Dans la troisième étape, on calcule le pas de la descente  $\alpha^{(k)}$ . La règle d’Armijo avec rebroussement est choisie avec  $\omega_1 = 10^{-4}$ ,  $\tau = 0,5$  et  $N_{\max} = 100$  (cf. section 3.5.2.2.2). Le pas initial  $\alpha^{(0)}$  est, par défaut, mis à 1. L’utilisateur aura le libre choix de l’échanger et même de choisir une des règles de recherche linéaire exposées dans la section 3.5.2.2. Ces dernières ont été toutes implémentées dans l’algorithme AESD. Cependant, il est déconseillé d’utiliser la règle de Wolfe forte pour les problèmes d’optimisation non différentiables [Noc99].

L’algorithme AESD se termine par une phase d’actualisation durant laquelle on veille toujours à ce que le nouveau point  $x^{(k+1)}$  soit différentiable.

Les paramètres utilisés dans AESD sont regroupés dans une seule structure qui permet à l’utilisateur de les modifier a priori afin de les adapter à son problème d’optimisation.

L’algorithme AESD a été implémenté sous Matlab en utilisant les boites à outils MOSEK et SeDuMi qui présentent une bonne stabilité numérique pour la résolution des problèmes quadratiques rencontrés lors de la phase de détermination de la direction de descente. Un ensemble de résultats numériques sera présenté dans la section 4.3.

#### 4.2.5. Algorithme du epsilon-sous-différentiel modifié (AESDM)

À chaque itération, le pari dans l’algorithme AESD s’est fait sur une approximation du sous-différentiel afin d’en déduire une bonne direction de descente qui permet de réduire au mieux le critère à optimiser. Ce pari paraît tout de même excessif car il engendre un temps de calcul important dû principalement à l’évaluation et au stockage en mémoire des  $m$  gradients à chaque itération de l’algorithme. Ainsi le choix fait dans AESD risque de réduire considérablement sa performance et son efficacité même pour des problèmes d’optimisation différentiable simples.

Il serait donc plus intéressant de faire appel à l’approximation du  $\varepsilon$ -sous-différentiel seulement lorsque les méthodes de descente classiques échouent ; particulièrement près des zones non différentiables.

Il est clair que le choix du paramètre  $m=1$ , transforme la méthode AESD en un simple algorithme de plus profonde descente, ce qui est généralement suffisant lorsque les itérées sont loin d’une zone non différentiable. En effet, la méthode du gradient est robuste et bien adaptée à la phase initiale des recherches, loin de l’optimum (cf. section 3.5.2.1.1). En outre, le calcul, dans ce cas, de plus d’un gradient engendrera une redondance inutile dans l’information car les gradients échantillonnés sont très peu différents et le sous-problème d’optimisation pour la détermination du  $\varepsilon$ -sous-gradient  $g^{(k)}$  devient injustifié.

Ainsi, nous procédons à une modification de l’algorithme AESD afin d’améliorer ses performances en termes de temps de calcul tout en conservant son efficacité près des zones non différentiables du critère. Cette modification consiste à initialiser l’algorithme comme pour une simple profonde descente ( $m=1$ ) et à commuter vers l’algorithme AESD ( $m>n$ ) dès que le calcul du pas d’optimisation échoue (atteindre le nombre maximal de réduction du pas  $r=N_{\max}$ ) trois fois consécutivement. À ce moment là, nous estimons que la stratégie du gradient n’est plus efficace et que les directions calculées sont mal orientées ou qu’elles ne sont pas des directions de descente (cf. théorème 4.2, figure 4.13 et figure 4.14). Les résultats présentés dans la section 4.3 montrent que ce simple test de commutation entre les deux stratégies de calcul de la direction de descente est efficace pour améliorer le temps d’exécution des problèmes d’optimisation et présente un bon critère de proximité pour les zones non différentiables et les bandes de convergence lente (si le problème est mal conditionné). Dans la section 4.3, nous présenterons un autre critère, plutôt d’analyse cette fois, qui est basé sur la notion d’ouverture d’un champ de vecteur et qui permet de mesurer à chaque itération de l’algorithme la dispersion des gradients échantillonnés et le caractère localement non différentiable d’un critère.

L’algorithme modifié est nommé AESDM, il conserve la même structure que AESD et présente quelques modifications qui touchent surtout à différencier les deux stratégies (à base de gradient et à base du  $\varepsilon$ -sous-différentiel) afin d’éviter des calculs inutiles comme dans le cas de la deuxième étape. Le nombre maximal  $N_{\max}$  des itérations pour la recherche linéaire qui était de 100 dans AESD (ce qui est largement suffisant pour obtenir un bon pas via la règle d’Armijo) sera réduit à  $N_{\max} = 30$  pour la



phase d’initialisation afin de détecter les moins bonnes directions de descente et d’éviter ainsi les petits pas.

#### 4.2.5.1. Notations

Outre les paramètres de l’algorithme AESD, les paramètres supplémentaires suivants sont utilisés :

$Cpt$  : Le compteur d’itérations manquées de l’algorithme du gradient.

$\bar{N}$  : Le nombre maximal d’itérations manquées pour l’algorithme du gradient (par défaut  $\bar{N} = 3$ ).

#### 4.2.5.2. AESDM

La forme algorithmique de la méthode est donnée comme suit :

**Algorithme AESDM**

**Etape 0 :** (Initialisation)

Soit  $x^{(0)} \in \mathfrak{S} \cap D$ ,  $\tau \in ]0,1[$ ,  $\omega_1 \in ]0,1[$ ,  $\varepsilon_0 > 0$ ,  $\nu_0 \geq 0$ ,  $\mu \in ]0,1]$ ,  $\theta \in ]0,1]$ ,  $k=0$ ,  $m=1$ ,  $r=0$ ,  $Cpt=0$  et  $\bar{N}=3$ .

**Etape 1 :** (Approximation du  $\varepsilon$ -sous-différentiel de Clarke  $G^{(k)}$ )

Si  $m \neq 1$  // Test de la méthode de descente

$x_0^{(k)} = x^{(k)}$ ,

Soient  $u_1^{(k)}, u_2^{(k)}, \dots, u_m^{(k)}$  des échantillons uniformes de  $B$ ,

Faire  $x_0^{(k)} = x^{(k)}$  et  $x_j^{(k)} = x^{(k)} + \varepsilon_j u_j^{(k)}$  pour  $j=1, \dots, m$ ,

Si l’un des échantillons ( $j=1, \dots, m$ ) vérifie  $x_j^{(k)} \notin D$  // Test de différentiabilité

Aller à l’étape 1.

Sinon

Faire  $G^{(k)} = clconv\{\nabla f(x_0^{(k)}), \nabla f(x_1^{(k)}), \dots, \nabla f(x_m^{(k)})\}$ ,

Fin Si

Fin Si

**Etape 2 :** (Calcul de la direction de descente  $d^{(k)}$ )

Si  $m=1$  // Méthode de plus profonde descente

$g^{(k)} = \nabla f(x^{(k)})$ ,

Sinon // Méthode AESD

Soit  $g^{(k)} \in G_k$  la solution du problème quadratique positif :  $g^{(k)} = \arg(\min_{g \in G_k} \|g\|_2)$ ,

Fin Si

Si  $\|g^{(k)}\|_2 = 0$

Fin de l’algorithme.

Sinon

Si  $\|g^{(k)}\|_2 \leq \nu_k$

Faire  $\alpha^{(k)} = 0$  et  $\nu_{k+1} = \theta \nu_k$ ,

Si  $m=1$

Faire  $\varepsilon_{k+1} = \varepsilon_k$ ,

Sinon

Faire  $\varepsilon_{k+1} = \mu \varepsilon_k$ ,

Fin Si

<p>           Aller à l'étape 4.            Sinon                Faire <math>v_{k+1} = v_k</math> et <math>\varepsilon_{k+1} = \varepsilon_k</math>,                Faire <math>d^{(k)} = -g^{(k)} / \ g^{(k)}\ _2</math>,            Fin Si            Fin Si         </p> <p> <u>Etape 3 :</u> (Calcul du pas d'optimisation <math>\alpha^{(k)}</math>)            Faire <math>\alpha^{(k)} = 1</math>,            Tant que <math>f(x^{(k)} + \alpha^{(k)} d^{(k)}) \geq f(x^{(k)}) - \omega_1 \alpha^{(k)} \ g^{(k)}\ _2</math>                Faire <math>\alpha^{(k)} = \tau \cdot \alpha^{(k)}</math> et <math>r = r + 1</math>,                Si <math>r = N_{\max}</math>                    Si <math>m = 1</math>                        Faire <math>Cpt = Cpt + 1</math>,                        Si <math>Cpt = \bar{N}</math>                            <math>m \in \{n+1, n+2, \dots\}</math>,                        Fin Si                    Fin Si                Faire <math>\alpha^{(k)} = 0</math>,                Arrêt Tant que,            Fin Si            Fin Tant que         </p> <p> <u>Etape 4 :</u> (Actualisation)            Si <math>x^{(k)} + \alpha^{(k)} d^{(k)} \in D</math>                Faire <math>x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}</math> et <math>k = k + 1</math>,                Aller à l'étape 1.            Sinon                Choisir <math>\hat{x}^{(k)} \in \hat{x}^{(k)} + \varepsilon_k B</math> tel que : <math>\hat{x}^{(k)} + \alpha^{(k)} d^{(k)} \in D</math> et <math>f(\hat{x}^{(k)} + \alpha^{(k)} d^{(k)}) &lt; f(x^{(k)}) - \omega_1 \alpha^{(k)} \ g^{(k)}\ _2</math>,                Faire <math>x^{(k+1)} = \hat{x}^{(k)} + \alpha^{(k)} d^{(k)}</math> et <math>k = k + 1</math>,                Aller à l'étape 1.            Fin Si         </p>	<p>// Direction de descente</p> <p>// Règle d'Armijo avec rebroussement</p> <p>// Test du nombre maximal de réduction</p> <p>// Test de commutation entre méthodes</p> <p>// Test de différentiabilité</p>
--	--

Les paramètres numériques du nouvel algorithme sont, par défaut, les mêmes que ceux utilisés dans l'algorithme AESD. L'algorithme AESDM est également implémenté sur Matlab. Quelques résultats sont présentés dans la section 4.3.

#### 4.2.6. Algorithme de gradient Universel

C'est une procédure générique de descente qui permet de résoudre des problèmes d'optimisation non linéaires avec et sans contraintes de bornes. Elle comporte la plus part des méthodes de descente basées sur l'unique information du gradient (cf. section 3.5.2.1). Les règles de recherche linéaire d'Armijo, de Wolfe et de Goldstein & Price ont été implémentées afin d'offrir une dizaine de combinaisons d'algorithmes entre direction et pas de recherche. L'utilisateur disposera ainsi d'une structure d'options qui lui permet de sélectionner la stratégie de descente, la stratégie du calcul de la longueur de descente et la stratégie de calcul du gradient. Nous reviendrons à cette troisième option dans le chapitre suivant.

Les contraintes de bornes sont prises en compte suivant la méthode de reparamétrisation développée dans la section 4.1.3.2. L’interface utilisateur de l’algorithme AGU permet la prise en compte de ces contraintes sans aucun calcul supplémentaire. Si ces contraintes existent, L’utilisateur n’aura qu’à les spécifier dans le champ d’options correspondant.

Cet algorithme est développé à base d’une structure modulaire, il offre ainsi la possibilité d’étendre le panel des techniques à base de gradient pour traiter des problèmes spécifiques telles que les problèmes d’optimisation mal conditionnés ou non différentiables.

Dans ce qui suit, nous ne développons qu’une combinaison particulière de cet algorithme. Elle compose l’algorithme AESD avec une stratégie de quasi-Newton de type BFGS (cf. section 3.5.2.2.1). L’objectif de cette méthode est de mieux adapter la descente de l’algorithme lors des zones différentiables et ceci en considérant la courbure de la fonction critère. Pour ce faire, on initialise l’algorithme par une stratégie BFGS qui permet de générer à chaque itération une matrice  $B^{(k)}$  qui tend vers l’inverse du hessien, sans jamais avoir à inverser de matrice et en utilisant que le gradient. Initialement, on choisit  $B^{(0)} = I_n$  afin de débiter l’algorithme comme dans une méthode de gradient. Ceci permet une bonne exploration locale et une convergence rapide vers un bassin d’attraction. Les mises à jour de la matrice  $B^{(k)}$  permettent d’accélérer la convergence à l’intérieur du bassin d’attraction et arrivent très vite à détecter une zone précise de convergence. Même si la stratégie BFGS est puissante près d’un optimum, elle souffre elle aussi, comme la méthode du gradient, d’une inefficacité près des régions non différentiables du critère. On utilise alors le même type de commutation que dans 4.2.5 afin de parfaire la convergence avec l’algorithme spécialisé AESD.

Conséquemment, cette méthode fusionne trois stratégies de descente dans un ordre qui permet à chacune d’elles de conserver son plus grand avantage, à savoir :

- La robustesse et la rapidité de la méthode de gradient lors d’une phase d’initialisation (loin de l’optimum).
- La convergence rapide de la méthode BFGS dès qu’un bassin d’attraction est atteint.
- La robustesse vis-à-vis des critères non différentiables et mal conditionnés de l’algorithme AESD.

#### 4.2.6.1. Notations

La recherche linéaire choisie suit la règle Wolfe faible (cf. section 3.5.2.2.3), l’algorithme proposé utilise les paramètres supplémentaires suivants :

$\underline{\alpha}$  et  $\bar{\alpha}$  : Les bornes minimale et maximale sur le pas de descente  $\alpha^{(k)}$ .

$\tau_i$  et  $\tau_e$  : Les facteurs de réduction et d’expansion pour la recherche de Wolfe.

$\omega_2$  : Le facteur de la condition de courbure de Wolfe.

#### 4.2.6.2. AGU

Le schéma général de cette méthode est donné par la représentation algorithmique suivante :

**Algorithme AGU**

**Etape 0 : (Initialisation)**

Soit  $x^{(0)} \in \mathcal{S} \cap D$ ,  $\tau_i \in ]0, 1/2[$ ,  $\tau_e > 1$ ,  $0 < \omega_1 < \omega_2 < 1$ ,  $\varepsilon_0 > 0$ ,  $v_0 \geq 0$ ,  $\mu \in ]0, 1]$ ,  $\theta \in ]0, 1]$ ,  $k = 0$ ,  $m = 1$ ,  $r = 0$ ,  $Cpt = 0$ ,  $\bar{N} = 3$  et  $B^{(0)} = I_n$ .

**Etape 1 : (Approximation du  $\varepsilon$ -sous-différentiel de Clarke  $G^{(k)}$ )**

Si  $m \neq 1$  // Test de la méthode de descente

$$x_0^{(k)} = x^{(k)},$$

Soient  $u_1^{(k)}, u_2^{(k)}, \dots, u_m^{(k)}$  des échantillons uniformes de  $B$ ,

Faire  $x_0^{(k)} = x^{(k)}$  et  $x_j^{(k)} = x^{(k)} + \varepsilon_k u_j^{(k)}$  pour  $j = 1, \dots, m$ ,

Si l’un des échantillons ( $j = 1, \dots, m$ ) vérifie  $x_j^{(k)} \notin D$  // Test de différentiabilité

Aller à l’étape 1.

Sinon

$$\text{Faire } G^{(k)} = \text{clconv}\{\nabla f(x_0^{(k)}), \nabla f(x_1^{(k)}), \dots, \nabla f(x_m^{(k)})\},$$

Fin Si

Fin Si

**Etape 2 : (Calcul de la direction de descente  $d^{(k)}$ )**

Si  $m = 1$

$$g^{(k)} = \nabla f(x^{(k)}),$$

Sinon

Soit  $g^{(k)} \in G_k$  la solution du problème quadratique positif :  $g^{(k)} = \arg(\min_{g \in G_k} \|g\|_2)$ ,

Fin Si

Si  $\|g^{(k)}\|_2 = 0$

Fin de l’algorithme.

Sinon

Si  $\|g^{(k)}\|_2 \leq v_k$

Faire  $\alpha^{(k)} = 0$  et  $v_{k+1} = \theta v_k$ ,

Si  $m = 1$

Faire  $\varepsilon_{k+1} = \varepsilon_k$ ,

Sinon

Faire  $\varepsilon_{k+1} = \mu \varepsilon_k$ ,

Fin Si

Aller à l’étape 4.

Sinon

Faire  $v_{k+1} = v_k$  et  $\varepsilon_{k+1} = \varepsilon_k$ ,

Si  $m = 1$

Faire  $d^{(k)} = -B^{(k)} g^{(k)}$ ,

// Direction de descente BFGS

Sinon

Faire  $d^{(k)} = -g^{(k)} / \|g^{(k)}\|_2$ ,

// Direction de plus profonde descente

Fin Si

Fin Si

Fin Si

**Etape 3 : (Calcul du pas d’optimisation  $\alpha^{(k)}$ )**

Soient  $\underline{\alpha} = 0$  et  $\bar{\alpha} = +\infty$ ,

// Règle de Wolfe (faible)

Choisir un premier pas  $\alpha > 0$ ,

Pour  $r = 1 : 1 : N_{\max}$

Si  $r = N_{\max}$  // Test du nombre maximal de réduction

Si  $m = 1$

Faire  $Cpt = Cpt + 1$ ,

Si  $Cpt = \bar{N}$  // Test de commutation entre méthodes

$m \in \{n+1, n+2, \dots\}$ ,

Fin Si

Fin Si

Faire  $\alpha^{(k)} = 0$ ,

Arrêt Pour,

Fin Si

Poser  $\underline{\alpha} = \alpha$ ,

Si  $f(x^{(k)} + \alpha^{(k)} d^{(k)}) \geq f(x^{(k)}) - \omega_1 \alpha^{(k)} \|g^{(k)}\|_2$

$\bar{\alpha} = \alpha$ ,

Choisir  $\alpha \in [(1 - \tau_i) \underline{\alpha} + \tau_i \bar{\alpha}, \tau_i \underline{\alpha} + (1 - \tau_i) \bar{\alpha}]$ ,

Sinon

Si  $\nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})^T d^{(k)} \geq \omega_2 \nabla f(x^{(k)})^T d^{(k)}$

Arrêt Pour,

Sinon

$\underline{\alpha} = \alpha$ ,

Si  $\bar{\alpha} = +\infty$

Choisir  $\alpha \in [\tau_i \underline{\alpha}, +\infty[$ ,

Sinon

Choisir  $\alpha \in [(1 - \tau_i) \underline{\alpha} + \tau_i \bar{\alpha}, \tau_i \underline{\alpha} + (1 - \tau_i) \bar{\alpha}]$ ,

Fin Si

Fin Si

Fin Si

Fin Pour

**Etape 4 : (Actualisation)**

Si  $x^{(k)} + \alpha^{(k)} d^{(k)} \in D$  // Test de différentiabilité

Faire  $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$  et  $k = k + 1$ ,

Si  $m = 1$  // Actualisation BFGS

Faire  $s = x^{(k+1)} - x^{(k)}$  et  $y = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$ ,

Faire  $B^{(k+1)} = B^{(k)} + \frac{s \cdot s^T}{s^T \cdot y} - \frac{B^{(k)} y \cdot y^T B^{(k)}}{y^T B^{(k)} y} + \frac{1}{y^T B^{(k)} y} \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right) \left( \frac{y^T B^{(k)} y}{s^T y} s - B^{(k)} y \right)^T$ ,

Fin Si

Aller à l'étape 1.

Sinon

Choisir  $\hat{x}^{(k)} \in \hat{x}^{(k)} + \varepsilon_k B$  tel que :  $\hat{x}^{(k)} + \alpha^{(k)} d^{(k)} \in D$  et  $f(\hat{x}^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)}) - \omega_1 \alpha^{(k)} \|g^{(k)}\|_2$ ,

Faire  $x^{(k+1)} = \hat{x}^{(k)} + \alpha_k d^{(k)}$  et  $k = k + 1$ ,

Si  $m = 1$

Faire  $s = x^{(k+1)} - x^{(k)}$  et  $y = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$ ,

$$\text{Faire } B^{(k+1)} = B^{(k)} + \frac{s \cdot s^T}{s^T \cdot y} - \frac{B^{(k)} y \cdot y^T B^{(k)}}{y^T B^{(k)} y} + \frac{1}{y^T B^{(k)} y} \left( \frac{y^T B^{(k)} y}{s^T \cdot y} s - B^{(k)} y \right) \left( \frac{y^T B^{(k)} y}{s^T y} s - B^{(k)} y \right)^T,$$

Fin Si

Aller à l’étape 1.

Fin Si

Dans cette méthode, les valeurs, par défaut, des nouveaux paramètres sont :  $\tau_i = 0,4$  ,  $\tau_e = 1,2$  ,  $N_{\max} = 25$  et  $\omega_2 = 0,99$  .

Les conditions de Wolfe peuvent bien être remplacées par des conditions de Goldstein et Price afin de réduire encore le nombre de gradients calculés. Dans ce cas, la structure de l’étape 3 ne change pas et c’est juste la dérivée  $\nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})^T$  , dans la condition de courbure, qui est estimée selon l’équation (3-40).

L’algorithme AGU que nous avons développé présente la même structure de descente que les deux algorithmes précédents, sa différence majeure réside dans le fait qu’il possède deux stratégies de descente : Newtonienne (BFGS) et de Cauchy (plus profonde descente). Dans la section suivante ses performances numériques sont évaluées et comparées avec l’ensemble des méthodes développées dans ce chapitre.

### 4.3. Évaluation et comparaison des algorithmes développés

Cette section est consacrée à l’évaluation des algorithmes précédemment développés sur une sélection de problèmes d’optimisation non linéaires avec et sans contraintes de bornes. Les algorithmes développés n’assurent qu’une convergence locale, nous ne nous intéresserons alors qu’aux problèmes test à un seul minimum global et en particuliers aux problèmes à deux variables, qui permettent une analyse graphique simple et rapide des résultats. Les problèmes d’optimisation de plus grande dimension sont laissés au chapitre 6 où ils seront discutés et illustrés via des problèmes d’analyse et de validation de cahiers des charges.

Nous précisons, que l’étude qui sera présentée n’a pas pour objectif de mettre en évidence des avantages définitivement acquis d’un algorithme par rapport à un autre. Elle ne vise pas non plus à dresser un bilan exhaustif pour les algorithmes. Basée sur un nombre réduit de problèmes, elle ne peut évidemment pas prétendre à justifier une éventuelle décision. Le but est de pouvoir différencier les capacités de deux classes d’algorithmes et de mettre en évidence l’effet des modifications apportées sur l’une de leurs méthodes.

#### 4.3.1. Les problèmes test

Souvent utilisés comme des problèmes test dans la littérature, les problèmes d’optimisation choisis présentent des complexités liées essentiellement au conditionnement et/ou à la non différentiabilité du critère. Dans le tableau 4.1, nous exposons l’ensemble de problèmes test choisis. Pour chacun de ces problèmes, nous définissons le critère à optimiser, les paramètres initiaux et les contraintes de bornes.

Les représentations graphiques de la fonction critère et ses courbes de niveau sont aussi données, elles permettent de percevoir la complexité de chaque problème test.

Selon la prise en compte ou non des contraintes d’encadrement, deux séries de tests seront présentées :

### 4.3.2. Résolution des problèmes test sans contraintes d’encadrement

Dans un premier temps, chaque problème d’optimisation est résolu sans la prise en compte des contraintes de bornes. Pour ce faire, deux classe d’algorithmes sont testées et comparées. La première comporte cinq méthodes de recherche directe de type simplexe :

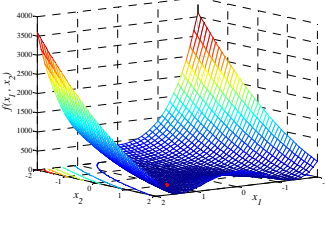
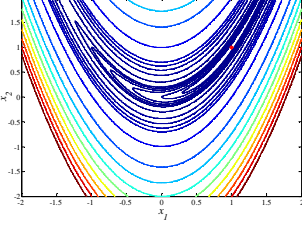
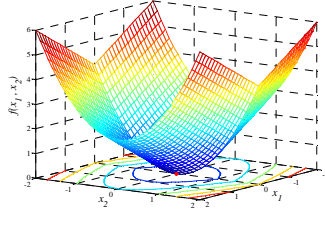
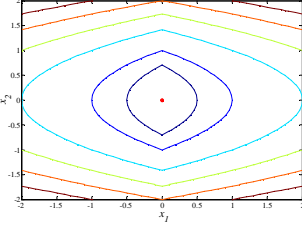
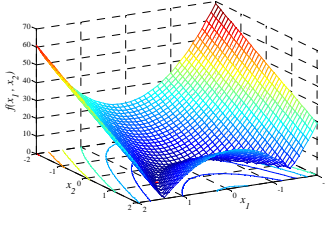
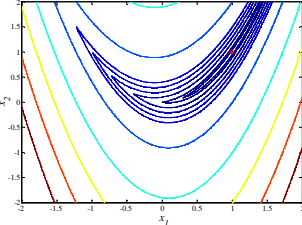
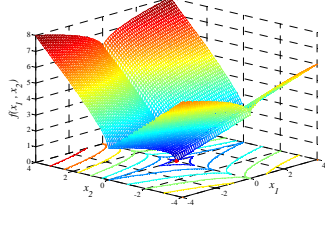
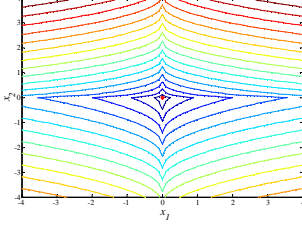
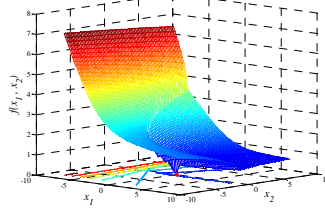
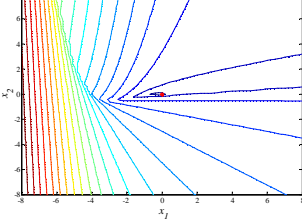
- Algorithme 1 : Algorithme de Nelder Mead original [Nel65].
- Algorithme 2 : Algorithme 1 + traitement des dégénérescences.
- Algorithme 3 : Algorithme 2 + modification du simplexe initial + changement du critère d’arrêt (ASM sans contraintes).
- Algorithme 4 : Algorithme du simplexe implémenté sous la fonction “fminsearch” de la boîte à outils Optimisation de Matlab [Lag98].
- Algorithme 5 : Algorithme de Torçzon [Tor89b, Tor91] implémenté sous la fonction “mdsmax” de la boîte à outil Matrice de Matlab [Hig93, Hig02a, Hig02b].

Les améliorations apportées à l’algorithme de base, dans les algorithmes 2 et 3, visent à améliorer ses performances et surtout à corriger son principal défaut qui est la dégénérescence (cf. section 4.1.2). Ces algorithmes sont ensuite comparés avec deux autres algorithmes bien adaptés aux problèmes de dégénérescence tirés des boîtes à outils de Matlab.

Le deuxième groupe d’algorithmes comporte cinq méthodes de descente à base de la généralisation du gradient :

- Algorithme 6 : Algorithme de plus profonde descente (du gradient) avec une recherche linéaire à base de la règle d’Armijo.
- Algorithme 7 : Algorithme de quasi-Newton avec une mise à jour BFGS et une recherche linéaire à base d’une interpolation mixte quadratique et cubique, implémenté sous la fonction “fminunc” de la boîte à outil Optimisation de Matlab.
- Algorithme 8 : Algorithme du  $\varepsilon$ -sous-différentiel (AESD).
- Algorithme 9 : Algorithme du  $\varepsilon$ -sous-différentiel modifié (AESDM).
- Algorithme 10 : Algorithme du gradient universel (AGU).

*Note : nous avons indiqué en [ ] les algorithmes développés.*

(P1) $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ (Fonction de Rosenbrock)		
Fonction quadratique convexe mal paramétrée	 	
$(x_1^{(0)}, x_2^{(0)}) = (2, -3)$		
$(x_1, x_2) \in ]-\infty, 1/2[ \times ]-\infty, 1/2]$		
(P2) $f(x_1, x_2) =  x_1  + x_2^2$ (Fonction de norme mixte)		
Fonction non différentiable	 	
$(x_1^{(0)}, x_2^{(0)}) = (1, -6)$		
$(x_1, x_2) \in [2, 4] \times [-1, 4]$		
(P3) $f(x_1, x_2) = (1 - x_1)^2 + 10 x_2 - x_1 $ (Fonction de Rosenbrock non lisse)		
Fonction non différentiable mal paramétrée	 	
$(x_1^{(0)}, x_2^{(0)}) = (-5, 4)$		
$(x_1, x_2) \in [0, 1] \times \mathfrak{R}$		
(P4) $f(x_1, x_2) = \sqrt{ x_1 } + \text{Re}(\sqrt{x_2}) +  x_2 $		
Fonction non différentiable non Lipschitz	 	
$(x_1^{(0)}, x_2^{(0)}) = (6, -4)$		
$(x_1, x_2) \in [-2, +\infty[ \times [-4, -2]$		
(P5) $f(x_1, x_2) = \max_i(\text{Re}(\lambda_i))$ tel que $\lambda_i^4 + x_1(\lambda_i^3 - \lambda_i^2) + x_2(\lambda_i + 1) = 0$		
Fonction non différentiable non Lipschitz	 	
$(x_1^{(0)}, x_2^{(0)}) = (-7, 5)$		
$(x_1, x_2) \in [1, +\infty[ \times \mathfrak{R}$		

Tab. 4.1: Les problèmes d’optimisation test

#### 4.3.2.1. Paramètres des algorithmes

Pour pouvoir comparer l’efficacité des différents algorithmes d’optimisation, deux critères assurant l’arrêt en un temps raisonnable de l’algorithme sont choisis ; le nombre maximal d’itérations  $C_{\max}$  et le nombre maximal d’évaluations de la fonction critère  $N_{\max}$ . Dans tous les algorithmes, ces deux paramètres sont fixés à 1000 et 10000 respectivement.



Algorithme 1 : La taille du simplexe régulier  $a = 0,05$  (cf. équation (4-2)),  
La tolérance sur la fonction critère  $\varepsilon_f = 10^{-9}$  (cf. équation (4-3)).

Algorithme 2 : La tolérance de dégénérescence  $\varepsilon_{\text{deg1}} = 10^{-6}$  (cf. équation (4-4)).

Algorithme 3 :  $\varepsilon_x = 10^{-4}$ ,  $\varepsilon_f = 10^{-4}$  (cf. équation (4-15)).

Contrairement à l’initialisation proposée dans (4-1) et (4-2), les algorithmes à base du simplexe peuvent aussi être initialisés par des simplexes irréguliers. Dans cette catégorie, on cite les simplexes à angle droit, les simplexes proportionnés et les simplexes aléatoires. Le choix de l’un de ces initialisations peut influencer grandement l’évolution de l’algorithme [Lag98]. Dans l’algorithme 3, on propose de tester l’initialisation suggérée par L. Pfeffer [Lag98]. Cette dernière consiste à perturber les éléments non nuls du point initial de 5% et de remplacer les éléments nuls par 0.00025. Ce choix s’est effectué suite à une série d’essais qui montrent son efficacité.

Algorithme 4 :  $\varepsilon_x = 10^{-4}$ ,  $\varepsilon_f = 10^{-4}$  (voir la fonction “fminserach” de la boîte à outils Optimisation de Matlab).

Algorithme 5 : La taille du simplexe initial  $\varepsilon = 10^{-4}$  (voir la fonction “mdsmax” de la boîte à outils Matrice de Matlab).

Algorithme 6 : Le critère d’arrêt est  $\|\nabla f(x)\|_2 < \varepsilon$  avec  $\varepsilon = 10^{-6}$ .

Algorithme 7 :  $\varepsilon_x = 10^{-6}$ ,  $\varepsilon_f = 10^{-6}$  (voir la fonction “fminunc” de la boîte à outils Optimisation de Matlab).

Algorithme 8 : voir la section 4.2.4.

Algorithme 9 : voir la section 4.2.5.

Algorithme 10 : voir la section 4.2.6.

L’ensemble des tests a été effectué sous Matlab 6.5 en utilisant un format de calcul double précision. Cependant, les résultats qui seront présentés, ci-dessous, sont donnés en un format virgule fixe à cinq chiffres.

### 4.3.2.2. Analyse des résultats numériques

#### 4.3.2.2.1. Le problème (P1)

Le problème (P1) traite la fonction de Rosenbrock<sup>1</sup>, appelée aussi fonction de Chebyquad. Cette fonction est assez difficile à optimiser : le premier terme de la fonction définit une quadratique dont les bords sont très abrupts, le fond est plat et défini par la parabole  $x_2 = x_1^2$ . C’est sur cette courbe qu’il faut ensuite optimiser le second terme, sachant que toute petite perturbation de  $x_1$  entraîne immédiatement une forte variation du premier terme si  $x_2$  ne varie pas simultanément pour que le point  $(x_1, x_2)$  reste sur la parabole  $x_2 = x_1^2$ . Une étude élémentaire montre qu’il existe un seul et unique minimum qui est le point  $x^* = (1,1)$  et que la valeur du minimum en ce point est de  $f^* = 0$ .

<sup>1</sup> Dans certains ouvrages [Ger00], la fonction de Rosenbrock est définie avec un coefficient 105 au lieu de 100. Cette différence n’affectera pas les résultats obtenus et n’aura aucun effet sur les conclusions faites par la suite.

En utilisant le même point initial (cf. tableau 4.1), les résultats des différents algorithmes d’optimisation testés pour le problème (P1), sont résumés dans le tableau récapitulatif suivant.

Algorithme	Méthodes de recherche directe				
	1	2	3	4	5
Critère $\bar{f}$	1.1024e-08	1.1024e-08	3.7661e-09	4.0366e-09	0.0068
Solution $\bar{x}$	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)	(0.9173, 0.8414)
Nb itérations	64	64	56	60	112
Nb éval.	125	125	108	114	3151
$\ \nabla f(\bar{x})\ _2$	4.6295e-04	4.6295e-04	5.6330e-06	5.0241e-04	0.0948
Temps (s)	0.0710	0.0800	0.0410	0.0200	0.6910
Algorithme	Méthodes de descente à base de gradient				
	6	7	8	9	10
Critère $\bar{f}$	0.0114	3.4695e-09	2.9925e-12	2.7642e-10	0
Solution $\bar{x}$	(0.9029, 0.8107)	(1.0000, 0.9999)	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)
Nb itérations	1000	13	49	48	16
Nb éval.	9682	56	577	368	247
Nb éval. $\nabla f(x)$	1000	13	196	108	40
$\ \nabla f(\bar{x})\ _2$	1.6628	0.0020	2.3382e-06	9.0676e-07	9.0676e-07
Temps (s)	42.5720	0.0610	3.6260	1.3530	0.0920

**Tab. 4.2: Résultats du problème (P1) sans contraintes de bornes**

Mise à part l’algorithme de Torçzon (algorithme 5) qui se bloque en un point sur la courbe  $x_2 = x_1^2$ , les autres méthodes de recherche directe atteignent assez rapidement et efficacement l’optimum global  $x^* = (1,1)$ . Dans cette catégorie, l’algorithme ASM (algorithme 3) présente des résultats meilleurs, en termes de nombre d’évaluations et d’itérations, par rapport à ceux obtenus par la fonction “fminserach” de Matlab. Cette dernière présente, néanmoins, un temps de calcul deux fois plus rapide.

Le constat est le même pour les méthodes de descente, où la méthode du gradient (algorithme 6), mal adaptée à ce type de problème, s’arrête au bout de 1000 itérations sur la vallée définie par  $x_2 = x_1^2$  sans atteindre le minimum global. Plus on se rapproche de l’optimum et plus la convergence de cet algorithme devient lente. Ceci est dû à la taille du pas d’optimisation qui devient rapidement trop petite à cause des oscillations autour de la courbe  $x_2 = x_1^2$ .

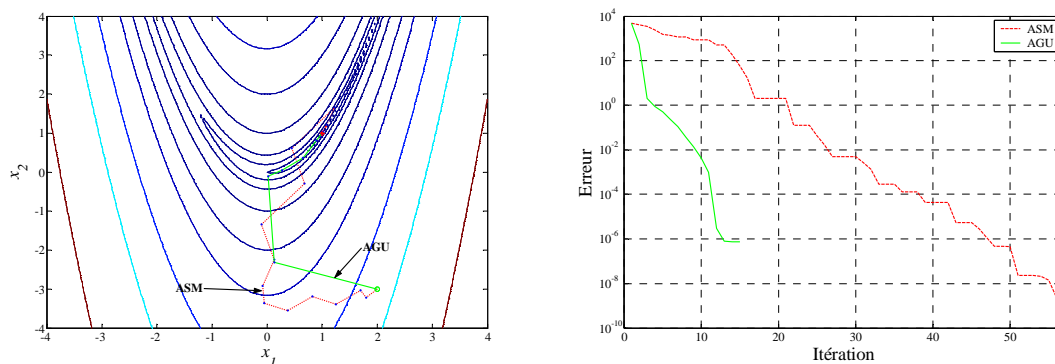
Par ailleurs, la fonction “fminunc” (algorithme 7) ne met que 13 itérations pour converger avec une tolérance  $\epsilon_x$  fixée à  $10^{-6}$  et détient ainsi le meilleur nombre d’itérations et avec seulement 56 évaluations de la fonction objectif. L’efficacité de cette méthode vient de la mise à jour BFGS qui permet de se ramener rapidement à un algorithme de type Newton avec une bonne approximation du hessien du critère quadratique.

Malgré l’amélioration apportée à l’algorithme AESD, dans sa version modifiée AESDM, le nombre d’itérations et d’évaluations de la fonction objectif reste élevé. Ceci est dû à l’inefficacité de la phase initiale de cet algorithme (phase du gradient) qui atteint rapidement ses limites près de la courbe  $x_2 = x_1^2$  et qui fait basculer la méthode vers une stratégie à base de  $\epsilon$ -sous-gradient (gradient

échantillonné). Cette dernière est gourmande en termes d’évaluation de la fonction critère et de temps de calcul.

En revanche, en échangeant la phase initiale du gradient dans l’algorithme AESDM par une mise à jour BFGS, le nombre d’itérations se réduit considérablement (3 fois en moins). La stratégie BFGS montre toute son efficacité vis-à-vis du problème quadratique (P1). De plus, si la mise à jour BFGS est initialisée par une bonne estimée du hessien (au lieu de la matrice identité), le résultat serait nettement meilleur puisque le problème (P1) est quadratique.

La figure ci-dessous décrit les courbes de trajectoire et de décroissance des algorithmes ASM et AGU.



**Fig. 4.18** Comparaison des performances des algorithmes ASM et AGU pour le problème (P1)

Nous constatons que ces algorithmes permettent de résoudre assez efficacement le problème (P1) et ils peuvent être jugés bien adaptés aux problèmes quadratiques mal paramétrés. Toutefois, en utilisant des directions de descente à base de gradient, l’algorithme AGU paraît, tout de même, plus pertinent. Il présente une trajectoire beaucoup plus courte avec une décroissance plus rapide.

L’efficacité, en termes de temps de calcul, des méthodes de recherche directe par rapport aux méthodes de descente à base de gradient n’est pas indépendante du type de problèmes traités. En effet, pour des problèmes de dimension plus grande, les méthodes de recherche directe deviennent très rapidement inopérantes [Bon97]. Ce point sera nettement constaté dans les exemples du chapitre 6.

#### 4.3.2.2.2. Le problème (P2)

Il s’agit de tester l’efficacité des algorithmes développés pour une nouvelle complexité qui est la non différentiabilité du critère. Pour ce faire, le problème (P2) est utilisé. Il consiste à minimiser un critère de norme mixte rassemblant un terme quadratique et un terme de module. La même série de tests donne les résultats suivants :

Pour le problème (P2), le minimum global est atteint pour  $x^* = (0, 0)$ . Nous remarquons, tout d’abord, que les algorithmes 5 et 6 se bloquent sur la droite  $x_1 = 0$  où ils oscillent stérilement autour de cette zone non différentiable.

Dans l’ensemble, le reste des algorithmes convergent vers le minimum global avec une précision et une rapidité diverses. Ceci est principalement dû aux caractéristiques du critère traité qui est différentiable presque partout et à dominance quadratique.

Algorithme	Méthodes de recherche directe				
	1	2	3	4	5
Critère $\bar{f}$	1.0973e-09	1.0973e-09	5.9462e-07	5.3881e-08	1.7099e-02
Solution $\bar{x}$	(1.1e-9, 4.2e-6)	(1.1e-9, 4.2e-6)	(-9.9e-6, -6.2e-6)	(5.3e-8, 3.7e-6)	(-7.1e-6, 1.3e-1)
Nb itérations	98	98	56	60	14
Nb éval.	181	181	100	111	87
$\ \nabla f(\bar{x})\ _2$	1.0000	1.0000	1.0000	1.0000	1.0336
Temps (s)	0.0600	0.0600	0.0500	0.0700	0.1300
Algorithme	Méthodes de descente à base de gradient				
	6	7	8	9	10
Critère $\bar{f}$	2.8632e-03	4.2742e-07	7.1833e-08	3.1000e-08	1.0001e-09
Solution $\bar{x}$	(0, -5.3e-2)	(4.1e-7, -1.4e-4)	(7.2e-7, 6.8e-7)	(3.1e-8, -9.7e-8)	(1.0e-9, -1e-11)
Nb itérations	1000	8	47	40	18
Nb éval.	8355	66	693	438	106
Nb éval. $\nabla f(x)$	1000	8	188	70	48
$\ \nabla f(\bar{x})\ _2$	1.0057	1.0000	1.0000	1.0000	1.0000
Temps (s)	33.3550	0.1400	3.8060	0.8420	0.1120

Tab. 4.3: Résultats du problème (P2) sans contraintes de bornes

Les tests numériques effectués, jusqu’à ici, montrent aussi que l’algorithme ASM présente des performances légèrement supérieures à celles de la fonction “fminserach” de la boîte à outils Optimisation de Matlab. Cette différence est principalement due à l’initialisation du simplexe choisie.

Les algorithmes développés dans le cadre de notre étude (colonnes jaunes sur les tableaux) parviennent à joindre le minimum rapidement et les algorithmes ASM et AGU restent les plus performants.

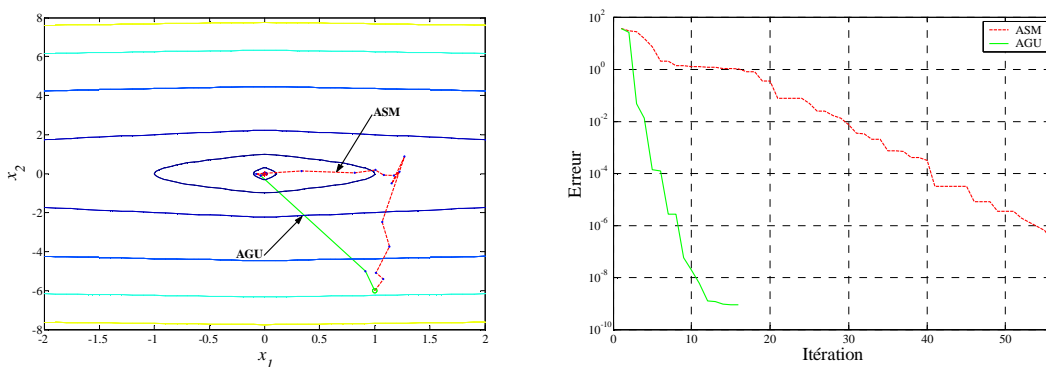


Fig. 4.19 Comparaison des performances des algorithmes ASM et AGU pour le problème (P2)

Comme pour le cas précédent, les résultats numériques prouvent que la stratégie AGU est plus intelligente et elle permet de converger vers l’optimum en peu d’itérations et avec plus de précision. Cependant, le temps de calcul enregistré pour la méthode AGU est deux fois supérieur à celui de la méthode ASM. Ceci est surtout dû à l’estimation de la direction et du pas de descente qui sont issus de l’estimation du gradient ou de sa généralisation ( $\varepsilon$ -sous-différentiel).

4.3.2.2.3. Le problème (P3)

Le troisième problème (P3), dit aussi fonction de Rosenbrock non lisse, traite une variante du premier problème (P1) où le deuxième terme en puissance carrée, dans la fonction critère, est remplacé par une valeur absolue. Le problème (P3) devient, donc, à la fois mal paramétré et mixte (un terme quadratique et un terme non différentiable).

Les résultats obtenus, en effectuant la même série de tests, sont résumés par le tableau 4.4.

Les résultats des méthodes de recherche directe montrent que les algorithmes testés réussissent à converger rapidement à l’exception de l’algorithme de Torçzon (algorithme 5) qui se bloque, comme précédemment, au bout de 11 itérations. L’algorithme “fminserach”, sous Matlab, qui converge assez rapidement au début, stagne après 278 itérations sans atteindre l’optimum global  $x^* = (1,1)$ . Ces deux algorithmes se heurtent à la difficulté de glisser le long d’une zone non différentiable, d’une fonction mal paramétrée, définie par  $x_2 = x_1^2$ .

Les algorithmes 1, 2 et 3 ne présentent pas de différences significatives mis à part une amélioration de la précision qui passe de  $3,4778.10^{-6}$  à  $5,4001.10^{-7}$ .

Dans la deuxième classe d’algorithmes testés, l’inefficacité des méthodes du gradient et “fminunc” est confirmée. Ces dernières ne convergent pas parce qu’elles sont mal adaptées aux problèmes non différentiables. L’algorithme du gradient s’arrête après avoir atteint le nombre maximal d’itérations autorisées, tandis que la fonction “fminunc” se bloque après seulement 33 itérations sur la “faille”  $x_2 = x_1^2$ . Quant aux algorithmes 8, 9 et 10, ils assurent une convergence de plus en plus rapide. La version AGU permet même de résoudre le problème en 0,25 seconde avec une précision de  $10^{-16}$  ce qui est de loin meilleur que la performance de l’algorithme ASM sur ce problème (cf. figure 4.20).

Algorithme	Méthodes de recherche directe				
	1	2	3	4	5
Critère $\bar{f}$	3.4778e-06	1.4203e-06	5.4001e-07	1.1285e-04	0.1040
Solution $\bar{x}$	(0.9998, 0.9997)	(0.9999, 0.9999)	(1.0000, 1.0000)	(0.9894, 0.9789)	(0.6778, 0.4594)
Nb itérations	473	488	478	278	11
Nb éval.	851	874	898	490	95
$\ \nabla f(\bar{x})\ _2$	22.3575	22.3589	22.3606	22.1906	16.3304
Temps (s)	0.2400	0.2500	0.2720	0.1300	0.0700
Algorithme	Méthodes de descente à base de gradient				
	6	7	8	9	10
Critère $\bar{f}$	0.7452	0.4894	2.5261e-10	5.3566e-09	0
Solution $\bar{x}$	(0.1368, 0.0187)	(0.3330, 0.1109)	(1.0000, 1.0000)	(1.0000, 0.9999)	(1.0000, 1.0000)
Nb itérations	1000	33	105	90	24
Nb éval.	9804	222	1764	1144	427
Nb éval. $\nabla f(x)$	1000	33	420	150	60
$\ \nabla f(\bar{x})\ _2$	10.9503	11.3298	22.3608	22.3611	22.3607
Temps (s)	24.0160	0.2000	7.9120	3.0400	0.2540

**Tab. 4.4: Résultats du problème (P3) sans contraintes de bornes**

De plus, la pertinence de l’information de descente issue du calcul des  $\varepsilon$ -sous-différentiel est montrée sur la courbe de décroissance qui présente une pente beaucoup plus raide que celle résultante de l’algorithme ASM.

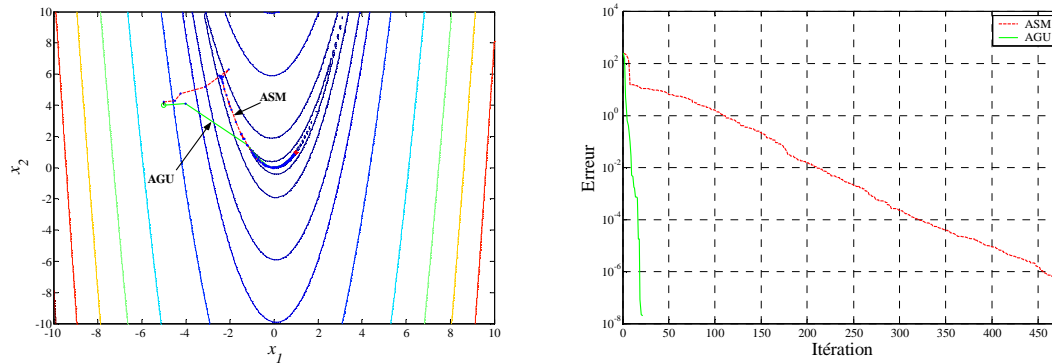


Fig. 4.20 Comparaison des performances des algorithmes ASM et AGU pour le problème (P3)

4.3.2.2.4. Le problème (P4)

Il s’agit d’un problème d’optimisation où le but est de minimiser une fonction dont le minimum global est atteint non seulement en un point non différentiable comme le cas précédent, mais aussi en un point non Lipschitz. La fonction traitée présente un seul minimum donné par  $x^* = (0, 0)$  et sa dérivée en ce point est infinie.

Les résultats numériques obtenus sont donnés par le tableau ci-dessous.

Algorithme	Méthodes de recherche directe				
	1	2	3	4	5
Critère $\bar{f}$	5.2510e-09	5.2510e-09	2.0039e-12	4.0678e-04	4.5748
Solution $\bar{x}$	(0, -1.1e-9)	(0, -1.1e-9)	(0, -2e-12)	(-6.8e-8, -1.5e-4)	(20.93, -1.8e-4)
Nb itérations	217	217	202	110	8
Nb éval.	409	409	389	207	79
$\ \nabla f(\bar{x})\ _2$	1.2013e+8	1.2013e+8	1.0000	1.9125e+3	1.0060
Temps (s)	0.1250	0.1250	0.1400	0.1050	0.0600
Algorithme	Méthodes de descente à base de gradient				
	6	7	8	9	10
Critère $\bar{f}$	3.1540	2.20046	3.1903e-04	3.3184e-05	5.8312e-05
Solution $\bar{x}$	(5.1606, 0)	(4.84, -1.0e-10)	(-1.1e-7, -1.7e-7)	(-1.1e-9, -1.8e-8)	(-3.4e-9, -2.4e-9)
Nb itérations	1000	1000	43	27	17
Nb éval.	8443	1013	564	284	162
Nb éval. $\nabla f(x)$	1000	1000	172	63	56
$\ \nabla f(\bar{x})\ _2$	1.0240	1.0255	1.5681e+3	1.5075e+4	8.5749e+3
Temps (s)	18.5400	0.3500	3.0250	1.1050	0.1020

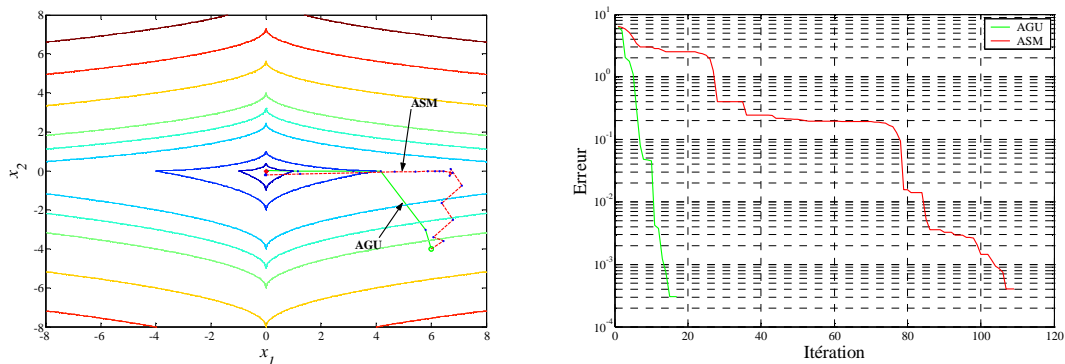
Tab. 4.5: Résultats du problème (P4) sans contraintes de bornes

Hormis le mauvais résultat de l’algorithme de Torçzon, les solutions obtenues avec les autres algorithmes de recherche directe montrent qu’elles sont efficaces et robustes même avec des critères non Lipschitz. Ces derniers parviennent à surmonter la complexité du problème en s’alignant sur la droite non différentiable  $x_2 = 0$  et en glissant tout au long afin de minimiser le critère par rapport à la première composante  $x_1$ . Cette deuxième phase est efficacement effectuée dans les algorithmes 1, 2 et 3 où l’optimum (l’origine) est parfaitement atteint.

Similairement à cette approche, l’algorithme du gradient, comme celui de Torçzon, exploite la position du point initial  $x_0 = (6, -4)$  pour avancer du côté de la non différentiabilité la plus simple c’est-à-dire, celle de la droite  $x_2 = 0$ . En effet, si  $x_2 \leq 0$ , la fonction critère devient  $f(x_1, x_2) = \sqrt{|x_1|} - x_2$  et il est, par conséquence, plus simple d’annuler la composante  $x_2$  qui présente une simple variation constante comparant à celle du premier terme. Malheureusement, ces deux algorithmes n’achèvent pas ce processus et se bloquent près de la droite  $x_2 = 0$ .

La méthode quasi-Newton avec la mise à jour BFGS ne parvient pas, elle aussi, à assurer la convergence vers le minimum. Semblable à un algorithme de gradient dans ses premières itérations, son caractère quadratique ne lui permet d’approcher le minimum efficacement. Ceci est principalement dû à l’inefficacité de la formule BFGS pour l’estimation du hessien quand il s’agit de points non différentiables.

Les algorithmes 7, 8 et 9 à base de  $\varepsilon$ -sous-différentiel ne trouvent aucune difficulté à résoudre ce problème. La version AGU réussit même à atteindre des temps de calcul de l’ordre de  $10^{-3}$ s comme dans le cas de l’algorithme ASM. La figure ci-dessous compare l’évolution de ces deux algorithmes.



**Fig. 4.21** Comparaison des performances des algorithmes ASM et AGU pour le problème (P4)

4.3.2.2.5. Le problème (P5)

Le problème (P5) a déjà été introduit dans le chapitre 2. Il consiste à minimiser l’abscisse spectrale des racines d’un polynôme dépendant de deux paramètres  $x_1$  et  $x_2$ . Ce critère est non différentiable et non Lipschitz et a pour seul minimum l’origine.

Les tests numériques obtenus (cf. tableau 4.6) montrent que seuls les algorithmes développés au cours de notre étude (colonnes en jaune du tableau) réussissent à converger vers le minimum global  $x^* = (0, 0)$ .

Algorithme	Méthodes de recherche directe				
	1	2	3	4	5
Critère $\bar{f}$	0.6167	7.0167e-09	4.7768e-9	0.5467	0.5423
Solution $\bar{x}$	(386.06, 34.721)	(-1.3e-8, 0)	(-3.7e-4, 0)	(5.4574, 0.4203)	(5.0556, 0.3853)
Nb itérations	131	313	224	1000	8
Nb éval.	256	589	431	1805	83
$\ \nabla f(\bar{x})\ _2$	1.3674e-3	1.0535e+8	1.0472e+8	1.1696e-1	1.2753e-1
Temps (s)	0.1910	0.3610	0.4130	0.6510	0.0900
Algorithme	Méthodes de descente à base de gradient				
	6	7	8	9	10
Critère $\bar{f}$	0.5189	0.6136	1.0512e-03	1.5015e-05	8.5184e-06
Solution $\bar{x}$	(3.5319, 0.2535)	(111.12, 9.9307)	(-1.6e-3, 2.7e-7)	(-2.6e-5, 4e-11)	(-1.7e-5, 2e-13)
Nb itérations	1000	227	55	33	19
Nb éval.	9359	267	598	332	214
Nb éval. $\nabla f(x)$	1000	227	220	78	61
$\ \nabla f(\bar{x})\ _2$	0.6178	4.7982e-3	1.1248e+3	5.7017e+4	6.2618e+4
Temps (s)	13.7760	0.3410	3.2260	1.0370	0.2910

Tab. 4.6: Résultats du problème (P5) sans contraintes de bornes

La figure ci-dessous décrit l’évolution de l’algorithme AGU et celle de la fonction “fminserach”.

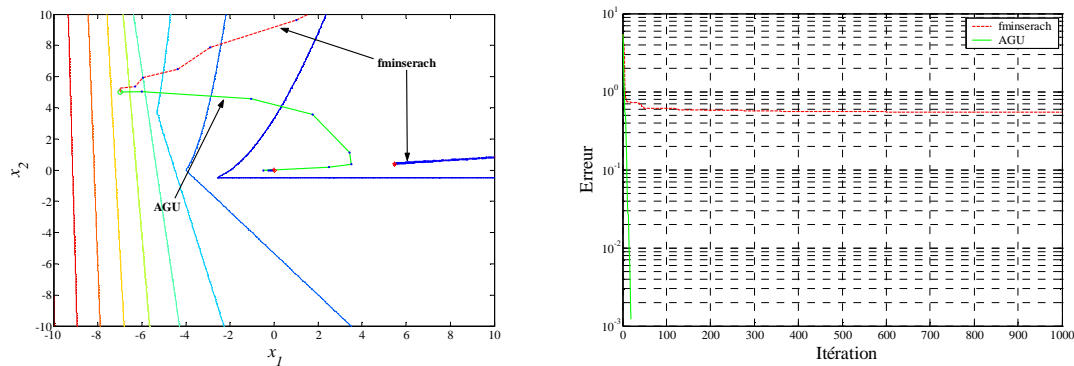


Fig. 4.22 Comparaison des performances des algorithmes AGU et fminserach pour (P5)

L’algorithme “fminserach” ne permet pas d’atteindre le minimum global et se bloque sur une zone non différentiable le long de la droite  $x_2 = 0$ . Le parcours tracé par ses 1000 itérations est complètement dévié. En effet, l’algorithme ne permet pas d’approcher directement la zone non différentiable mais progresse plutôt dans la direction d’une zone plate qui limite fortement son avancée (pas trop petits).

En général, la comparaison des résultats de l’algorithme AGU avec ceux des autres algorithmes montrent qu’il est de loin le plus efficace et qu’il est mieux adapté aux problèmes d’abscisse spectrale qui peuvent être rencontrés en automatique : stabilisation, stabilisation simultanée, synthèse  $H_\infty$  à ordre fixe...etc. Un exemple de ce dernier problème sera étudié dans le chapitre 6.



### 4.3.3. Résolution des problèmes test avec contraintes d’encadrement

Dans cette section, nous reprenons les mêmes problèmes tests afin de les résoudre, cette fois, sous les contraintes de bornes susmentionnées au tableau 4.1.

Nous avons choisi d’expérimenter les deux stratégies de prise en compte des contraintes détaillées dans la section 4.1.3 et ceci pour les deux algorithmes ASM et AGU.

Afin de pouvoir évaluer les résultats obtenus, nous les avons comparés avec les résultats de la fonction Matlab “Patternsearch” (algorithme de recherche par motif généralisé) de la boîte à outils Algorithme Génétique et Recherche Direct. L’ensemble des algorithmes considérés est donc le suivant :

- Algorithme 11 : Algorithme 3 (AESD) + prise en compte des bornes par projection.
- Algorithme 12 : Algorithme 3 (AESD) + prise en compte des bornes par reparamétrisation.
- Algorithme 13 : Algorithme de recherche par motif généralisé implémenté sous la fonction “patternsearch”.
- Algorithme 14 : Algorithme 10 (AGU) + prise en compte des bornes par projection.
- Algorithme 15 : Algorithme 10 (AGU) + prise en compte des bornes par reparamétrisation.

Note : nous avons indiqué en      les algorithmes développés.

#### 4.3.3.1. Paramètres des algorithmes

Les paramètres d’implémentation des algorithmes ASM et AGU sont conservés et la fonction “patternsearch ” est exécutée avec ses paramètres par défaut.

#### 4.3.3.2. Analyse des résultats numériques

##### 4.3.3.2.1. Le problème (P1)

Dan ce paragraphe, nous proposons de résoudre le problème d’optimisation (P1) sous les contraintes de bornes  $x \in ]-\infty, 1/2[ \times ]-\infty, 1/2[$  tout en conservant le même point initial  $x_0 = (2, -3)$ . Il est clair que ce point est infaisable et il sera, dans un premier temps, projeté sur la frontière de l’espace faisable pour les deux stratégies de prise en compte des contraintes de bornes (projection et reparamétrisation).

Les résultats numériques obtenus sont récapitulés par le tableau ci-dessous.

Algorithme	Méthodes de recherche directe			Méthodes de descente à base de gradient	
	11	12	13	14	15
Critère $\bar{f}$	0.2500	0.2500	0.2500	0.2500	0.2500
Solution $\bar{x}$	(0.5000, 0.2499)	(0.4999, 0.2500)	(0.5000, 0.2500)	(0.5000, 0.2500)	(0.5000, 0.2500)
Nb itérations	42	35	62	27	16
Nb éval.	2095	67	167	691	128
Nb éval. $\nabla f(x)$	0	0	0	108	40
$\ \nabla f(\bar{x})\ _2$	0.9995	1.0021	1.0001	1.0000	1.0000
Temps de calcul	1.1220	0.4410	0.7100	0.9540	0.5200

Tab. 4.7: Résultat du problème (P1) avec contraintes de bornes

Nous remarquons que les différents algorithmes testés réussissent à converger vers le minimum global faisable  $x^* = (1/2, 1/4)$ . Les meilleurs temps de calcul sont respectivement obtenus avec les algorithmes ASM et AGU et ceci avec une stratégie de prise en compte des contraintes de bornes avec reparamétrisation. Cette stratégie semble donc plus efficace car elle génère beaucoup moins de calculs.

Nous remarquons aussi que peu importe la technique de prise en compte des contraintes de bornes, le nombre d’itérations de la méthode AGU reste toujours le plus faible. Ce constat découle de l’adéquation et la pertinence de l’information de descente issue du  $\varepsilon$ -sous-différentiel même lorsqu’il s’agit de glisser sur des contraintes.

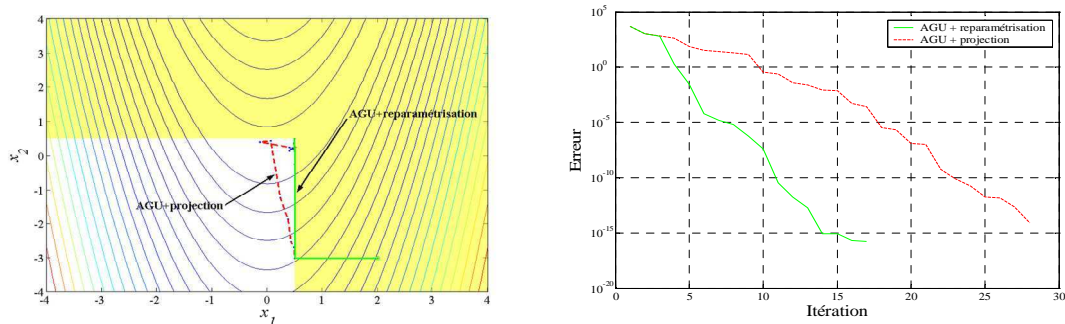


Fig. 4.23 Evolution de l’algorithme AGU pour le problème (P1) avec contraintes de bornes

La figure ci-dessus décrit la trajectoire tracée par les itérations de l’algorithme AGU avec les deux stratégies de prise en compte des contraintes. Nous pouvons distinguer rapidement la courbe correspondante à la technique de projection (en pointillés rouges) qui réussit via une réinitialisation à basculer dans l’espace faisable et de continuer son avancée avant de finir sa convergence de nouveau sur la contrainte active  $x_1 = 1/2$ .

Contrairement à cette évolution, la technique de reparamétrisation permet de rester en parallèle près de la frontière de l’espace faisable tout en progressant efficacement vers le minimum. Les courbes de décroissance témoignent de l’efficacité de la stratégie de reparamétrisation.

4.3.3.2.2. Le problème (P2)

Pour ce problème, le point d’amorce  $x_0 = (-5, 1)$  des différents algorithmes est infaisable (cf. tableau 4.1). Conformément au principe des techniques de prise en compte des contraintes d’encadrement proposées dans la section 4.1.3, ce point est initialement projeté sur la contrainte la plus proche afin de définir le nouveau point d’initialisation des algorithmes. Comme nous le remarquons sur la figure 4.24, ce nouveau point initial est  $x_0 = (2, -1)$ .

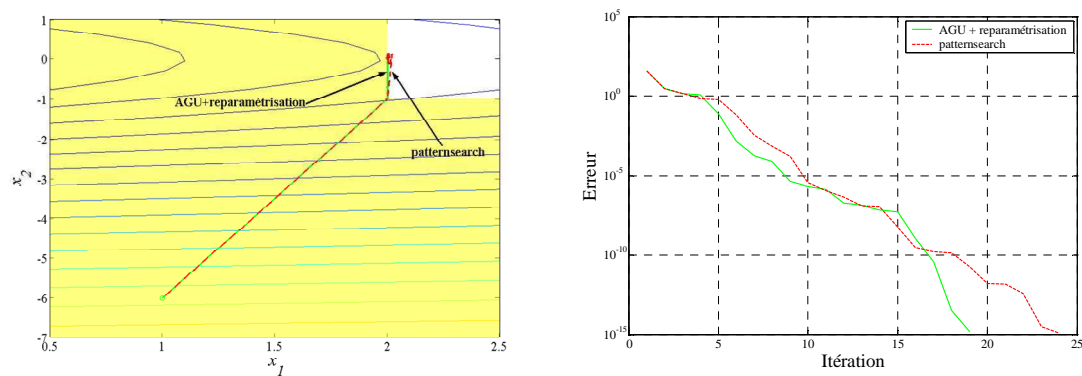
Algorithme	Méthodes de recherche directe			Méthodes de descente à base de gradient	
	11	12	13	14	15
Critère $\bar{f}$	2.0000	2.0000	2.0000	2.0000	2.0000
Solution $\bar{x}$	(2.0000, -5e-6)	(2.0000, 1.3e-6)	(2.0000, 0)	(2.0000, 6e-8)	(2.0000, 1e-9)
Nb itérations	39	30	22	33	17
Nb éval.	1142	74	65	442	79
Nb éval. $\nabla f(x)$	0	0	0	132	50
$\ \nabla f(\bar{x})\ _2$	1.0000	1.0000	1.0000	1.0000	1.0000
Temps de calcul	0.9910	0.3080	0.2790	0.7540	0.2930

Tab. 4.8: Résultat du problème (P2) avec contraintes de bornes

Les résultats d’optimisation du problème (P2) confirment la convenance de la stratégie de reparamétrisation proposée dans la section 4.1.3.2. Avec cette stratégie, nous remarquons que le nombre d’évaluations de la fonction critère est beaucoup plus petit que celui résultant de la technique de projection. Dans le cas de l’algorithme ASM, cette réduction est de 15 fois et elle est principalement due aux différentes réinitialisations opérées par cet algorithme afin d’éviter les éventuelles convergences sur des bornes actives.

Cette diminution du nombre d’évaluations de la fonction objectif est moins importante dans le cas de l’algorithme AGU (moins de 6 fois) car ce dernier est beaucoup plus performant lorsqu’il s’agit de déterminer une direction de descente dans une zone singulière (non différentiable, mal paramétrée ou sur les bornes). Le nombre de gradients nécessaires pour cette méthode est aussi moins important (50 au lieu de 132), ce qui représente un précieux gain en temps de calcul.

La technique de projection reste néanmoins opérante et permet d’atteindre le minimum avec une excellente précision.



**Fig. 4.24 Evolution de l’algorithme AGU et “patternsearch” pour le problème (P2) avec contraintes de bornes**

La figure 4.24 illustre une comparaison entre l’évolution de l’algorithme AGU avec prise en compte des contraintes par reparamétrisation et l’algorithme de recherche par motif généralisé implémenté sous Matlab. Nous apercevons qu’une fois sur les bornes, les trajectoires des deux algorithmes divergent. Les itérées de l’algorithme AGU continuent à progresser sur la frontière  $x_1 = 2$ , alors que celles de la fonction “patternsearch” se déplacent vers l’espace faisable pour continuer la progression vers le minimum  $x^* = (2, 0)$  en petit zigzag. La décroissance de la fonction “patternsearch” se trouve alors légèrement freinée alors qu’elle était même parfois meilleure que celle de l’algorithme AGU.

#### 4.3.3.2.3. Le problème (P3)

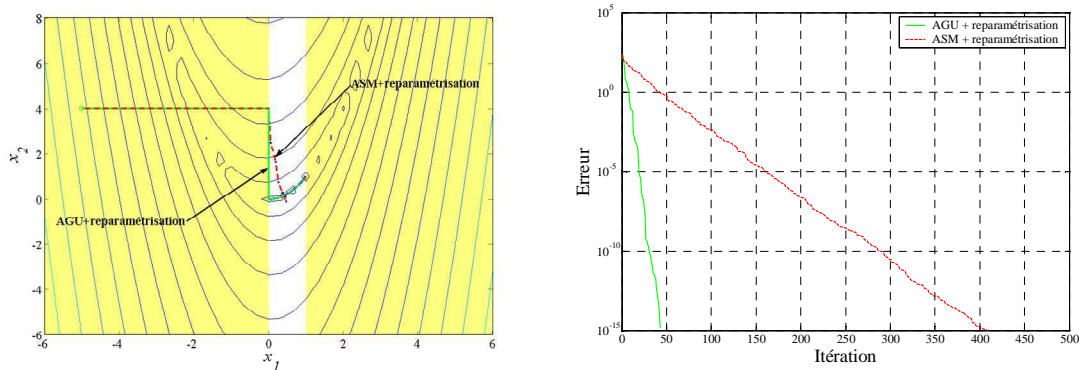
Rappelons que ce problème consiste à minimiser la variante de la fonction de Rosenbrock non lisse sous les contraintes  $x \in [0, 1] \times \mathcal{R}$  et ceci à partir du point initial infaisable  $x_0 = (-5, 4)$ .

Algorithme	Méthodes de recherche directe			Méthodes de descente à base de gradient	
	11	12	13	14	15
Critère $\bar{f}$	6.7419e-04	5.4001e-07	0.7656	3.6100e-06	5.0516e-09
Solution $\bar{x}$	(0.9746, 0.9490)	(1.0000, 0.9999)	(0.1254, 0.0160)	(0.9981, 0.9962)	(1.0000, 1.0000)
Nb itérations	714	407	50	58	41
Nb éval.	1262	727	156	692	239
Nb éval. $\nabla f(x)$	0	0	0	232	101
$\ \nabla f(\bar{x})\ _2$	21.8514	22.3406	1.7500	21.8738	22.3606
Temps de calcul	1.4520	1.0130	0.6400	1.2070	0.6980

**Tab. 4.9: Résultat du problème (P3) avec contraintes de bornes**

Les résultats numériques obtenus sont donnés par le tableau 4.9. Ils témoignent de la complexité du problème traité. En effet, contrairement à ce que nous avons constaté dans le premier problème de Rosenbrock (version lisse), l’algorithme 13 de recherche par motif généralisé stagne cette fois définitivement au bout de 50 itérations en un point près de la vallée non différentiable  $x_2 = x_1^2$ .

De même, pour le problème (P3), la technique de projection, employée dans les algorithmes 11 et 14, manque d’efficacité (nombre d’évaluation de la fonction coût) et même de précision (erreur) et elle ne permet pas d’atteindre exactement l’optimum global qui se trouve sur la contrainte active  $x_1 = 1$ . Les itérées de l’algorithme 11 et 14 s’accumulent tout au long de la zone non différentiable  $x_2 = x_1^2$  avec des pas très réduits ce qui étouffe la convergence des algorithmes.



**Fig. 4.25 Evolution de l’algorithme AGU et ASM pour le problème (P3) avec contraintes de bornes**

Cependant, les meilleurs résultats sont obtenus en utilisant la technique de reparamétrisation à base de la fonction sinus (cf. section 4.1.3.2). Cette dernière permet de conserver l’efficacité prouvée des algorithmes AGU et ASM dans le cas non contraint (cf. tableau 4.5).

Alors que l’algorithme AGU progresse davantage sur la contrainte  $x_1 = 0$ , la trajectoire suivie par l’algorithme ASM (cf. figure 4.25) se distingue par son écartement rapide de cette frontière sur laquelle l’itérée initiale est projetée. Même si le chemin parcouru par l’algorithme AGU est plus long, sa convergence rapide (en peu d’itérations) lors de sa phase finale (composante de l’algorithme AESD) permet d’approcher infailliblement le minimum global  $x^* = (1,1)$ .

4.3.3.2.4. Le problème (P4)

Le but de ce problème est de minimiser la fonction non partout différentiable du problème (P4) sous les contraintes  $x \in [-2, \infty[ \times [-4, -2]$ . Le point initial  $x_0 = (6, -4)$  est faisable et se situe sur les bornes.

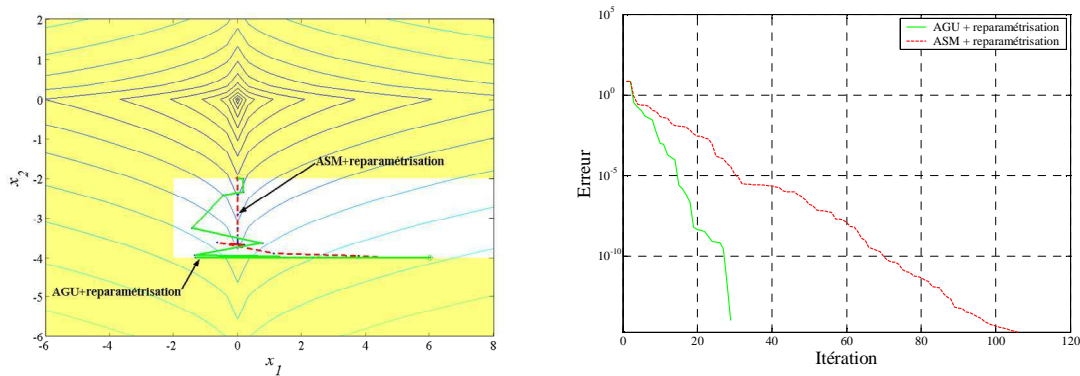
La même batterie de tests exhibe les résultats synthétisés par tableau ci-dessous :

Algorithme	Méthodes de recherche directe			Méthodes de descente à base de gradient	
	11	12	13	14	15
Critère $\bar{f}$	2.0000	2.0000	2.0000	2.0000	2.0000
Solution $\bar{x}$	(0, -2.0000)	(0, -2.0000)	(0, -2.0000)	(0, -2.0000)	(0, -2.0000)
Nb itérations	104	77	30	43	27
Nb éval.	397	173	278	189	93
Nb éval. $\nabla f(x)$	0	0	0	172	78
$\ \nabla f(\bar{x})\ _2$	5.0002e+6	5.4567e+7	3.6667e+7	2.3727e+7	4.9998e+8
Temps de calcul	0.7190	0.3200	0.4700	1.7500	0.2950

**Tab. 4.10: Résultat du problème (P4) avec contraintes de bornes**

Pour ce problème, les résultats obtenus avec les différents algorithmes sont satisfaisants. Même les algorithmes 11 et 14 à base de la technique de prise en compte des contraintes par projection parviennent à converger efficacement en suivant initialement le même chemin que dans le cas non contraint.

Les résultats de l’algorithme “patternserach” sont meilleurs. En effet, cet algorithme tire avantage de l’orientation de son motif généralisé qui présente deux axes parallèles aux failles du problème (P4) ( $x_1 = 0$  et  $x_2 = 0$ ) et converge ainsi rapidement au bout de 30 itérations seulement. Un changement d’orientation du problème (P4) risque de changer fortement cette performance.



**Fig. 4.26 Evolution de l’algorithme AGU et ASM pour le problème (P4) avec contraintes de bornes**

Par ailleurs, une comparaison des chemins tracés par les itérations des algorithmes 12 et 15, avec le cas non contraint (cf. figure 4.21), montre que la reparamétrisation (changement de variable à base de la fonction sinus) modifie la géométrie de la fonction critère et empêche, contrairement au cas non contraint, une progression rapide dans le sens des  $x_2$  croissants. Les trajectoires évoluent plutôt parallèlement à la contrainte initialement active  $x_2 = -4$  avant de modifier de direction de descente une fois  $x_1 = 0$  est franchie. Malgré cela, la convergence de l’algorithme AGU reste nettement plus rapide que celle de l’algorithme ASM.

4.3.3.2.5. Le problème (P5)

Dans cette section, nous discutons le problème d’abscisse spectrale sous la contrainte d’appartenance à l’espace paramétrique admissible  $x \in [1, \infty] \times \mathfrak{R}$ . L’itérée initiale  $x_0 = (-7,5)$  est en dehors de ce dernier.

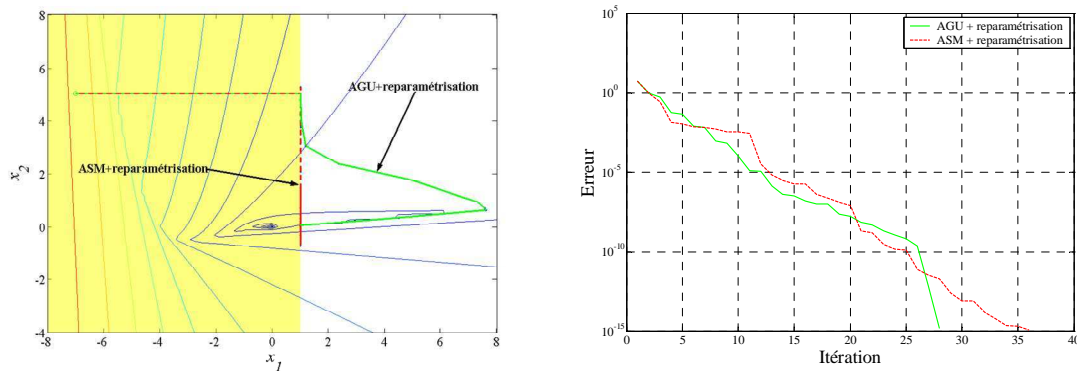
Algorithme	Méthodes de recherche directe			Méthodes de descente à base de gradient	
	11	12	13	14	15
Critère $\bar{f}$	0.4058	0.4058	0.5978	0.4058	0.4058
Solution $\bar{x}$	(1.000, 5.03e-2)	(1.000, 5.03e-2)	(23.121, 2.0000)	(1.000, 5.03e-2)	(1.000, 5.03e-2)
Nb itérations	63	34	54	32	26
Nb éval.	265	168	169	509	109
Nb éval. $\nabla f(x)$	0	0	0	128	53
$\ \nabla f(\bar{x})\ _2$	0.8554	0.8553	0.02418	0.8554	0.8554
Temps de calcul	0.4650	0.0900	0.5100	1.1090	0.4890

**Tab. 4.11: Résultat du problème (P5) avec contraintes de bornes**

Les relevés du tableau 4.11 confirment l’ensemble des analyses déjà faites, à savoir :

L’avantage de l’approche de prise en compte des contraintes de bord par reparamétrisation par rapport à l’approche par projection.

L’efficacité de la stratégie de descente AGU par rapport à l’algorithme ASM et la fonction “patternserach”.



**Fig. 4.27 Evolution de l’algorithme AGU et ASM pour le problème (P5) avec contraintes de bornes**

La figure ci-dessus décrit l’évolution des algorithmes 12 et 15 pour le problème (P5). La trajectoire suivie par ce dernier est similaire à celle trouvée dans le cas non contraint. Les itérées s’écartent petit à petit de la contrainte  $x_1 = 1$  et retrouvent une descente quadratique selon la stratégie BFGS avant de terminer la convergence sur la zone non différentiable. Les courbes de décroissance des algorithmes AGU et ASM sont assez comparables.

## 4.4. Conclusions

En nous basant sur les techniques d’optimisation qui ont été exposées dans le chapitre précédent et après de nombreux essais et analyses des critères classiquement utilisés en Automatique, nous avons développé des algorithmes spécifiques. Ils sont adaptés aux types de problèmes rencontrés : différentiables presque partout, avec une géométrie particulière des zones non différentiables (de type nappe).

Les deux familles retenues sont :

- les algorithmes de recherche directe issus du simplexe de Nelder-Mead
- les algorithmes de descente à base des généralisations du gradient

Dans un premier temps, nous avons modifié l’algorithme de Nelder-Mead afin d’éviter le problème de dégénérescence. Ceci arrive fréquemment pour les problèmes mal conditionnés et/ou non différentiables et réduit “sensiblement” l’efficacité de la méthode du simplexe. Du point de vue structurel, la solution proposée consiste à effectuer une série de tests détectant le type de dégénérescences et à réinitialiser la recherche de façon adaptée dans des voisinages des points courants afin de décerner les vrais optima. Dans le cas d’un problème d’optimisation avec des contraintes de bornes, le problème de dégénérescence sur les bornes est discuté et deux résultats ont été proposés pour prendre en compte les contraintes : prise en compte des contraintes par projection ou par reparamétrisation.

Dans un deuxième temps, nous avons développé un algorithme à base d’une notion généralisée du gradient qui est l’ $\varepsilon$ -sous-différentiel de Clarke. Cette notion mathématique possède plusieurs propriétés intéressantes qui permettent entre autre de l’approximer par un ensemble convexe fermé défini à base d’un nombre fini de gradients au voisinage. Par la suite, l’association de ce résultat à la notion de  $\varepsilon$ -stationnarité de Clarke permet de déterminer une bonne direction de descente.

L’ensemble de ces propriétés est utilisé pour développer un premier algorithme de descente dit AESD. Ce dernier s’avère très coûteux en terme de temps calcul, mais très efficace en nombre d’itérations. Cela confirme que  $\varepsilon$ -sous-différentiel est un bon choix stratégique pour se déplacer dans l’espace de recherche mais que son évaluation est elle-même très coûteuses. En effet, à chaque itération, Il nécessite le calcul de plusieurs gradients.

Constatant que dans les zones “régulières” le gradient et le  $\varepsilon$ -sous-gradient se confondent, un deuxième algorithme de descente est alors proposé. Il commute entre deux stratégies de descente. Une première phase initiale à base de gradient et une deuxième phase terminale à base de l’approximation du  $\varepsilon$ -sous-différentiel. Afin d’améliorer encore cet algorithme, sa phase initiale est affinée, et utilise une stratégie de descente de type quasi-Newton avec une mise à jour BFGS. L’algorithme résultant est nommé AGU.

Une série de problèmes tests est utilisée dans ce chapitre. Bien que difficiles pour l’optimisation, ces problèmes de référence sont simples à reproduire car ils ont une description analytique. Ils sont de complexités différentes et couvrent plusieurs situations critiques : mixte, mal paramétré, non différentiable, non Lipschitz.

L’ensemble des algorithmes développés est testé et comparé à d’autres algorithmes d’optimisation plus classiques et faisant référence en optimisation non linéaire. Les résultats numérique montrent que certains algorithmes classiques pourtant réputés échouent à résoudre certains problèmes test. En revanche, les résultats des algorithmes ASM et AGU prouvent leurs efficacités et performances vis-à-vis aux problèmes précités et confirment leur aptitude à prendre en compte les non différentiabilités.

Notons que la technique de prise en compte des contraintes par une reparamétrisation à base de la fonction périodique sinus semble très efficace et permet de résoudre n’importe quel problème d’optimisation sous contraintes de bornes avec des algorithmes d’optimisation sans contraintes.

Des applications à l’Automatique seront étudiées au chapitre 6 afin d’analyser et de confirmer l’apport de ces algorithmes.





# Chapitre 5

## Analyse du $\varepsilon$ -sous différentiel et calcul du gradient

Autant que nous ayons pu en juger, l'optimisation dans un vrai contexte de faisabilité d'un cahier des charges industriel reste souvent confinée à des approches très rudimentaires. Les simulations numériques étant lourdes, il est fréquent de procéder en deux temps. D'abord on réalise une approximation du résultat de la simulation numérique par une fonction simplifiée (polynomiale) d'un petit nombre de paramètres. La détermination de cette fonction est par exemple issue d'un plan d'expérience ou d'une simulation numérique simplifiée. Dans un second temps, ayant ainsi rendues les évaluations de la fonction gratuites, un algorithme d'optimisation du type gradient est utilisé. Une alternative à cette approche en deux temps est de conserver la simulation numérique en tant que telle et de l'inclure dans un algorithme d'ordre 0 du type simplexe.

Notre démarche part du constat que dans le monde de la recherche plus académique, à la fois des méthodes de calcul du gradient efficaces et des méthodes d'optimisation avancées utilisant ce gradient existent. Nous avons pour objectif de vérifier la portabilité de telles méthodes pour des problèmes de validation de cahiers des charges industriels, rendant ainsi possible l'utilisation directe de la simulation numérique ou d'une de ses approximations. Des démarches similaires ont déjà lieu dans des domaines industriels très en pointe du point de vue de la simulation numérique (structures aéronautiques).

Dans cette optique, le chapitre précédent nous a permis d'affirmer que les méthodes de gradient peuvent répondre particulièrement efficacement à nos objectifs. Il reste à montrer que des méthodes de calcul de gradient adéquates existent, et, à choisir celle qui répond du mieux à nos exigences. Si on met à part le cas du calcul symbolique, très complexe et moins pertinent dans le contexte industriel, deux grands types de méthodes ressortent : les méthodes directes comprenant les différences finies, la dérivation complexe, les fonctions de sensibilité et la différentiation automatique en mode direct, et les méthodes basées sur la résolution de l'adjoint comprenant notamment la différentiation automatique en mode adjoint.

Pour notre étude, l'utilisation de l'algorithme AGU fait intervenir un grand nombre d'évaluations du gradient  $\nabla f(x)$ , qui va représenter une composante majeure de l'ensemble des calculs. Il importe donc de rendre ce calcul aussi rapide et précis que possible. Ainsi, le fil conducteur de ce chapitre sera le suivant :

Dans une première partie, nous exposons les outils de calcul nécessaires à l'estimation du  $\varepsilon$ -sous-différentiel utilisé dans l'algorithme AGU. Comme nous l'avons déjà vu au chapitre précédent, l'estimation du  $\varepsilon$ -sous-différentiel d'une fonctionnelle en un point donné de l'espace paramétrique peut se ramener au calcul d'un nombre fini de gradients échantillonnés de cette fonctionnelle autour de ce même point [Bur02]. Nous analyserons ainsi la qualité de l'estimation du  $\varepsilon$ -sous-différentiel et donc de la direction de descente vis-à-vis du nombre d'échantillons et de la précision de calcul des

gradients échantillonnés. Par ailleurs, les espaces fonctionnels ne seront que rarement introduits car nous n'en aurons pas besoin ici<sup>1</sup>. Les lecteurs intéressés par des compléments mathématiques pourront se référer aux travaux publiés par Clarke [Cla75], [Cla83] et [Cla90].

La seconde partie de ce chapitre est dédiée à l'exposition des techniques de calcul du gradient. L'objet de cette section n'est pas de décrire de manière exhaustive toutes les méthodes de différentiation et de calcul du gradient tant la bibliographie est importante et les approches souvent différentes selon la communauté scientifique (automaticien, mathématicien appliqué, informaticien...) abordant le sujet. Plus humblement, nous souhaitons présenter quelques idées essentielles, susceptibles de faciliter la résolution des problèmes d'optimisation formulés.

## 5.1. Estimation du $\varepsilon$ -sous-différentiel de Clarke

Rappelons tout d'abord l'expression de l'ensemble  $\hat{G}_\varepsilon(x)$  approximant le  $\varepsilon$ -sous différentiel de Clarke proposée par Burke et al. [Bur02]

$$\hat{G}_\varepsilon(x) = \text{clconv} \left( \bigcup_{i=1}^m (\nabla f(x+b_i) \cap D) \right) \text{ avec } \|b_i\|_2 \leq \varepsilon \quad (5-1)$$

où  $m$  est le nombre d'échantillons  $b_i$  à l'intérieur de l'hypersphère qui a pour centre l'origine 0 et pour rayon  $\varepsilon$ .  $D \in \mathfrak{R}^n$  est l'ensemble dense et ouvert où  $f$  est localement Lipschitz continue et continuellement différentiable.  $\text{clconv}()$  est l'opérateur d'enveloppe convexe fermée.

L'ensemble  $\hat{G}_\varepsilon(x)$  représente donc l'enveloppe convexe fermée constituée à base d'un tirage de gradients autour du point  $x$ . C'est un sous-ensemble du  $\varepsilon$ -sous-différentiel  $\bar{\partial}_\varepsilon f(x)$  mais son calcul est de loin plus simple ; il dépend seulement du calcul d'un nombre fini  $m$  de gradients échantillonnés.

Le choix de  $m=1$  revient à calculer un seul gradient près de  $x$ . Ce choix n'est pas plus efficace que l'estimation d'un sous-gradient lorsqu'il s'agit d'un point non différentiable  $x$ . De même, le choix d'un grand nombre  $m$  de gradients risque d'être inutile et redondant ce qui rend les itérations de l'algorithme d'optimisation plus lourdes en terme de temps de calcul. La situation idéale serait plutôt de s'adapter selon la topologie du critère traité pour choisir le nombre  $m$  et déterminer ainsi le plus simplement possible une direction de descente efficace.

Après avoir adopté une première solution dans le chapitre 4 qui consiste à commuter entre plusieurs stratégies de descente de type gradient (cf. sections 4.2.4, 4.2.5 et 4.2.6), nous allons explorer la question de choix du nombre de gradients afin d'alléger encore plus les dernières itérations de l'algorithme AGU.

### 5.1.1. Échantillonnage uniforme dans une hyperboule

Lorsqu'on est près d'une zone non différentiable, il est suffisant d'échantillonner un gradient de chaque sous-espace délimité par les zones non différentiables. En effet, les valeurs des gradients très proches dans ces sous-espaces ont des valeurs également très proches, et l'ajout de ces échantillons similaires à des valeurs de gradients déjà existantes ne change pas l'enveloppe convexe de l'approximant  $\hat{G}_\varepsilon(x)$ . Ces gradients donnent l'information nécessaire pour construire une bonne approximation du  $\varepsilon$ -sous-différentiel et donc une bonne direction de descente.

<sup>1</sup> Notre objectif est d'utiliser le plus rigoureusement possible des outils mathématiques en vue de résoudre un problème d'ingénieurs.

Cependant, cet échantillonnage nécessite une étude géométrique spécifique et il n'est pas facilement réalisable. En particulier, quand il s'agit de traiter des critères implicites où les zones non différentiables sont très difficiles à définir, nous lui préférons une méthode générique.

Face à cette difficulté, l'idée d'effectuer un échantillonnage uniforme autour du point traité  $x$  nous permet de garantir une distribution homogène d'échantillons équiprobables à l'intérieur de hyperboule  $B = \{x \in \mathcal{R}^n \mid \|x\|_2 \leq \varepsilon\}$ . Notons la similitude de cette démarche avec la définition formelle. Reste à construire un échantillonnage uniforme...

Cependant, quel est l'algorithme le plus efficace pour effectuer un tirage aléatoire suivant une distribution aléatoire uniforme hypersphérique ?

S'il est facile de tirer aléatoirement un point dans un hypercube (en tirant chaque coordonnée indépendamment et uniformément sur un segment) ce n'est pas le cas pour une hyperboule.

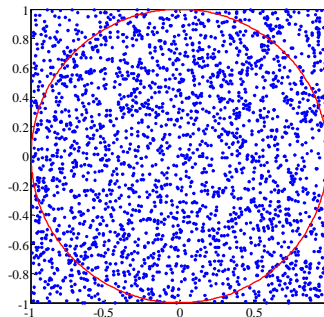
L'idée de partir d'un tirage aléatoire uniforme dans un hypercube et de ne conserver que les points d'une hyperboule contenue n'est pas une bonne idée du point de vue de la complexité (en terme de nombre de calculs).

En effet, si pour le cas d'un espace à une dimension, il suffit de choisir une distribution uniforme dans la direction cartésienne, cette solution devient très vite inexploitable pour les dimensions supérieures. La probabilité pour que le point issu d'un échantillonnage uniforme dans les directions rectangulaires soit à l'intérieur de l'hyperboule diminue avec la dimension de cette dernière. Le tableau 5.1 rappelle le rapport volumétrique d'une hyperboule unitaire par rapport à un hypercube unitaire de même dimension.

La dimension	Volume de l'hypercube	Volume de l'hyperboule	Le rapport
1D	2	2	1
2D	4	$\pi$	$\pi / 4$
3D	8	$4\pi / 3$	$\pi / 6$
nD	$2^n$	$\pi^{n/2} / \Gamma(1 + n/2)$	$2^{-n} \pi^{n/2} / \Gamma(1 + n/2)$

**Tab. 5.1: Rapports volumétriques d'une hyperboule et d'un hypercube unitaires**

A titre d'exemple, ce tableau montre que pour  $n = 10$ , la probabilité pour qu'un point de l'hypercube de coté 2 soit dans l'hyperboule de rayon 1 contenue est de 0.25%. Cette infime probabilité prouve l'inefficacité de cette technique : il faut tirer de très nombreux point dans un hypercube pour en obtenir quelques-uns dans l'hyperboule. Nous illustrons ce résultat à travers l'exemple à deux dimensions suivant.



**Figure. 5.1: Echantillonnage uniforme à base de coordonnées rectangulaires**

La deuxième solution venant à l'esprit consiste à se placer dans un espace en coordonnées polaires et à tirer dans une loi uniforme chaque coordonnée. Dans un espace à  $n$  dimensions, on tire ainsi le rayon

$r$  et les angles  $a_1, a_2, \dots, a_n$ . Puis on passe en coordonnées rectangulaires pour avoir les angles  $x_1, x_2, \dots, x_n$ . Les résultats de cette alternative sont incorrects comme le montre la figure qui suit :

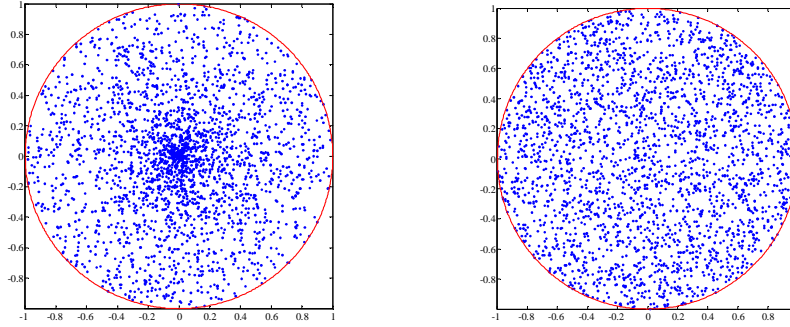


Figure 5.2: Echantillonnage uniforme à base de coordonnées polaires

En effet, il est donc incorrect d'utiliser deux variables uniformément distribués  $r \in [0,1]$  et  $\theta \in [0,2\pi]$  et de choisir

$$\begin{cases} x = r \cos(\theta) \\ y = r \sin(\theta) \end{cases} \quad (5-2)$$

car l'élément de surface  $dA = 2\pi r dr$  dépend du rayon  $r$  ce qui crée une concentration de points au centre (figure 5.1 à gauche). La transformation correcte est donné par :

$$\begin{cases} x = \sqrt{r} \cos(\theta) \\ y = \sqrt{r} \sin(\theta) \end{cases} \quad (5-3)$$

Le résultat de cette dernière est donné par la figure 5.2 à droite.

La généralisation à une dimension  $n$  consiste à calculer chaque coordonnée cartésienne  $x_i$  en utilisant la formule

$$x_i = \frac{r_i^{1/n} a_i}{\|a\|_2} = \frac{r_i^{1/n} a_i}{\sqrt{\sum_{j=1}^n a_j^2}} \quad (5-4)$$

où  $r_i$  suit une loi uniforme  $U_{[0,1]}$  et  $a_i$  une loi normale  $N(0,1)$ .

Nous remarquons que pour  $n = 2$ , l'expression (5-4) se ramène à (5-3) car si  $\theta = \arctan(a_2/a_1)$  suit une loi uniforme  $U_{[0,2\pi]}$ , alors  $a_1$  et  $a_2$  suivent tous les deux une loi normale  $N(0,1)$  [Knu97].

Les résultats de cette approche pour le cas trois dimensions sont donnés par la figure ci-dessous.

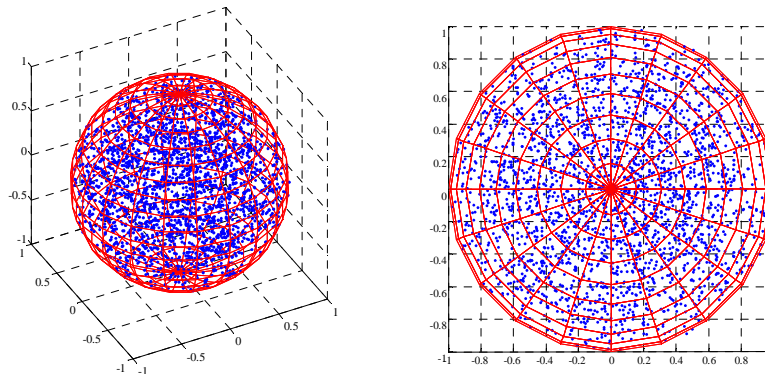


Figure 5.3: Echantillonnage sphérique uniforme à base de la relation (5-4)

### 5.1.2. Choix du nombre d'échantillons $m$

Lorsqu'il s'agit d'un espace paramétrique de dimension  $n$  avec éventuellement  $n$  zones non différentiables, le nombre de sous-espaces produits est génériquement de  $2 \cdot n$ . Partant de ce constat, le nombre minimal conseillé d'échantillons est égal à  $m = 2n$ .

Les résultats de 100 exécutions de l'algorithme AGU pour la minimisation de l'abscisse spectrale du problème (P5) du précédent chapitre (cf. section 4.) ont permis d'évaluer le taux de convergence vers la solution du problème (P5) en fonction du nombre d'échantillons  $m$ . La figure 5.4 permet de confirmer qu'à partir d'un nombre d'échantillons égal à 2 fois la dimension de l'espace d'optimisation, l'algorithme AGU tend vers la solution "presque" toujours.

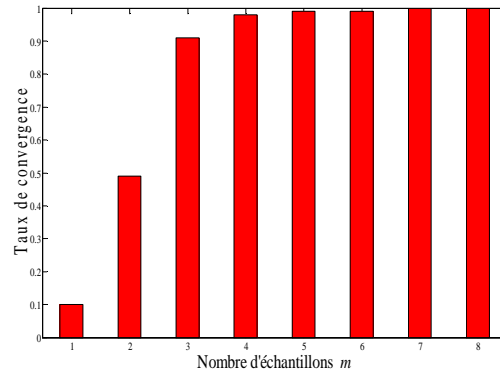


Figure 5.4: Taux de convergence de l'algorithme AGU en fonction du nombre d'échantillons  $m$

Ce résultat ne concerne pas que cet exemple, il est vérifié pour l'ensemble des problèmes que nous avons traités.

Dans ce qui suit, nous proposons d'étudier les propriétés statistiques de l'estimation du  $\varepsilon$ -sous-différentiel pour deux fonctions analytiques scalaires non différentiables.

La première fonction est la fonction valeur absolue  $f(x) = |x|$ . Cette fonction possède un sous-différentiel connu défini par

$$\partial f(x) = \begin{cases} -1 & \text{si } x < 0 \\ [-1,1] & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases} \quad (5-5)$$

Pour un nombre d'échantillons égal à  $m$ , nous pouvons ainsi définir les probabilités de tirer  $m_1$  points dans le demi-plan droit et  $m - m_1$  points dans le demi-plan gauche. La fonction  $f(x)$  est suffisamment simple pour pouvoir calculer ces probabilités et par suite la moyenne et la variance de l'estimée du  $\varepsilon$ -sous-différentiel de  $f$  à l'origine. Ces dernières dépendent du rayon  $\varepsilon$  et du nombre d'échantillons  $m$ . Pour la fonction  $f(x) = |x|$ , ces fonctions sont définies par :

$$M(m, \varepsilon) = 2^{-n} \left( - \left( 1 - \frac{x}{\varepsilon} \right)^n + \left( 1 + \frac{x}{\varepsilon} \right)^n \right) \quad (5-6)$$

$$\sigma^2(m, \varepsilon) = P_A(x) \left( \hat{G}_\varepsilon(x > 0) - M(m, \varepsilon) \right)^2 + P_B(x) \left( \hat{G}_\varepsilon(x < 0) - M(m, \varepsilon) \right)^2 + P_{A,B}(x) \left( \hat{G}_\varepsilon(0) - M(m, \varepsilon) \right)^2 \quad (5-7)$$

où  $P_{A,B}(x) = \sum_{k=1}^{m-1} C_m^k P_A^k P_B^{m-k}$  ;  $P_A(x) = (\varepsilon + x)/2/x$  et  $P_B(x) = (\varepsilon - x)/2/x$

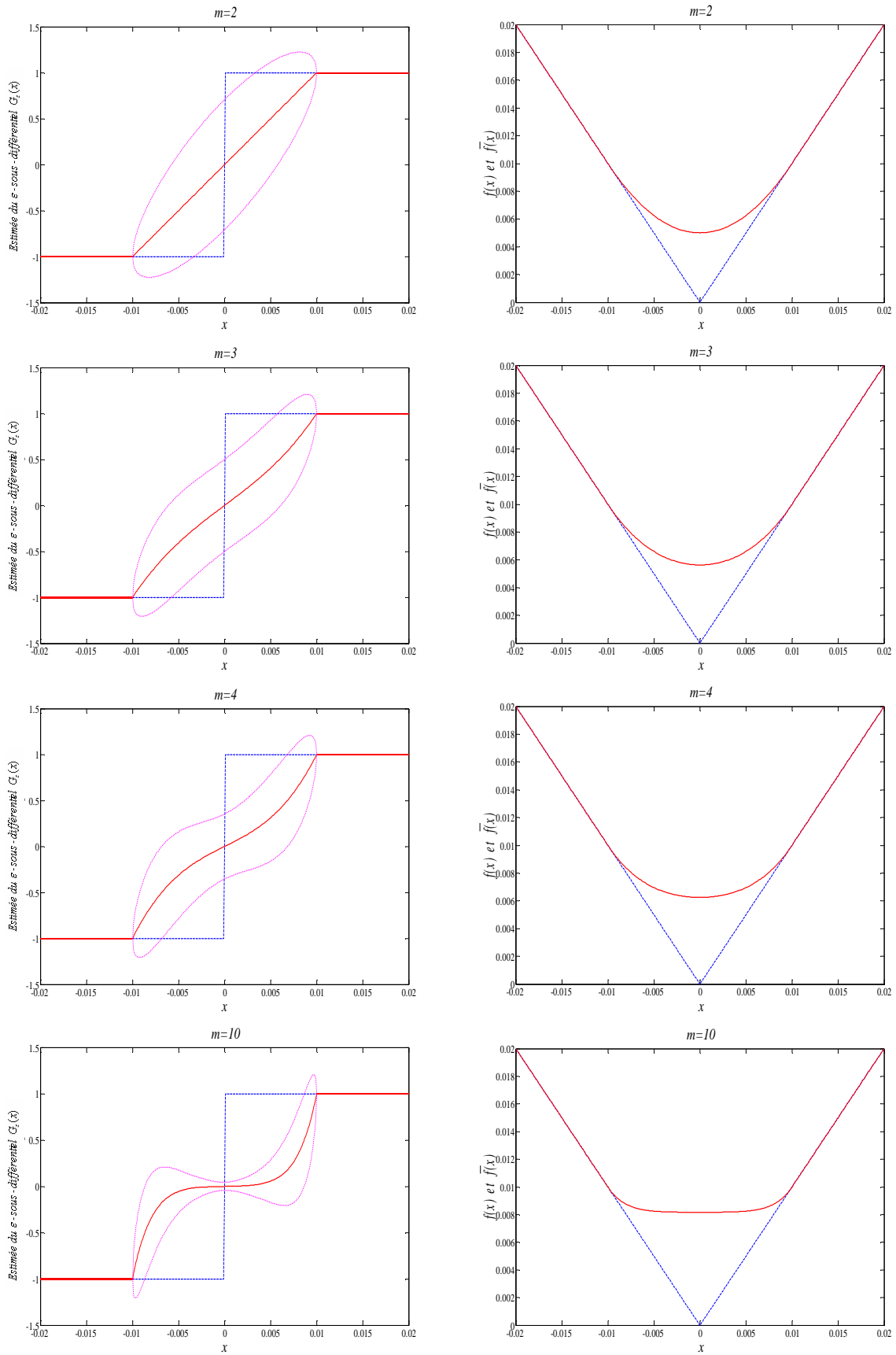


Figure. 5.5: Evolutions de la moyenne et de l'écart-type de l'estimée du  $\varepsilon$ -sous-différentiel  $G_\varepsilon(x)$  pour la fonction  $f(x) = |x|$

La figure 5.5 permet de comparer l'estimée du  $\varepsilon$ -sous-différentiel  $\hat{G}_\varepsilon(x)$  au sous-différentiel  $\partial f(x)$  pour plusieurs nombres d'échantillons  $m$ . Les tracés en tirés dans la colonne de droite et de gauche de la figure représentent le sous-différentiel  $\partial f(x)$  et la fonction  $f(x)$  respectivement. Les courbes en lignes continues dans la colonne de droite et de gauche représentent la moyenne de l'estimée du  $\varepsilon$ -sous-différentiel  $\hat{G}_\varepsilon(x)$  et son intégrale respectivement. Finalement, les courbes en pointillés fins représentent l'écart-type de l'estimée  $\hat{G}_\varepsilon(x)$ .

Nous observons que plus le nombre d'échantillons  $m$  augmente plus la moyenne de l'estimée  $\hat{G}_\varepsilon(x)$  prend la forme d'un relais entre  $-1$  et  $+1$  avec une zone morte entre  $-\varepsilon$  et  $+\varepsilon$  ayant pour valeur 0 (ce qui correspond au  $\varepsilon$ -sous-différentiel théorique). Ceci se traduit par une fonction "équivalente" à minimiser  $\bar{f}(x)$  quadratique de plus en plus plate autour de l'origine. Cette nouvelle fonction est lisse autour de l'optimum et conserve la propriété de convexité de la fonction  $f(x)$ . Ainsi, la phase finale de l'algorithme AGU sera équivalente à un simple algorithme de plus profonde descente où la fonction à minimiser est  $f(x)$  et le gradient équivalent est égal à la dérivée  $\partial \bar{f}(x)/\partial x$  de la fonction lisse  $\bar{f}(x)$ . Dans ce cas, ce type d'algorithmes suffit amplement pour atteindre l'optimum car le gradient calculé dans la l'intervalle  $[-\varepsilon, \varepsilon]$  sera toujours ascendant ce qui permet de déterminer une bonne direction de descente.

Comme nous l'avons déjà avancé, la courbe de la fonction  $\bar{f}$  ressemble beaucoup à la forme d'une noue de toiture (cf. figure 5.7). La descente lors de la phase finale de l'algorithme AGU est similaire à la récupération des eaux de pluie dans cette partie du toit.



**Figure 5.6: Analogie entre la fonction équivalente  $\bar{f}$  et une noue de toiture**

La deuxième fonction traitée est la fonction non différentiable  $f(x) = \sqrt{|x|}$ . La dérivée de cette fonction est discontinue à l'origine. Bien que ces situations soient hors du champ d'étude initial, nous avons pu constater que les algorithmes développés n'avaient aucune difficulté à traiter ce type de situation. En fait, l'estimateur  $\hat{G}_\varepsilon(x)$  produit le même type de "lissage" que dans le cas simplement non différentiable et permet également un bon comportement de l'optimisation. Par exemple, la figure 5.7 montre la variation de la moyenne et l'écart-type de l'estimée du  $\varepsilon$ -sous-différentiel pour cette fonction ainsi que sa fonction équivalente  $\bar{f}(x)$ .

Ces résultats statistiques de l'estimée du  $\varepsilon$ -sous-différentiel montrent que malgré une discontinuité infinie de la dérivée de  $f(x)$ , la moyenne  $M(m, \varepsilon)$  de l'estimée  $\hat{G}_\varepsilon(x)$  est continue à zéro et présente une variation continue et lisse pour un petit nombre d'échantillons. En augmentant le nombre d'échantillons  $m$ , cette moyenne  $M(m, \varepsilon)$  devient proche d'une forme cubique passant par l'origine ce qui crée deux discontinuités aux points  $x = -\varepsilon$  et  $x = +\varepsilon$ . Quant à l'intervalle formé par l'écart-type  $\sigma(m, \varepsilon)$ , il devient plus serré autour de la moyenne  $M(m, \varepsilon)$ .



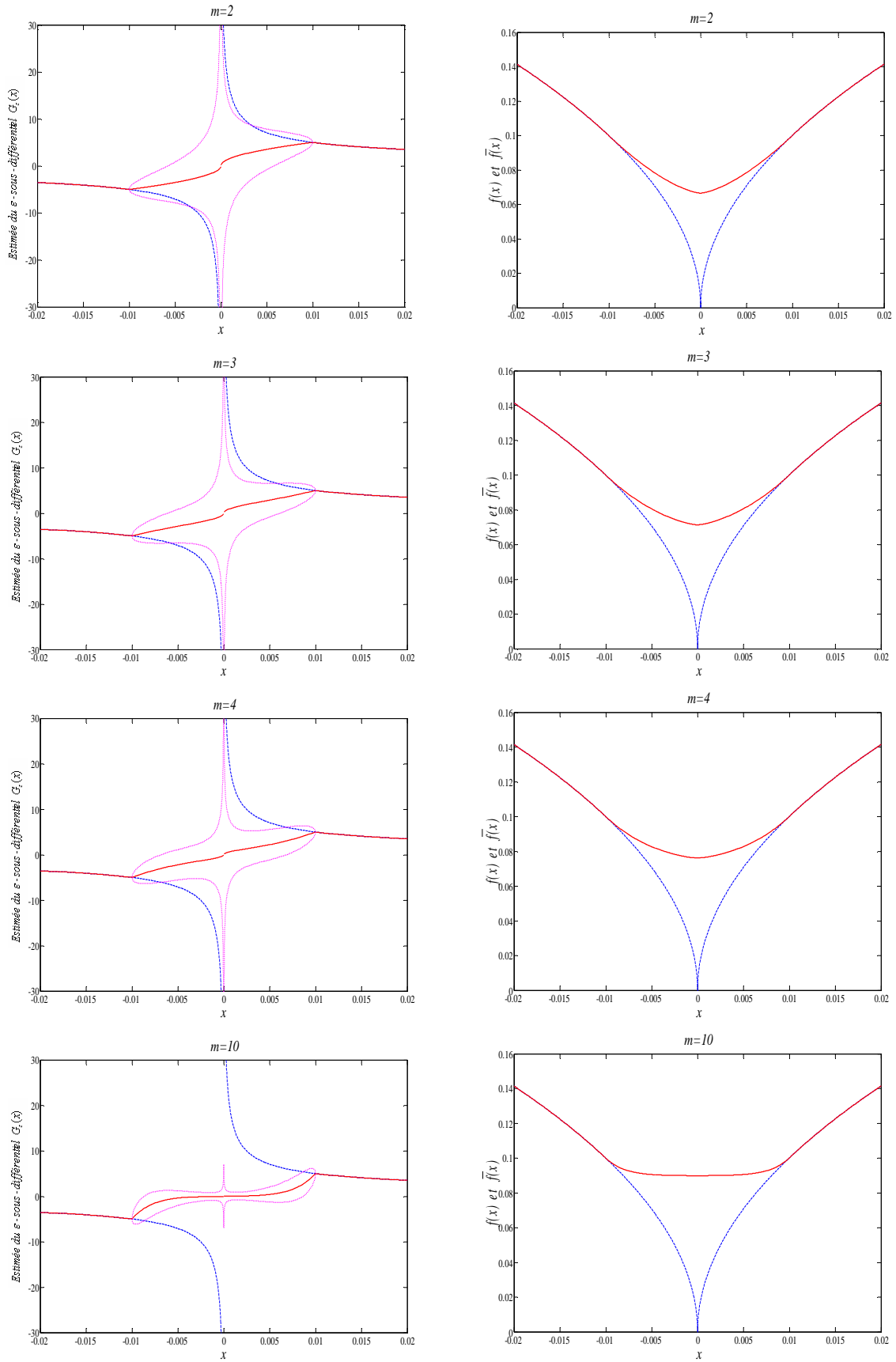


Figure. 5.7: Evolutions de la moyenne et de l'écart-type de l'estimée du  $\varepsilon$ -sous-différentiel  $G_\varepsilon(x)$  pour la fonction  $f(x) = \sqrt{|x|}$

In fine, le sous-différentiel de Clarke peut être assimilé à un opérateur de lissage pour les fonctions non différentiables. Il permet de ramener la minimisation d’une fonction non différentiable en la minimisation d’une fonction de même type (convexe, concave ou autre) où les zones non différentiables deviennent quadratiques.

### 5.1.3. Détection de zones non différentiables et ouverture d’un $\varepsilon$ -sous-différentiel

Une fois les gradients échantillonnés calculés, les itérations des algorithmes AESD, AESDM et AGU peuvent bien être exécutées. Toutefois, nous avons remarqué que ces derniers convergent souvent vers des zones non différentiables où d’autres algorithmes mal adaptés se bloquent.

Vu que nos algorithmes ont été conçus pour pouvoir trouver une bonne direction de descente autour des ces régions, il est alors possible d’utiliser l’information issue de l’échantillonnage isotrope des gradients autour d’un point donné pour pouvoir qualifier son appartenance à une zone non différentiable.

Pour ce faire, nous avons défini un critère qui mesure la dispersion des gradients échantillonnés. Il est basé sur le calcul des angles entre tous les gradients. En effet, en évaluant l’angle maximal parmi toutes les combinaisons des angles possibles entre gradients, nous pouvons juger de l’ouverture (ou de la fermeture) de l’ensemble approximant le  $\varepsilon$ -sous-différentiel. Plus cet angle est proche de  $\pi$ , plus la probabilité que ce point soit non différentiable est proche de 1.

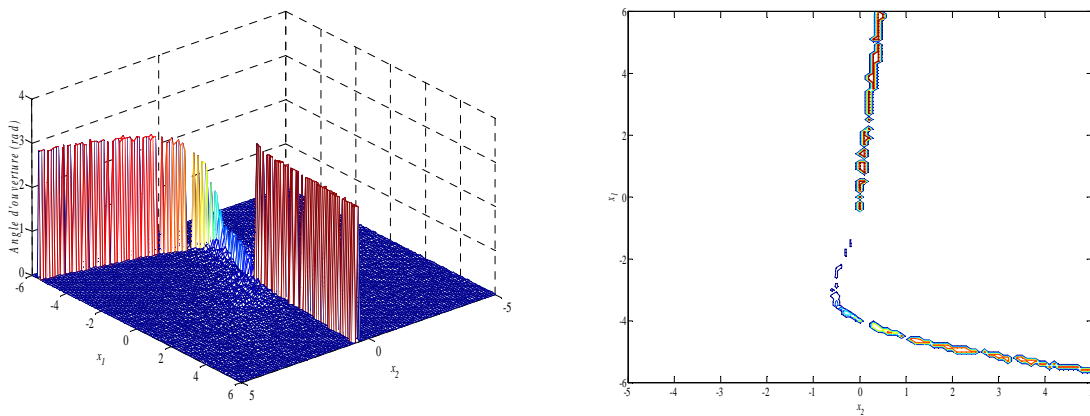


Figure. 5.8: Zones non différentiables du critère d’abscisse spectrale (P5)

La figure ci-dessus illustre la variation de cet angle maximal pour la fonction coût définie par le problème P5 (cf. section 4.3.2). Nous remarquons que le contour de la zone non différentiable a été détecté.

En effet, nous pourrions utiliser ce critère pour détecter les zones non différentiables et commuter entre la stratégie de gradient “classique” et la stratégie du  $\varepsilon$ -sous différentiel dans l’algorithme AESDM ou entre la stratégie quasi-Newton et la stratégie du  $\varepsilon$ -sous différentiel dans l’algorithme AGU. Toutefois, la complexité algorithmique liée au calcul de ce critère est du même ordre que celle du calcul du sous gradient, il n’y a donc aucun avantage à évaluer ce critère en un point et ne pas utiliser un sous gradient en ce même point.

Ainsi, ce critère représente surtout un bon outil d'analyse mais son implémentation pour des besoins algorithmiques n'est pas adaptée, c'est pourquoi cette option est écartée.

## 5.2. Calcul et estimation du gradient

Dans cette section, nous donnons une sommaire description des différentes techniques développées pour le calcul du gradient. Nous consacrons un traitement un peu plus exhaustif aux différences finies et aux calculs de sensibilités, techniques que nous avons choisies de mettre en œuvre car très adaptées aux problèmes de simulations rencontrés dans les cahiers des charges. Nous indiquerons également des éléments qui permettent de gérer le problème du pas de différentiation inhérent aux approches numériques.

Après avoir exposé les méthodes dans un cadre général, nous donnerons quelques exemples explicites pour faire sortir les avantages et les inconvénients et verrons comment mener à bien ce calcul dans une situation concrète.

### 5.2.1. Différences finies

Comme la plus part des méthodes d'optimisation les plus courantes, l'algorithme de gradient universel AGU est fondé sur l'usage des dérivées d'une fonctionnelle par rapport à des paramètres de décision. Quand on ne peut les calculer explicitement (ce qui est presque toujours le cas), on peut parfois évaluer une approximation de ces dérivées par *différences finies*.

Cette technique consiste à calculer une approximation de la dérivée d'une fonction  $f(x)$  en évaluant la dérivée d'une fonction estimée  $\hat{f}(x)$  de cette fonction. Ceci peut être symboliquement schématisé par :

$$f(x) \longrightarrow \hat{f}(x) \Rightarrow \frac{\partial}{\partial x}(f(x)) \longrightarrow \frac{\partial}{\partial x}(\hat{f}(x)) \quad (5-8)$$

L'estimée  $\hat{f}(x)$  est souvent obtenue via une interpolation polynomiale où il s'agit de faire passer par les points de colocation un polynôme d'interpolation, puis à dériver celui-ci le nombre de fois nécessaire. On peut ainsi estimer la dérivée aux points de colocation ou entre deux.

Le cas le plus simple est celui où il n'y a que deux points de colocation :  $(x_0, f_0)$  et  $(x_1, f_1)$ . Par ces points passe la droite d'équation  $\hat{f}(x)$  donnée par :

$$\hat{f}_1(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) \quad (5-9)$$

L'estimation de la dérivée première équivaut donc au coefficient directeur de la droite :

$$f'(x) \approx \hat{f}'_1(x) = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f_1 - f_0}{h}, \quad x \in [x_0, x_1] \quad (5-10)$$

Notons qu'elle prend la même valeur en tout point de l'intervalle  $[x_0, x_1]$ . Par ailleurs, la dérivée seconde et toutes les dérivées supérieures sont nulles.

Pour savoir quelle confiance accorder à l'expression ci-dessus, il faut connaître l'erreur.

$$f(x) = \hat{f}_n(x) + \varepsilon_n(x) \Rightarrow f'(x) = \hat{f}'_n(x) + \varepsilon'_n(x) \quad (5-11)$$

Supposons dorénavant que l'échantillonnage est régulier (les points sont équidistants), et appelons  $h = x_{i+1} - x_i$  pas entre deux abscisses voisines. On peut alors montrer que :

$$\varepsilon'_n = (-1)^n \frac{f^{(n+1)}(\xi)}{(n+1)!} h^n, \quad \xi \in [x_0, x_n] \quad (5-12)$$

L'erreur varie donc comme  $\varepsilon'_n \sim h^n$ . On dira qu'elle est d'ordre  $n$  et on écrira fréquemment de façon abrégée  $\varepsilon'_n(x) \sim o(h^n)$ .

Avec deux points, l'expression de la dérivée première d'ordre 1 peut s'écrire de deux façons différentes. Soit on estime la pente de la droite qui passe par le point de colocation suivant (différence avant), soit on prend la pente de la droite passant par le point qui précède (différence arrière).

$$f'(x) = \frac{f(x_{k+1}) - f(x_k)}{h} + o(h) \quad \text{différence avant d'ordre 1}$$

$$f'(x) = \frac{f(x_k) - f(x_{k+1})}{h} + o(h) \quad \text{différence arrière d'ordre 1}$$

Avec trois points  $(x_{-1}, f_{-1}), (x_0, f_0)$  et  $(x_1, f_1)$ , le polynôme d'interpolation devient une parabole. A partir de cette dernière, on peut évaluer la dérivée première en chacun des trois points

$$f'(x_{k-1}) = \frac{-f(x_{k+1}) + 4f(x_k) - 3f(x_{k-1}))}{2h} + o(h^2) \quad \text{différence avant d'ordre 2}$$

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1}))}{2h} + o(h^2) \quad \text{différence centrée d'ordre 2}$$

$$f'(x_{k+1}) = \frac{3f(x_{k+1}) - 4f(x_k) + f(x_{k-1}))}{2h} + o(h^2) \quad \text{différence arrière d'ordre 2}$$

Pour les différences d'ordre 2, l'erreur varie asymptotiquement comme  $h^2$  alors que pour les différences d'ordre 1, elle varie comme  $h$ . Pour une fonction  $f$  suffisamment lisse et pour un "petit" pas  $h$  donné, la différence d'ordre 2 donnera généralement une erreur plus petite.

Par ailleurs, trois raisons nous poussent à choisir la différence centrée d'ordre 2 :

- D'abord, son terme d'erreur est en  $o(h^2)$  et non en  $o(h)$ ,
- Un calcul plus détaillé montre ensuite que parmi les trois estimateurs d'ordre 2, c'est la différence centrée qui possède en moyenne l'erreur la plus petite,
- Enfin, et c'est là un point crucial : la différence centrée d'ordre 2 ne nécessite que la connaissance de deux points. En effet, la valeur du point en lequel on estime la dérivée n'entre pas en jeu. Le coût en calcul est donc identique à celui d'une différence d'ordre 1, pour un résultat meilleur.

L'estimation de la dérivée seconde est donnée par :

$$f''(x_{k-1}) = f''(x_k) = f''(x_{k+1}) = \frac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1}))}{h^2} + o(h^2)$$

A titre de comparaison, la différence centrée d'ordre 4 s'écrit :

$$f''(x_k) = \frac{-f(x_{k+2}) + 16f(x_{k+1}) - 30f(x_k) + 16f(x_{k-1}) - f(x_{k-2}))}{h^2} + o(h^4)$$

Tous ces résultats sont équivalents aux approximations issues du développement en série de Taylor de la fonction  $f(x)$  et se généralisent au cas multidimensionnel. Les estimées d'ordre un en amont et d'ordre deux centrée du gradient  $\nabla f(x)$  sont respectivement définies comme suit :

$$\begin{aligned} \hat{\nabla}_1 f(x) &= \frac{\partial \hat{f}(x)}{\partial x_i} = \frac{1}{h} [f(x + he_i) - f(x)], \quad \text{pour } i = 1, \dots, n \text{ et } h > 0 \\ \hat{\nabla}_2 f(x) &= \frac{\partial \hat{f}(x)}{\partial x_i} = \frac{1}{2h} [f(x + he_i) - f(x - he_i)], \quad \text{pour } i = 1, \dots, n \text{ et } h > 0 \end{aligned} \quad (5-13)$$

où  $e_i$  est le  $i^{\text{ème}}$  vecteur unitaire.

Ces différentes expressions suggèrent que pour améliorer le résultat du calcul, on peut soit diminuer le pas  $h$  entre les points soit augmenter leur nombre.

On pourrait penser qu'il est a priori intéressant de choisir un pas  $h$  très petit pour augmenter la précision du calcul. Ceci est n'est pas vrai en règle générale. Lorsque  $h$  devient trop petit, des erreurs d'arrondi affectent le calcul et font à nouveau croître l'erreur. En effet, suivant le type de fonction à dériver, il arrivera un moment où l'écart relatif  $(f(x+h) - f(x))/f(x)$  sera inférieur à la précision du calculateur. Le résultat sera alors erroné. Il existe donc une valeur optimale du pas qui dépendra de la fonction  $f(x)$  et de la précision du calculateur.

D'ailleurs, en précision finie, l'erreur commise se compose de deux termes : l'erreur d'approximation et l'erreur d'arrondi. Nous allons analyser précisément comment se combinent ces deux effets. Pour simplifier, nous nous plaçons dans le cas d'une fonction  $f(x)$  scalaire d'une variable réelle  $x$ . Dans ce cas, le développement de Taylor de  $f$  à l'ordre un donne :

$$f'(x) = \frac{f(x+h) - f(x)}{h} + h/2 f''(x) + o(h^2) \quad (5-14)$$

Supposons par ailleurs que  $f$  soit calculée avec une précision relative  $e_f$ . Ce peut être simplement l'erreur d'arrondi, auquel cas  $e_f$  est la précision de l'arithmétique de l'ordinateur (de l'ordre de  $10^{-16}$  en double précision), ou bien une valeur bien plus grande si  $f$  est le résultat d'un calcul complexe. Dans ce cas, ce qui est effectivement calculé est  $\tilde{f} = (1 + e_f) f$ , et la différence entre le quotient calculé et la vraie dérivée vaut, en négligeant l'erreur d'arrondi due à la division :

$$f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} = h/2 f''(x) + 2e_f / hf(x) + o(h^2) \quad (5-15)$$

La somme des deux premiers termes est minimisée par le choix :

$$h = 2 \sqrt{e_f \frac{f(x)}{f''(x)}} \quad (5-16)$$

et l'erreur totale est alors proportionnelle à  $e_f^{\frac{1}{3}}$ . En double précision, cela veut dire que la dérivée aura environ 8 chiffres exacts. Notons que la valeur de la fonction à dériver, et de sa dérivée seconde, influent sur le choix du pas optimal, comme le montre l'équation (5-16). Plus  $f(x)$  sera grand, plus on pourra choisir  $e_f$  grand. De même, plus  $f''(x)$  sera grand, c'est-à-dire plus  $f'(x)$  varie rapidement, plus le choix de  $e_f$  devra être petit. Ainsi, le choix effectif du pas reste délicat, même si  $\sqrt{e_f}$  est une première estimation raisonnable.

Pour le cas d'une différence finie, le pas optimal est donné par

$$h = \sqrt[3]{e_f \frac{3f}{2f^{(3)}}} \tag{5-17}$$

L'erreur totale est alors proportionnelle à  $e_f^{\frac{1}{3}}$ . En double précision, cela veut dire que la dérivée aura environ 12 chiffres exacts (cf. figure 5.9).

L'ouvrage [Den96] contient un algorithme de choix du pas, dans le cas de plusieurs variables, qui prend en compte les divers facteurs d'échelle qui peuvent intervenir.

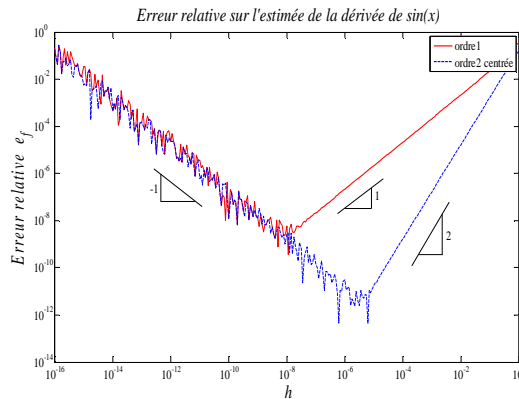


Figure. 5.9 Erreur relative  $e_f = (\hat{f}' - f') / f'$  de la dérivée de  $f(x) = \sin(x)$  à  $x = 0.4$

La figure 5.9 illustre un exemple d'estimation de la dérivée première de la fonction sinus en un point  $x=0.4$ . Nous distinguons facilement deux zones qui correspondent respectivement à des erreurs d'approximation et des erreurs de compensation/troncation avec deux pentes différentes pour la première zone où l'erreur d'approximation dépend de l'ordre d'approximation contrairement à la zone compensation/troncation où les deux courbes possèdent la même pente de variation. En effet, quand on compare deux nombres qui sont presque identiques en utilisant une précision arithmétique finie, leur erreur relative est proportionnelle à l'inverse de la différence entre ces deux nombres. Le pas optimal est de l'ordre de  $10^{-8}$  pour l'estimée du premier ordre et de  $10^{-5}$  pour l'estimation du second ordre centré.

Outre le choix du pas  $h$ , augmenter le nombre de points peut sembler favorable, mais les polynômes d'interpolation de degré supérieur à 3 sont rarement recommandables car leur plus-value en informations est insignifiante devant le nombre d'opérations et le temps de calcul nécessaires.

En général, plus le terme d'erreur  $o(h^p)$  est d'ordre élevé, plus le résultat tendra à être précis. Mais ceci n'est pas toujours vrai lorsque les données sont affectées de bruit. En effet, la différentiation

numérique est une procédure instable qui amplifie fortement le bruit dans un signal. Ainsi, les dérivées d'ordre supérieur et les expressions d'ordre supérieur à 2 sont rarement utilisées.

En supposant que la fonction  $f(x)$  n'est pas connue explicitement et qu'elle ne peut être exactement reconstruite, tous les calculs de  $f(x)$  sont des évaluations de fonctions sujettes à une erreur (une perturbation). Ainsi, pour  $x \in \mathfrak{R}^n$ , nous noterons l'approximation :

$$g(x) = f(x) + b(x) \quad (5-18)$$

où l'erreur  $b(x)$  est assimilée à une variable aléatoire de moyenne nulle  $E[b(x)] = 0$  et de variance  $V[b(x)] = \sigma^2$ .

Par conséquent, chaque composante estimée  $i$  du vecteur gradient peut être caractérisée individuellement par une moyenne  $E[\hat{f}'(x_i)]$ , une variance  $V[\hat{f}'(x_i)]$  et une covariance  $Cov[\hat{f}'(x_i), \hat{f}'(x_{j \neq i})]$ . Cependant, notre objectif est surtout d'évaluer cette estimée pour l'ensemble du vecteur gradient et non pas pour l'un de ses éléments. Une mesure logique de la qualité du gradient estimé est l'espérance de l'erreur quadratique  $E[\|\hat{f}'(x_i) - \nabla f(x)\|^2]$  et sa variance  $V[\|\hat{f}'(x_i) - \nabla f(x)\|^2]$ . Cette dernière peut être toujours décomposée en la somme d'un terme déterministe et un autre stochastique :

$$V[\|\hat{f}'(x_i) - \nabla f(x)\|^2] = \|\text{erreur}_d\|^2 + E[\|\text{erreur}_s\|^2] \quad (5-19)$$

Par exemple, pour l'estimée du gradient par différences finies d'ordre 1 en amont :

$$\text{erreur}_d = \begin{pmatrix} (f(x + he_1) - f(x))/h \\ \vdots \\ (f(x + he_n) - f(x))/h \end{pmatrix} - \nabla f(x), \quad \text{erreur}_s = \begin{pmatrix} (b(x + he_1) - b(x))/h \\ \vdots \\ (b(x + he_n) - b(x))/h \end{pmatrix} \quad (5-20)$$

Le tableau ci-dessous résume l'ensemble des propriétés statistiques des estimées de gradient par différences finies d'ordre 1 en amont et d'ordre 2 centrées :

	Différence avant d'ordre 1	Différence centrée d'ordre 2
Nombre d'évaluations <sup>2</sup>	$n + 1$	$2n$
$E[\hat{f}'(x_i)]$	$f'(x_i)$	$f'(x_i)$
$V[\hat{f}'(x_i)]$	$2\sigma^2 / h^2$	$\sigma^2 / 2h^2$
$Cov[\hat{f}'(x_i), \hat{f}'(x_{j \neq i})]$	$\sigma^2 / h^2$	0
$\ \text{erreur}_d\ ^2 \leq$	$nh^2 D_2^2 / 4$	$nh^4 D_3^2 / 36$
$E(\ \text{erreur}_s\ ^2)$	$2n\sigma^2 / h^2$	$n\sigma^2 / 2h^2$
$h_{e,opt}$	$\sqrt[4]{8\sigma^2 / D_2^2}$	$\sqrt[4]{9\sigma^2 / D_3^2}$
$V[\ \text{erreur}_s\ ^2]$	$n[n(M_4 - \sigma^4) + M_4 + 3\sigma^4] / h^4$	$n[M_4 + \sigma^4] / 8h^4$
$V[\ \text{erreur}_d + \text{erreur}_s\ ^2] \leq$	$n[n(M_4 - \sigma^4) + M_4 + 3\sigma^4] / h^4 + 2n\sigma^2 D_2^2$	$n[M_4 + \sigma^4] / 8h^4 + nh^2 \sigma^2 D_3^2 / 18$
$h_{v,opt}$	-	$\sqrt[4]{9(M_4 + \sigma^4) / 2\sigma^2 D_3^2}$

**Tab. 5.2: Comparaison statistique entre les estimés du gradient par différences finies d'ordre 1 en amont et d'ordre 2 centrée**

<sup>2</sup> Pour la résolution d'un problème d'optimisation où le gradient est estimé par une différence finie de premier ordre, le nombre d'évaluations est réduit à  $n$  dans le décompte global car le calcul de la fonction  $f(x)$  est nécessaire à chaque itération de l'algorithme.

Dans ce tableau,  $D_2$  et  $D_3$  sont respectivement les dérivées maximales d'ordre 2 et 3 de la fonction  $f(x)$ .  $M_4$  est égal à  $E(b(x)^4)$ .

Nous remarquons des résultats de la première partie du tableau 5.2 que les erreurs déterministes sont croissantes en fonction du pas de différentiation  $h$ , alors que les erreurs stochastiques sont décroissantes. Les expressions de l'erreur totale sont convexes en fonctions de  $h$  ce qui permet de calculer le pas optimal  $h_e$  pour chaque estimée.

Il est clair que les valeurs  $\sigma$ ,  $D_2$  et  $D_3$  ne sont pas connus en général. Toutefois, pour un problème pratique, ils peuvent être estimés par un échantillonnage. D'autre part, le pas optimal obtenu  $h_e$  renseigne un ensemble d'indications : il est croissant en fonction de  $\sigma$  et décroissant en fonction de  $D_2$  et  $D_3$  ce qui confirme notre intuition.

De la littérature, il est connu que l'estimée du gradient d'ordre 2 centrée donne une erreur déterministe beaucoup plus petite comparant à celle obtenue par une estimée d'ordre 1. Toutefois, le nombre d'évaluations de la fonction  $f(x)$  est double.

Dans la deuxième partie du tableau, nous avons listé les variances des erreurs stochastiques et totales. Notons que le calcul du pas optimal  $h_{e,opt}$  dans la première partie du tableau, ne prend pas en compte les variances des erreurs. Nous pouvons bien déterminer un pas de différentiation différent en minimisant la précédente erreur plus un certain nombre de fois l'écart-type. Nous pouvons aussi facilement vérifier que le pas optimal sera plus grand. Dans la dernière ligne du tableau 5.2, nous avons calculé le pas optimal  $h_{v,opt}$  qui minimise la variance totale. Ce calcul n'est pas possible pour l'estimée du premier ordre car sa variance est décroissante en fonction de  $h$ . Nous pouvons facilement vérifier que, pour un bruit  $b(x)$  suivant une distribution normale, les pas  $h_{v,opt}$  et  $h_{e,opt}$  sont reliés par la relation  $h_{v,opt} = \sqrt[4]{2}h_{e,opt} \approx 1,1 \cdot h_{e,opt}$  pour  $M_4 = 3\sigma^4$ . Ceci signifie que le pas  $h_{e,opt}$  qui minimise l'erreur totale est toujours très proche du pas optimal réduisant la limite supérieure de la variance de l'erreur.

Il est donc conseillé d'utiliser pour les dérivées d'ordre 1 et 2, les expressions les plus intéressantes qui sont les différences centrées d'ordre 2 :

$$\begin{cases} f'(x) = \frac{f(x+h) - f(x-h)}{2h} + o(h^2) \\ f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + o(h^2) \end{cases} \quad (5-21)$$

En résumé, la méthode de différences finies est surtout utilisée lorsqu'on ne possède pas l'expression ou le code source de la fonction  $f(x)$  mais juste un exécutable jouant le rôle d'une boîte noire. Ceci fait de cette méthode, la technique idéale pour la construction d'une librairie logicielle indépendante des signaux et modèles à traiter. Elle est, en apparence, très simple à mettre en oeuvre, ce qui peut expliquer sa popularité, mais elle n'est pas recommandée systématiquement, puisque non seulement son coût est proportionnel au nombre de paramètres à identifier, et qu'elle donne un résultat approché, avec une précision difficile à évaluer. Dans certaines circonstances, elle peut servir à valider un calcul de gradient d'une autre méthode par comparaison.

### 5.2.2. Méthodes de fonctions de sensibilité

Il s'agit de la méthode la plus naturelle pour calculer le gradient d'un critère  $f$  dans lequel la fonction est définie implicitement par une équation différentielle. Elle consiste à dériver l'équation d'état du



système explicitement par rapport au paramètre  $\theta$ , puis à utiliser la règle dérivation d'une fonction composée. Contrairement à la méthode précédente qui conduit toujours à un résultat approché, cette technique ne comporte aucune approximation dans son principe. Elle est très adaptée à de nombreux problèmes du domaine de l'Automatique car elle ramène le calcul du gradient d'une sortie ou d'un état donné à la résolution d'un système dynamique augmenté.

Supposons pour simplifier que le signal erreur  $e$  soit scalaire, et notons par  $S_e$  sa fonction de sensibilité par rapport aux paramètres  $\theta$

$$s_e(t, \theta) = \frac{\partial e(t, \theta)}{\partial \theta} \quad (5-22)$$

Il est en général possible d'expliciter le gradient du critère en fonction de  $S_e$ . Ainsi par exemple

$$f(\theta) = \frac{1}{2} \sum_{i=1}^{n_t} w_i [e(t_i, \theta)]^2 \Rightarrow \frac{\partial f}{\partial \theta} = \sum_{i=1}^{n_t} w_i s_e(t_i, \theta) e(t_i, \theta) \quad (5-23)$$

De même pour le cas d'un critère non lisse

$$f(\theta) = \sum_{i=1}^{n_t} w_i |e(t_i, \theta)|_+ \Rightarrow \frac{\partial f}{\partial \theta} = \sum_{i=1}^{n_t} \frac{w_i}{2} s_e(t_i, \theta) \left( 1 + \frac{|e(t_i, \theta)|}{e(t_i, \theta)} \right) \quad (5-24)$$

Quand l'erreur est vectorielle, les fonctions de sensibilité de l'erreur interviennent aussi dans l'expression du gradient du critère. Par exemple, pour un critère

$$f(\theta) = \ln \det M(\theta) \quad (5-25)$$

où

$$M(\theta) = \frac{1}{n_t} \sum_{i=1}^{n_t} e(t_i, \theta) e^T(t_i, \theta) \quad (5-26)$$

La  $k^{\text{ième}}$  composante du gradient  $\nabla f(\theta)$  est donnée par

$$\frac{\partial f}{\partial \theta_k} = \frac{2}{n_t} \sum_{i=1}^{n_t} e^T(t_i, \theta) M^{-1}(\theta) \frac{\partial e(t_i, \theta)}{\partial \theta_k} \quad (5-27)$$

Une des démarches envisageable pour calculer le gradient du critère consiste donc à calculer les fonctions de sensibilité de l'erreur par rapport à chacun des paramètres  $\theta_k$  ( $k = 1, \dots, n_\theta$ ).

Dans le cas particulier où  $e(t, \theta)$  est une erreur de sortie

$$e(t, \theta) = y(t) - y_m(t, \theta) \quad (5-28)$$

Cette sensibilité est reliée à celle de la sortie par

$$s_y(t, \theta) = -s_e(t, \theta) \quad (5-29)$$

Dans la suite de cette section, nous allons exposer la méthode à suivre pour évaluer les fonctions de sensibilité efficacement, en même temps que les grandeurs du système traité.

### 5.2.2.1. Fonction de sensibilités pour des systèmes non linéaires

Soit le système non linéaire d'ordre  $n$  décrit par la représentation d'état suivante :

$$\begin{cases} \frac{dx(t)}{dt} = f[x(t), \theta], & x(0) = x_0(\theta) \\ y_m(\theta, t) = h[x(t), \theta] \end{cases} \quad (5-30)$$

Les fonction  $f$  et  $h$  peuvent aussi dépendre des entrées de commande  $u(t)$ , et du temps  $t$ .

Par application de la règle de chaîne (dérivée de fonctions composées), nous définissons la sensibilité de la sortie  $y$  par rapport au paramètre  $\theta_i$  comme étant

$$[S_y(\theta, t)]_i = \frac{\partial y_m(\theta, t)}{\partial \theta_i} = \frac{\partial h[x, \theta]}{\partial x^T} \frac{\partial x}{\partial \theta_i} + \frac{\partial h[x(t), \theta]}{\partial \theta_i} \quad (5-31)$$

Dans cette équation, le terme inconnu est la sensibilité du vecteur d'état  $x(t)$  par rapport au paramètre  $\theta_i$  :  $[S_x(\theta, t)]_i = \partial x(t) / \partial \theta_i$ . Le calcul de cette dernière peut se faire en dérivant l'équation d'état (5-30). On obtient :

$$\frac{d[S_x(\theta, t)]_i}{dt} = \frac{\partial f[x, \theta]}{\partial x^T} [S_x(\theta, t)]_i + \frac{\partial f[x(t), \theta]}{\partial \theta_i}, \quad [S_x(\theta, 0)]_i = \frac{\partial x_0(\theta)}{\partial \theta_i} \quad (5-32)$$

Ainsi, les équations (5-30) et (5-32) construisent un système d'équations différentielles augmenté qui nécessite  $n_\theta + 1$  simulations : une simulation pour le calcul de  $x(t)$  et  $n_\theta$  simulations pour les sensibilités  $[S_x(\theta, t)]_i$ . De façon équivalente, le calcul de tout le vecteur de sensibilité  $S_x(\theta, t)$  se solde alors par la simulation d'un seul grand système d'ordre  $(n \cdot n_\theta + 1)$ .

Les  $n$  équations différentielles augmentées pour chaque paramètre  $\theta_i$  sont linéaires mais non stationnaires parce que le terme  $\partial f[x, \theta] / \partial x^T$  dépend du temps  $t$  et seuls le terme de commande est les conditions initiales dépendent de  $i$ . Ceci fait de cette méthode une technique pas plus compliquée que les différences finies, surtout qu'elle ne pose pas de problème de choix de paramètre pour les différences finies.

Notons que dans le cas où les coefficients  $\theta$  dépendent de d'autres paramètres  $a$  (paramètres de commande par exemple), la sensibilité paramétrique du signal de sortie  $y$  par rapport à ces nouveaux paramètres  $a$  est donnée par :

$$\frac{\partial y}{\partial a_k} = \sum_{i=1}^{n_\theta} [S_y(\theta, t)]_i \frac{\partial \theta_i}{\partial a_k} \quad (5-33)$$

### 5.2.2.2. Sensibilités des systèmes linéaires

Pour le cas particulier des systèmes linéaires décrits par un modèle sous forme d'une équation différentielle d'ordre  $n$ , quand les conditions initiales sont nulles ou d'effet négligeable, les fonctions de sensibilité de la sortie par rapport aux paramètres s'obtiennent très simplement par simulation d'une équation différentielle d'ordre  $2n$ .

En effet, pour un système linéaire invariant décrit par l'équation différentielle

$$\frac{d^n}{dt^n} y + \sum_{i=0}^{n-1} \theta_{n-i} \frac{d^i}{dt^i} y = \sum_{j=0}^m \theta_{n+m+1-j} \frac{d^j}{dt^j} u, \quad \frac{d^{(l)}}{dt^{(l)}} y \Big|_{t=0} = 0 \quad \text{pour } l = 0, \dots, n \quad (5-34)$$

les dérivées partielles par rapport aux coefficients  $\theta$  donnent  $(n+m+1)$  nouvelles équations différentielles. Pour  $k = 1, 2, \dots, n$ , on écrit

$$\frac{d^n}{dt^n} S_y + \sum_{i=0}^{n-1} \theta_{n-i} \frac{d^i}{dt^i} S_y = -\frac{d^{n-k}}{dt^{n-k}} y \quad (5-35)$$

et pour  $k = n+1, \dots, n+m+1$ ,

$$\frac{d^n}{dt^n} S_y + \sum_{i=0}^{n-1} \theta_{n-i} \frac{d^i}{dt^i} S_y = \frac{d^{n+m+1-k}}{dt^{n+m+1-k}} u \quad (5-36)$$

Nous remarquons que ces équations différentielles possèdent le même régime libre ce qui résume le problème en la résolution d'un système beaucoup plus simple composé de  $2n$  équations au lieu des  $(n+m+2)$  initiales.

A titre illustratif, considérons l'exemple suivant dont la transposition au temps discret est triviale [Wal94] :

$$\frac{d^2}{dt^2} y + \theta_1 \frac{d}{dt} y + \theta_2 y = \theta_3 \frac{d}{dt} u + \theta_4, \quad y(0) = 0, \quad \frac{d}{dt} y \Big|_{t=0} = 0 \quad (5-37)$$

Notons  $s_{y_i}$  la sensibilité de la sortie du modèle par rapport à  $\theta_i$ . En dérivant l'équation précédente et les conditions initiales par rapport à  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  et  $\theta_4$  successivement, il vient

$$\begin{cases} \frac{d^2}{dt^2} s_{y_1} + \theta_1 \frac{d}{dt} s_{y_1} + \theta_2 s_{y_1} = -\frac{d}{dt} y, & s_{y_1}(0) = 0, & \frac{d}{dt} s_{y_1} \Big|_{t=0} = 0 \\ \frac{d^2}{dt^2} s_{y_2} + \theta_1 \frac{d}{dt} s_{y_2} + \theta_2 s_{y_2} = -y, & s_{y_2}(0) = 0, & \frac{d}{dt} s_{y_2} \Big|_{t=0} = 0 \\ \frac{d^2}{dt^2} s_{y_3} + \theta_1 \frac{d}{dt} s_{y_3} + \theta_2 s_{y_3} = \frac{d}{dt} u, & s_{y_3}(0) = 0, & \frac{d}{dt} s_{y_3} \Big|_{t=0} = 0 \\ \frac{d^2}{dt^2} s_{y_4} + \theta_1 \frac{d}{dt} s_{y_4} + \theta_2 s_{y_4} = u, & s_{y_4}(0) = 0, & \frac{d}{dt} s_{y_4} \Big|_{t=0} = 0 \end{cases} \quad (5-38)$$

La simulation de  $y$  et des quatre fonctions de sensibilité associées semble donc nécessiter l'utilisation d'un modèle d'ordre 10, puisque chaque équation est du deuxième ordre. En fait, toutes ces équations ont un premier membre identique, et en utilisant les propriétés des modèles linéaires par rapport à l'entrée, on peut simplifier considérablement les calculs et arriver au schéma de la figure 5.10.

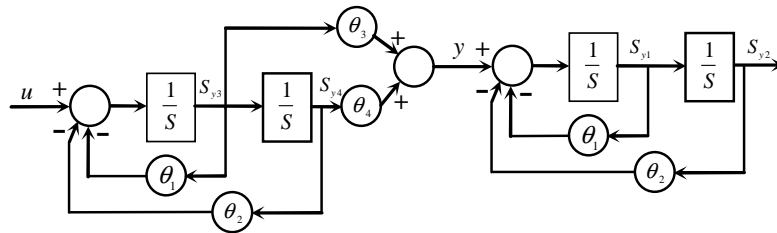


Figure. 5.10 Calcul des fonctions de sensibilité dans le cas d'un modèle linéaire

Les sous-programmes de simulation d'équations différentielles continues demandent en général que celles-ci soient fournies sous forme d'état générique suivante

$$\begin{cases} \frac{d}{dt}x = f(x, \theta, u, t), & x(0) = x_0(\theta) \\ y_e = g(x, \theta, u, t) \end{cases} \quad (5-39)$$

Ici le vecteur des sorties à calculer comprend la sortie et ses fonctions de sensibilité, de sorte que  $y_e = (y, s_{y_1}, s_{y_2}, s_{y_3}, s_{y_4})^T$ . Comme le système est linéaire, les équations d'état et d'observation prennent la forme usuelle

$$\begin{cases} \frac{d}{dt}x = A(\theta)x + B(\theta)u, & x(0) = x_0(\theta) \\ y_e = C(\theta)x \end{cases} \quad (5-40)$$

Les matrices  $A$ ,  $B$  et  $C$  peuvent prendre des formes variées, puisqu'il n'y a pas unicité de la représentation d'état. Convenons de prendre comme variables d'état les sorties des quatre intégrateurs. Il vient

$$A(\theta) = \begin{bmatrix} -\theta_1 & -\theta_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -\theta_3 & -\theta_4 & -\theta_1 & -\theta_2 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad C(\theta) = \begin{bmatrix} \theta_3 & \theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5-41)$$

Pour les systèmes linéaires, stationnaires et monovariables, il est donc possible de simplifier de façon importante le calcul des fonctions de sensibilité. Ces résultats peuvent être même généralisés au cas multivariable [Wil69, Neu71] et au cas linéaire non stationnaire [Neu72].

Notons que si les conditions initiales n'étaient pas nulles ou d'effet négligeable (tout au moins en régime établi), le calcul précédent serait à reprendre, et le degré de simplification atteint serait moindre.

Les gradients sont, finalement, le résultat d'une intégration numérique des systèmes différentiels augmentés. Sauf cas particulier, cette résolution est numérique et elle n'est jamais exacte ce qui perturbe la solution exacte de calcul des gradients. Cependant, si l'on sait obtenir une intégration de bonne qualité des équations différentielles considérées, cette erreur de calcul est beaucoup moins importante que celle issue d'une différence finie à base des estimées numériques des signaux du système.

### 5.2.2.3. Sensibilité paramétrique fréquentielle

Jusqu'à ici, l'approche de calcul des sensibilités paramétriques ne concernait que des critères temporels. Dans le cas des systèmes linéaires, cette approche trouve aussi une application dans le domaine fréquentiel.

En effet, le traitement de l'exemple précédent dans le domaine Laplace produit les résultats suivants :

$$\begin{cases} S_{y_1}(p) = \frac{\partial y}{\partial \theta_1}(p) = \frac{-p(\theta_3 p + \theta_4)}{(p^2 + \theta_1 p + \theta_2)^2} \\ S_{y_2}(p) = \frac{\partial y}{\partial \theta_2}(p) = \frac{-(\theta_3 p + \theta_4)}{(p^2 + \theta_1 p + \theta_2)^2} \\ S_{y_3}(p) = \frac{\partial y}{\partial \theta_3}(p) = \frac{p}{p^2 + \theta_1 p + \theta_2} \\ S_{y_4}(p) = \frac{\partial y}{\partial \theta_4}(p) = \frac{1}{p^2 + \theta_1 p + \theta_2} \end{cases} \quad (5-42)$$

Comme nous pouvons le remarquer, la représentation en fonctions de transferts est équivalente au système (5-38). Toutefois, elle n'est pas minimale ; il existe plusieurs relations simplificatrices entre les différentes sensibilités :  $S_{y_3} = S'_{y_4}$ ,  $S_{y_1} = S'_{y_2}$ ,  $y = S_{y_3} + S_{y_4}$ .

Contrairement à la représentation d'état, cette représentation ne permet pas de réduire efficacement la taille du système global. Elle reste toutefois utile pour évaluer les sensibilités paramétriques des spécifications fréquentielles telle que la pulsation de coupure  $\omega_c$ , les marges de stabilité  $\Delta G$  et  $\Delta \phi$  et les normes infinie des différents transferts.

### Sensibilité (dérivée) de la pulsation de coupure $\omega_c$

Soit  $L = GK$  le transfert de la boucle ouverte et  $\theta$  le vecteur des paramètres du correcteur  $K$ . Afin de calculer la dérivée de la pulsation de croisement  $\omega_c$ , sa définition est utilisée :

$$|L(j\omega_c)| = 1 \quad (5-43)$$

La dérivée totale de ce terme, pouvant être exprimée sous forme de dérivées partielles, est donc nulle :

$$\frac{d|L(j\omega_c)|}{d\theta} = \frac{\partial |L(j\omega_c)|}{\partial \theta} + \frac{\partial |L(j\omega)|}{\partial \omega} \Big|_{\omega_c} \frac{\partial \omega_c}{\partial \theta} = 0 \quad (5-44)$$

Le premier terme de l'équation susmentionnée peut être écrit comme :

$$\frac{\partial |L(j\omega_c)|}{\partial \theta} = |G(j\omega_c)| \frac{\partial |K(j\omega_c)|}{\partial \theta} \quad (5-45)$$

Ainsi,

$$\frac{d\omega_c}{d\theta} = \frac{\partial \omega_c}{\partial \theta} = -|G(j\omega_c)| \frac{\partial |K(j\omega_c)|}{\partial \theta} \left( \frac{\partial |L(j\omega)|}{\partial \omega} \Big|_{\omega_c} \right)^{-1} \quad (5-46)$$

L'équation précédente contient une dérivée de la réponse harmonique de la boucle, qui peut être estimée comme suit, grâce à une relation obtenue à l'aide de l'intégrale de Bode (cf. équation (1-39)) :

$$\frac{\partial |L(j\omega)|}{\partial \omega} \Big|_{\omega_c} \approx \frac{2}{\pi \omega_c} (\Delta \phi - \pi) |L(j\omega_c)| \quad (5-47)$$

En insérant cette dernière équation en (5-46), on obtient finalement :

$$\frac{d\omega_c}{d\theta} = \frac{\partial\omega_c}{\partial\theta} \approx -\frac{\pi\omega_c}{2(\Delta\phi - \pi)} \frac{\partial \ln |K(j\omega_c)|}{\partial\theta} \quad (5-48)$$

La dérivée restante dans cette dernière équation peut être réalisée de manière analytique, dans le cas, fréquent, où la fonction de transfert du régulateur est connue. Ainsi, en ne mesurant que la marge de phase et la pulsation de croisement, la dérivée de cette dernière peut être estimée facilement et rapidement.

### Sensibilité (dérivée) de la marge de phase $\Delta\phi$

Selon sa définition, la marge de phase est une fonction des paramètres du régulateur ainsi que de la pulsation critique, alors même que cette dernière est une fonction de  $\theta$  :

$$\Delta\phi = \Delta\phi(\theta, \omega_c(\theta)) \quad (5-49)$$

Ainsi, la dérivée de la marge de phase peut être exprimée à l'aide des dérivées partielles :

$$\frac{d\Delta\phi}{d\theta} = \frac{\partial\Delta\phi}{\partial\theta} + \left. \frac{\partial\Delta\phi}{\partial\omega_c} \right|_{\omega_c} \frac{\partial\omega_c}{\partial\theta} \quad (5-50)$$

En remplaçant  $\Delta\phi$  par  $\pi + \angle L(j\omega_c)$  on obtient :

$$\frac{d\Delta\phi}{d\theta} = \frac{\partial\angle L(j\omega_c)}{\partial\theta} + \left. \frac{\partial\angle L(j\omega)}{\partial\omega} \right|_{\omega_c} \frac{\partial\omega_c}{\partial\theta} \quad (5-51)$$

Le premier terme de l'équation susmentionnée peut être écrit comme suit :

$$\frac{\partial\angle L(j\omega_c)}{\partial\theta} = \frac{\partial\angle K(j\omega_c)}{\partial\theta} \quad (5-52)$$

De plus :

$$\left. \frac{\partial\angle L(j\omega)}{\partial\theta} \right|_{\omega_c} = \left. \frac{\partial\angle K(j\omega)}{\partial\omega} \right|_{\omega_c} + \left. \frac{\partial\angle G(j\omega)}{\partial\omega} \right|_{\omega_c} \quad (5-53)$$

Finalement, la dérivée de la marge de phase s'écrit :

$$\frac{d\Delta\phi}{d\theta} = \frac{\partial\angle K(j\omega_c)}{\partial\theta} + \left( \left. \frac{\partial\angle K(j\omega)}{\partial\omega} \right|_{\omega_c} + \left. \frac{\partial\angle G(j\omega)}{\partial\omega} \right|_{\omega_c} \right) \frac{\partial\omega_c}{\partial\theta} \quad (5-54)$$

À nouveau, toutes les dérivées relatives à la réponse harmonique du correcteur peuvent être effectuées de manière analytique. La dérivée de l'argument du système  $G$  peut, quant à elle, être estimée grâce l'intégrale de Bode

$$\left. \frac{d\angle G(j\omega)}{d\omega} \right|_{\omega_c} \approx \frac{1}{\omega_c} \angle G(j\omega_c) + \frac{2}{\pi\omega_c} [\log |G(0)| - \log |G(j\omega_c)|] \quad (5-55)$$

Ainsi, toutes les expressions relatives au calcul de la dérivée de la marge de phase peuvent à nouveau être évaluées simplement et rapidement.

**Sensibilité (dérivée) de la marge de gain  $\Delta G$** 

La dérivée de  $K_{-\pi}$  qui est définie comme étant l'inverse de la marge de gain  $\Delta G$ , peut être exprimée de la manière suivante :

$$\frac{dK_{-\pi}}{d\theta} = \frac{\partial |L(j\omega_{-\pi})|}{\partial \theta} + \frac{\partial |L(j\omega)|}{\partial \omega} \Big|_{\omega_c} \frac{\partial \omega_c}{\partial \theta} \quad (5-56)$$

Les deux premiers termes contenant des dérivées partielles peuvent s'écrire sous la forme :

$$\begin{aligned} \frac{\partial |L(j\omega)|}{\partial \theta} &= |G(j\omega_{-\pi})| \frac{\partial |K(j\omega_{-\pi})|}{\partial \theta} \\ \frac{\partial |L(j\omega)|}{\partial \omega} \Big|_{\omega_c} &= |G(j\omega_{-\pi})| \frac{\partial |K(j\omega)|}{\partial \omega} \Big|_{\omega_c} + |K(j\omega_{-\pi})| \frac{\partial |G(j\omega)|}{\partial \omega} \Big|_{\omega_c} \end{aligned} \quad (5-57)$$

À nouveau, toutes les dérivées relatives au correcteur peuvent être calculées de manière analytique. Le terme restant peut être estimé grâce à la relation découlant de l'intégrale de Bode :

$$\frac{\partial |G(j\omega)|}{\partial \omega} \Big|_{\omega_c} \approx \frac{2}{\pi\omega_{-\pi}} \angle G(j\omega_{-\pi}) |G(j\omega_{-\pi})| \quad (5-58)$$

La valeur de la dérivée  $\partial \omega_c / \partial \rho$  est établie en utilisant la définition de la pulsation critique  $\omega_{-\pi}$  :

$$\angle L(j\omega_{-\pi}) \equiv -\pi \quad (5-59)$$

Ainsi :

$$\frac{d\angle L(j\omega_{-\pi})}{d\theta} = \frac{\partial \angle L(j\omega_{-\pi})}{\partial \theta} + \frac{\partial \angle L(j\omega)}{\partial \omega} \Big|_{\omega_c} \frac{\partial \omega_c}{\partial \theta} = 0 \quad (5-60)$$

Le premier terme est simplifié de la sorte :

$$\frac{\partial \angle L(j\omega_{-\pi})}{\partial \theta} = \frac{\partial \angle K(j\omega_{-\pi})}{\partial \theta} \quad (5-61)$$

et le second est reformulé comme suit :

$$\frac{\partial \angle L(j\omega)}{\partial \omega} \Big|_{\omega_c} = \frac{\partial \angle K(j\omega)}{\partial \omega} \Big|_{\omega_c} + \frac{\partial \angle G(j\omega)}{\partial \omega} \Big|_{\omega_c} \quad (5-62)$$

Le terme  $\partial \omega_c / \partial \rho$  est donc finalement donné par :

$$\frac{\partial \omega_c}{\partial \theta} = - \left( \frac{\partial \angle K(j\omega)}{\partial \omega} \Big|_{\omega_c} + \frac{\partial \angle G(j\omega)}{\partial \omega} \Big|_{\omega_c} \right)^{-1} \frac{\partial \angle K(j\omega_{-\pi})}{\partial \theta} \quad (5-63)$$

Une fois encore, la seule dérivée ne pouvant être effectuée de manière analytique peut être approximée grâce à l'intégrale de Bode.

$$\frac{d\angle G(j\omega)}{d\omega} \Big|_{\omega_c} \approx \frac{1}{\omega_{-\pi}} \angle G(j\omega_{-\pi}) + \frac{2}{\pi\omega_{-\pi}} [\log |G(0)| - \log |G(j\omega_{-\pi})|] \quad (5-64)$$

### Dérivée de la norme infinie de la fonction de sensibilité

La dérivée de la marge de module  $\Delta M$  qui n'est autre que l'inverse de la norme infinie de la fonction de sensibilité peut s'écrire comme suit :

$$\frac{d\Delta M}{d\theta} = \frac{\partial \Delta M}{\partial \theta} + \left. \frac{\partial \Delta M}{\partial \omega} \right|_{\omega_0} \frac{\partial \omega_0}{\partial \theta} \quad (5-65)$$

où, pour rappel,  $\omega_0$  est la pulsation associée à la marge.

Le premier terme de l'équation peut être calculé analytiquement et vaut :

$$\frac{\partial \Delta M}{\partial \theta} = \frac{\partial |1 + L(j\omega_0)|}{\partial \theta} \quad (5-66)$$

Le deuxième terme est formulé de la façon suivante :

$$\left. \frac{\partial \Delta M}{\partial \omega} \right|_{\omega_0} = \left. \frac{\partial |1 + L(j\omega)|}{\partial \omega} \right|_{\omega_0} \quad (5-67)$$

Comme la fonction  $|1 + L(j\omega)|$  ainsi que sa première dérivée sont continues, un résultat évident, mais très intéressant, en découle : la dérivée décrite ci-après est nulle, car elle est effectuée à l'endroit qui correspond à l'extremum de la fonction :

$$\left. \frac{\partial \Delta M}{\partial \omega} \right|_{\omega_0} = 0 \quad (5-68)$$

Par conséquent :

$$\frac{d\Delta M}{d\theta} = \frac{\partial |1 + L(j\omega_0)|}{\partial \theta} \quad (5-69)$$

Comme toutes les dérivées relatives aux paramètres du régulateur peuvent être obtenues analytiquement, celle de l'équation (5-69) peut l'être de manière exacte. Ainsi, par la seule mesure du paramètre de synthèse  $\omega_0$  de la pulsation associée, ainsi que du point correspondant de la réponse harmonique du système à commander, la dérivée de l'équation (5-69) peut être calculée, et cela sans recourir aux dérivées de  $G(j\omega)$ .

### 5.2.3. État adjoint

Pour les systèmes décrits par des équations d'état linéaires ou non, les techniques d'état adjoint inspirées de la commande optimale permettent de simplifier considérablement les calculs.

Considérons un modèle d'état à temps discret éventuellement non linéaire décrit par :

$$\begin{cases} x(t+1) = f(x(t), \theta), & x(0) = x_0(\theta) \\ y(t, \theta) = h(x(t), \theta) \end{cases} \quad (5-70)$$

Ce modèle peut bien sûr dépendre aussi d'une entrée  $u$  et du temps  $t$ , mais cette dépendance est ici omise pour alléger les notations. Supposons que le critère utilisé peut s'exprimer comme suit :



$$f(\theta) = \sum_{t=0}^{n_t} r_t(x(t), \theta) \quad (5-71)$$

Ce qui le cas pour un très grand nombre des critères considérés jusqu'ici. Par exemple, pour un critère quadratique sur l'erreur de sortie, on aura

$$\begin{aligned} r_t(x(t), \theta) &= [y(t) - y(t, \theta)]^T Q(t) [y(t) - y(t, \theta)] \\ &= [y(t) - h(x(t), \theta)]^T Q(t) [y(t) - h(x(t), \theta)] \end{aligned} \quad (5-72)$$

Transformons le critère additif en critère terminal par une introduction d'une variable d'état supplémentaire

$$x^0(t+1) = x^0(t) + r_t(x(t), \theta) \quad \text{avec} \quad x^0(0) = 0 \quad (5-73)$$

de sorte que

$$f(\theta) = x^0(n_t + 1) \quad (5-74)$$

Définissons un vecteur d'état étendu par

$$x_e(t) = \begin{bmatrix} x^0(t) \\ x(t) \\ \theta \end{bmatrix} \quad (5-75)$$

Il vérifie

$$x_e(0) = \begin{bmatrix} 0 \\ x_0(\theta) \\ \theta \end{bmatrix} \quad (5-76)$$

et

$$x_e(t) = \begin{bmatrix} x^0(t) + r_t(x(t), \theta) \\ f(x(t), \theta) \\ \theta \end{bmatrix} \quad (5-77)$$

Le critère s'écrit maintenant

$$f(\theta) = (1, 0, \dots, 0) \cdot x_e(n_t + 1) \quad (5-78)$$

Ce qui entraîne

$$\frac{\partial f}{\partial \theta} = \frac{\partial x_e^T(n_t + 1)}{\partial \theta} \frac{\partial f}{\partial x_e(n_t + 1)} \quad (5-79)$$

avec

$$\frac{\partial f}{\partial x_e(n_t + 1)} = (1, 0, \dots, 0)^T \quad (5-80)$$

Par ailleurs, puisque  $x_e^T(t+1) = f_e^T(x_e(t))$ , on peut écrire, en utilisant la règle de dérivation des fonctions composées,

$$\left\{ \begin{array}{l} \frac{\partial x_e^T(n_i+1)}{\partial \theta} = \frac{\partial x_e^T(n_i)}{\partial \theta} \frac{\partial f_e^T(x_e(n_i))}{\partial x_e(n_i)} \\ \frac{\partial x_e^T(n_i)}{\partial \theta} = \frac{\partial x_e^T(n_i-1)}{\partial \theta} \frac{\partial f_e^T(x_e(n_i-1))}{\partial x_e(n_i-1)} \\ \dots \\ \frac{\partial x_e^T(1)}{\partial \theta} = \frac{\partial x_e^T(0)}{\partial \theta} \frac{\partial f_e^T(x_e(0))}{\partial x_e(0)} \end{array} \right. \quad (5-81)$$

On a donc, globalement,

$$\frac{\partial f}{\partial \theta} = \frac{\partial x_e^T(0)}{\partial \theta} \cdot \frac{\partial f_e^T(x_e(0))}{\partial x_e(0)} \cdot \frac{\partial f_e^T(x_e(1))}{\partial x_e(1)} \cdot \dots \cdot \frac{\partial f_e^T(x_e(n_i-1))}{\partial x_e(n_i-1)} \cdot \frac{\partial f_e^T(x_e(n_i))}{\partial x_e(n_i)} \cdot \frac{\partial f}{\partial x_e(n_i+1)} \quad (5-82)$$

Ce calcul de dérivations en chaîne peut donc être organisé en commençant par les termes situés au temps final et en remontant progressivement vers le temps initial. A cette fin, définissons le vecteur *état adjoint* à l'état étendu, noté  $d_x$ , qui vérifie la condition terminale.

$$d_x(n_i+1) = \frac{\partial f}{\partial x_e(n_i+1)} = (1, 0, \dots, 0)^T \quad (5-83)$$

La propagation des calculs de droite à gauche par l'équation de récurrence (à temps rétrogradé)

$$d_x(t-1) = \frac{\partial f_e^T(x_e(t-1))}{\partial x_e(t-1)} d_x(t), \quad \text{pour } t = n_i+1, \dots, 1 \quad (5-84)$$

qui permet le calcul de  $d_x(0)$ . Le gradient du critère est donné alors par

$$\frac{\partial f}{\partial \theta} = \frac{\partial x_e^T(0)}{\partial \theta} d_x(0) \quad (5-85)$$

où

$$\frac{\partial x_e^T(0)}{\partial \theta} = \left[ 0 \quad \frac{\partial x_0^T(0)}{\partial \theta} \quad I_{n_v} \right] \quad (5-86)$$

En temps direct, l'équation d'évolution de l'état adjoint devient

$$d_x(t+1) = \left[ \frac{\partial f_e^T(x_e(t))}{\partial x_e(t)} \right]^{-1} d_x(t) \quad (5-87)$$

Tandis que l'équation d'état linéarisée au voisinage de la trajectoire nominale s'écrit

$$\delta x_e(t+1) = \frac{\partial f_e^T(x_e(t))}{\partial x_e(t)} \delta x_e(t) \quad (5-88)$$

Le produit scalaire de l'état étendu linéarisé par l'état adjoint reste donc constant le long de la trajectoire (propriété de dualité)

$$\delta x_e^T(t+1)d_{x_e}(t+1) = \delta x_e^T(t)d_{x_e}(t) \quad (5-89)$$

Cette dualité peut être utilisée pour vérifier les calculs.

La mise en œuvre de la méthode peut s'effectuer sans définition explicite d'un état étendu. On a en effet

$$\frac{\partial f_e^T(x_e(t))}{\partial x_e(t)} = \begin{bmatrix} 1 & 0^T & 0^T \\ \frac{\partial r_t(x(t), \theta)}{\partial x(t)} & \frac{\partial f^T(x(t), \theta)}{\partial x(t)} & 0 \\ \frac{\partial r_t(x(t), \theta)}{\partial \theta} & \frac{\partial f^T(x(t), \theta)}{\partial \theta} & I_{n_\theta} \end{bmatrix} \quad (5-90)$$

La première composante de  $d_{x_e}$  vérifie donc

$$d_{x_e}(t-1) = d_{x_e}(t), \quad \text{avec} \quad d_{x_e}(n_t+1) = 1 \quad (5-91)$$

Les composantes associées à  $x$  vérifient

$$d_x(t-1) = \frac{\partial f^T(x(t), \theta)}{\partial x(t)} d_x(t) + \frac{\partial r_t(x(t), \theta)}{\partial x(t)}, \quad \text{avec} \quad d_x(n_t+1) = 0 \quad (5-92)$$

Les composantes associées à  $\theta$  satisfont

$$d_\theta(t-1) = d_\theta(t) + \frac{\partial f^T(x(t), \theta)}{\partial \theta} d_x(t) + \frac{\partial r_t(x(t), \theta)}{\partial \theta}, \quad \text{avec} \quad d_\theta(n_t+1) = 0 \quad (5-93)$$

Le gradient du critère vis-à-vis des paramètres est finalement donné par

$$\frac{\partial f}{\partial \theta} = \frac{\partial x_e^T(\theta)}{\partial \theta} d_{x_e}(0) + d_\theta(0) \quad (5-94)$$

En résumé, le calcul du gradient du critère s'effectue en deux phases. On simule d'abord le modèle (à temps croissant) pour la valeur de  $\theta$  à laquelle le gradient doit être évalué et l'on en déduit la suite des  $x_e(t)$ . C'est la *phase progressive*. On calcule ensuite l'état adjoint par simulation (à temps décroissant). C'est la *phase rétrograde*, qui permet le calcul de  $d_{x_e}(0)$  puis du gradient du critère.

L'intérêt essentiel de cette approche est de permettre le calcul du gradient en deux simulations (sans aucune approximation) quelle que soit la dimension du vecteur des paramètres et que le modèle d'état soit linéaire ou pas.

Cette idée peut être transposée aux modèles à temps continu. La simulation de ceux-ci fait en général intervenir des approximations, et d'autres approximations auront lieu lors de la simulation de l'état adjoint. Or le calcul de gradient par état adjoint est très sensible aux erreurs en apparence minimes pouvant fausser le résultat final. Il est donc souhaitable de s'assurer, en utilisant un modèle discrétisé,

que les équations utilisées pour la simulation dans la phase progressive subissent les mêmes approximations que celles utilisées pour la simulation dans la phase rétrograde.

#### 5.2.4. Différentiation automatique et code adjoint

Une autre technique de calcul de gradient est de faire appel à la différentiation explicite des équations, ce qui exige un travail considérable d'analyse et de codage. Or, il se trouve que les logiciels de différentiation automatique ont gagné en performance et fiabilité, révélant le fort potentiel de cette technique.

La différentiation automatique est le procédé qui, à partir d'un code informatique évaluant une fonction  $f$ , produit un code qui évalue les valeurs exactes (aux erreurs d'arrondi près) des dérivées partielles de  $f$ . Cette technique utilisée en mode adjoint présente l'avantage de calculer les dérivées de la fonctionnelle à minimiser en un temps raisonnable par rapport au nombre de paramètres par rapport auxquels on dérive. De plus, les dérivées calculées ne posent pas de problèmes de consistance avec les équations d'état résolues, contrairement à celles issues des autres techniques. Enfin, la différentiation automatique permet à l'utilisateur de supprimer facilement n'importe quelle partie du gradient, et donc d'analyser les différentes contributions des parties du code de simulation. Dans le même esprit, la prise en compte des modifications du code à différencier pour le code adjoint est plus aisée.

Ce type de technique s'applique donc à tout critère  $f$  calculable par un programme informatique, à condition bien sûr que  $f$  ainsi calculée soit dérivable par rapport aux paramètres au point considéré. La suite des instructions de ce programme forme ce que nous appellerons le *code direct*, qui calcule  $f(\theta)$  à partir de variables indépendantes  $(\theta, y, \dots)$ , en utilisant éventuellement des variables intermédiaires dont les valeurs dépendent de celles prises par les variables indépendantes.

En représentant chaque ligne du code par une fonction  $v_i$  élémentaire, le code n'est ni plus ni moins une composition de  $n$  fonctions élémentaires ( $n$  étant le nombre de lignes du code). La différentiation automatique consiste à appliquer les formules de dérivation des fonctions composées à chacune de ces fonctions élémentaires soit à chaque ligne du code. On peut appliquer les formules de composition de deux façons différentes qu'on appelle mode de différentiation automatique :

- le mode linéaire direct ou linéaire tangent (différentiation directe),
- le mode linéaire inverse ou cotangent (différentiation adjointe).

##### 5.2.4.1. Mode direct

La différentiation directe est le mode le plus simple et le plus intuitif. Le code est différencié classiquement ligne par ligne. A chaque variable du code  $v_i$ , il est associé une variable dérivée  $\dot{v}_i$  qui contient la dérivée directionnelle de  $v_i$  dans une direction donnée de  $\mathfrak{R}^n$ . C'est le bon mode pour calculer la dérivée d'un grand nombre de variables de sortie par rapport à un petit nombre de variables d'entrée.

Les dérivées obtenues ont l'avantage d'être plus précises que celles obtenues par différences finies. Si  $n_\theta$  représente le nombre de paramètres de décision du problème, le coût de calcul du gradient par la méthode des différences finies est de  $n_\theta$  calculs de  $f$  ; dans la méthode de différentiation directe, ce

même calcul requiert  $4 \cdot n_\theta$  évaluations [Mor85]. La différentiation directe est donc trop lourde lorsque le nombre de paramètres dépasse une dizaine car le coût de la fonction et le nombre de paramètres rendent illusoire l'application de ce mode.

### 5.2.4.2. Mode inverse

Le mode inverse de différentiation est moins intuitif et plus complexe. Ce mode s'effectue en dualisant ligne par ligne le code originel, mais dans l'ordre inverse à celui de l'exécution du code initial. Il s'agit du bon mode pour calculer le gradient d'une application de  $\mathfrak{R}^{n_\theta}$  dans  $\mathfrak{R}^m$ , ou le gradient de  $a^T f$  lorsque  $f$  est une fonction de  $\mathfrak{R}^{n_\theta}$  dans  $\mathfrak{R}^m$ . A chaque variable  $v_i$  il est associé une variable duale  $d_i$  qui contient la variation de  $a^T f$ ,  $a$  donnée de  $\mathfrak{R}^m$ , par rapport à une perturbation de  $v_i$ . On comprend ainsi que si  $v_i$  est une variable égale au paramètre d'entrée mais qui n'intervient pas dans le calcul de la variable de sortie, alors  $\dot{v}_i = 1$  et  $d_i = 0$ .

Ce mode possède les mêmes avantages (rapidité, précision), en plus d'une relative simplicité de mise en oeuvre, que les méthodes standard basées sur la résolution de l'adjoint. Avec ce mode, le coût d'évaluation théorique de l'adjoint est de 5 fois le coût du calcul de la fonction [Mor85], et ce indépendamment du nombre de paramètres par rapport auxquels on dérive. Cependant, comme lors de la résolution de l'adjoint, on peut rencontrer des problèmes de taille mémoire.

L'approche par code adjoint est particulièrement intéressante quand la dimension de  $\theta$  est grande, une situation où les approches par différences finies et par fonctions de sensibilité se traduisent par des calculs très lourds. L'idée de base (voir par exemple [Gil91, Gri91]) est très proche de celle des techniques d'état adjoint dont nous venons d'exposer. Comme ces dernières, les techniques de code adjoint alternent calculs en *sens direct* et en *sens inverse*.

### 5.2.4.3. Principes de différentiation automatique par code adjoint

La technique que nous allons présenter ne conduit pas, en général, au résultat le plus concis possible mais est simple et construit un code adjoint de façon systématique. Les instructions inutiles générées par cette approche pourront être éliminées par utilisation d'un compilateur optimisant le code.

Soit  $v$  le vecteur formé par toutes les variables intervenant dans le programme de calcul du critère  $f(\theta)$ . Toutes instruction d'affectation exécutée par le code direct vient modifier la valeur d'une des composantes de  $v$ , ce qui peut s'écrire

$$v_{\mu(k)} := \phi_k(\{v_i / i \in I_k\}) \quad (5-95)$$

où  $\mu(k)$  est l'indice de la composante de  $v$  modifiée par la  $k^{\text{ième}}$  instruction d'affectation exécutée et  $I_k$  est l'ensemble des indices des composantes de  $v$  dont dépend  $\phi_k$ .

L'idée [Spe80] consiste à interpréter cette instruction comme une transformation  $\phi_k$  de toutes les variables  $v$  du code, qui laisse toutes les composantes de  $v$  inchangées sauf  $v_{\mu(k)}$  :

$$(\phi_k(v))_i = \begin{cases} v_i & \forall i \neq \mu(k) \\ \phi_k(\{v_i / i \in I_k\}) & i = \mu(k) \end{cases} \quad (5-96)$$

Indexons  $v$  pour distinguer ses valeurs successives,  $v(0)$  désignant l'état initial de ces variables et  $v(\kappa)$  leur valeur finale. Le code direct peut être alors considéré comme l'équation d'évolution d'un système dynamique décrit par l'équation d'état suivante :

$$v(k) = \phi_k(v(k-1)), \quad \text{pour } k = 1, \dots, \kappa \quad (5-97)$$

Notons que l'instruction d'affectation associée à l'indice  $k$  dépendra de l'ordre dans lequel ces instructions seront exécutées. La présence dans le code direct d'instructions de branchement demandera donc un traitement particulier que nous examinerons plus loin.

Convenons de placer en tête de  $v$  les  $n$  variables indépendantes (en commençant par les composantes  $\theta$ ) puis les variables dépendantes, la dernière étant destinée à recevoir à l'issue de l'exécution du code direct la valeur du critère à calculer. Les  $n$  premières composantes de l'état initial  $v(0)$  sont donc égales aux valeurs des variables indépendantes pour lesquelles le critère doit être évalué. Toutes les autres composantes de  $v(0)$  peuvent être prises égales à zéro, car c'est l'exécution du code direct qui définira les valeurs à donner aux variables dépendantes. Une fois cette exécution terminée, la valeur du critère sera dans la dernière composante de  $v$ , soit

$$f(\theta) = (0, 0, \dots, 0, 1)v(\kappa) \quad (5-98)$$

La situation (critère terminal, équation d'état) est donc analogue à celle rencontrée lors de l'utilisation de l'état adjoint, et va recevoir le même traitement. Le gradient du critère par rapport à  $\theta$  peut s'écrire, en appliquant la règle de dérivation en chaîne.

L'état adjoint  $d$  associé à  $v$  sera donc initialisé par la condition terminale

$$d(\kappa) = \frac{\partial f}{\partial v(\kappa)} = (0, 0, \dots, 0, 1)^T \quad (5-99)$$

et calculé par récurrence rétrograde suivant la formule

$$d(k-1) = \frac{\partial \phi_k^T}{\partial v(k-1)} d(k), \quad \text{pour } k = \kappa, \dots, 1 \quad (5-100)$$

où  $\partial \phi_k^T / \partial v(k-1)$  est une matrice identité dont la  $\mu(k)$ <sup>ème</sup> colonne a été remplacée par  $\partial \phi^T / \partial v(k-1)$ . Compte tenu de cette structure particulière, la récurrence peut encore s'écrire

$$d_i(k-1) = \begin{cases} d_i(k) + \frac{\partial \phi_k}{\partial v_i(k-1)} d_{\mu(k)}(k) & \text{si } i \neq \mu(k) \\ \frac{\partial \phi_k}{\partial v_i(k-1)} d_{\mu(k)}(k) & \text{si } i = \mu(k) \end{cases} \quad (5-101)$$

Elle permet de calculer la valeur initiale  $d(0)$  de l'état adjoint. Comme

$$\frac{\partial f}{\partial \theta} = \frac{\partial v^T(0)}{\partial \theta} d(0) \quad \text{avec} \quad \frac{\partial v^T(0)}{\partial \theta} = [I_{n_\theta} \quad 0] \quad (5-102)$$

Les  $n_\theta$  premières composantes de  $d(0)$  contiennent la valeur de  $\nabla f(\theta)$ . Notons que les  $n - n_\theta$  composantes suivantes de  $d(0)$  contiennent les valeurs des dérivées partielles du critère par

rapport aux autres variables indépendantes, qui sont donc disponibles sans aucun calcul supplémentaire. Il n'est pas nécessaire de mémoriser toutes les valeurs prises par  $d(k)$  quand  $k$  varie de  $\kappa$  à 0, puisque seules les  $n$  premières composantes de  $d(0)$  nous intéressent.

L'instruction d'affectation  $v_{\mu(k)} := \phi_k(\{v_i / i \in I_k\})$  pourra donc se traduire par les instructions adjointes suivantes, dans cet ordre :

$$\left\| \begin{array}{l} \text{Pour tout } i \in I_k \setminus \{\mu(k)\} \text{ Faire} \\ \quad d_i := d_i + \frac{\partial \phi_k}{\partial v_i} d_{\mu(k)} \\ \text{FinPour} \\ d_{\mu(k)} := \frac{\partial \phi_k}{\partial v_{\mu(k)}} d_{\mu(k)} \end{array} \right.$$

Dans le cas d'une initialisation de la variable directe  $v_{\mu(k)}$ ,  $\phi_k$  ne dépend pas de  $v_{\mu(k)}$ , et la dernière de ces équations devient  $d_{\mu(k)} := 0$ .

Dans le cas d'une augmentation de  $v_{\mu(k)}$ , de la forme

$$v_{\mu(k)} := v_{\mu(k)} + \psi_k(v) \tag{5-103}$$

avec  $\psi_k$  indépendant de  $v_{\mu(k)}$ , elle devient  $d_{\mu(k)} := d_{\mu(k)}$ .

Afin d'enrichir tout ce développement, nous présentons un exemple de code adjoint pour une seule instruction. Supposons que la  $k^{\text{ième}}$  instruction d'affectation exécutée soit

$$f := f + [y(t) - y_m(t)]^2 \tag{5-104}$$

Cette instruction modifie la variable  $f$  en lui affectant la valeur de la fonction

$$\phi_k(f, y(t), y_m(t)) = f + [y(t) - y_m(t)]^2 \tag{5-105}$$

Notons  $df$ ,  $dy(t)$  et  $dy_m(t)$  les variables duales respectivement associées à  $f$ ,  $y(t)$  et  $y_m(t)$ . L'application des formules précédentes (5-101) se traduit alors par :

$$\left\| \begin{array}{l} dy(t) := dy(t) + \frac{\partial \phi_k}{\partial y(t)} df \\ dy_m(t) := dy_m(t) + \frac{\partial \phi_k}{\partial y_m(t)} df \\ df := \frac{\partial \phi_k}{\partial f} df \end{array} \right.$$

soit encore,

$$\left\| \begin{array}{l} dy(t) := dy(t) + 2[y(t) - y_m(t)]df \\ dy_m(t) := dy_m(t) + 2[y(t) - y_m(t)]df \\ df := df \end{array} \right.$$

La dernière de ces instructions duales étant bien entendu superflue. ■

Par ailleurs, le code direct contient le plus souvent des instructions de branchement, qui doivent être, elles aussi, dualisées. La dualisation d'une boucle répétitive (do, for, while...) se traduira par une autre boucle répétitive dans laquelle les instructions seront dualisées en sens inverse de leur exécution dans le code direct. En cas de branchement conditionnels, il faudra donc tenir compte des branches effectivement parcourues lors de l'exécution du code direct, de sorte que la dualisation de

```

|| Si (condition C) Alors
||   code A
|| Sinon
||   code B
|| Finsi
    
```

se traduira par

```

|| Si (condition C) Alors
||   code adjoint A
|| Sinon
||   code adjoint B
|| Finsi
    
```

Dans la méthode par état adjoint, on calculait l'évolution de l'état à temps direct avant de calculer l'évolution de l'état adjoint à temps rétrograde et d'en déduire le vecteur gradient. Ici, la situation est analogue : on commence par exécuter les instructions du code direct avant d'exécuter celles du code adjoint dans l'ordre inverse. Dans les deux cas, il est nécessaire de stocker les variables directes intervenant dans les expressions non linéaires pour permettre l'exécution des calculs adjoints.

La démarche à suivre peut être résumée ainsi :

- Définir  $v$ , en mettant les variables associées à  $\theta$  en tête et la variable associée à  $f(\theta)$  en queue ;
- Associer une variable adjointe à chaque composante de  $v$  ;
- Initialiser toutes les variables adjointes à zéro, sauf la dernière (associée au critère), qui est initialisée à 1 ;
- Dualiser les instructions dans l'ordre inverse de leur exécution (ce qui suppose d'inverser les sens des boucles) ;
- Exécuter le code direct en mémorisant les valeurs prises par les variables directes qui interviennent dans des expressions non linéaires et les branchements conditionnels effectués ;
- Exécuter le code adjoint ;
- Récupérer le gradient dans les  $n$  premières composantes de  $d$ . Les  $n_\theta$  premières composantes correspondent au gradient vis-à-vis des autres variables indépendantes.

Il convient d'être très minutieux et prudent, car des erreurs en apparence mineures peuvent rendre le code adjoint inutilisable. Les règles suivantes sont à recommander [Tal91] :

- Développer le code adjoint à partir du code direct (et non à partir de l'adjoint mathématique des équations mathématiques directes) ;
- A chaque sous-programme du code direct doit correspondre un sous-programme adjoint ;



- Donner aux variables, étiquettes et sous-programmes adjoints des noms clairement reliés à ceux des variables, étiquettes et sous-programmes directs ;
- Ne jamais modifier un code direct sans récupérer les changements sur son code adjoint.

Soit  $\delta v(k)$  l'état linéarisé au voisinage de la trajectoire nominale. Son produit scalaire avec l'état adjoint reste constant le long de la trajectoire :  $d^T(k+1)\delta v(k+1) = d^T(k)\delta v(k)$ . Comme dans la méthode des états adjoints, cette propriété de dualité peut être exploitée pour vérifier la validité du code adjoint, à condition de construire un code direct linéarisé pour générer la suite des  $\delta v$ .

Les techniques de code adjoint peuvent être étendues au calcul des dérivées d'ordre supérieur (calcul du Hessien d'un critère par exemple). Pour plus de détails, voir [Gil91].

De ce fait, la différentiation automatique paraît donc pouvoir fournir avec le mode adjoint un outil précis, rapide et simple de mise en œuvre, d'obtention du gradient d'une fonctionnelle dépendant d'un grand nombre de paramètres. En contrepartie, comme c'est le cas lors de la résolution de l'équation adjointe, on a besoin pour le calcul du gradient de sauvegarder toutes les parties de la trajectoire intervenant non linéairement dans le code direct ce qui représente grosso modo tous les états intermédiaires par lequel passe le système durant la simulation. Ceci a pour effet de rendre la méthode a priori inexploitable lors du traitement de grands systèmes industriels complexes.

Le thème de différentiation automatique a motivé plusieurs travaux de recherche qui se poursuivent. Au jour d'aujourd'hui, nous comptons plusieurs dizaines d'outils générateurs du code adjoint pour plusieurs langages scientifiques. A titre d'exemple, nous citons [Aut08] les outils ADC, ADIC et FAD pour le langage C/C++, les outils ADF, ADIFOR, COSY INFINITY et ODYSSEE pour le langage Fortran, les outils ADMAT/ADMIT, TOMLAB/MAD et DIFFEDGE pour Matlab et les outils SCIAD et DIFFCODE pour Scilab.

Même si la majorité de ces outils réalisent une partie absolument considérable du travail en fournissant toutes les lignes de codes nécessaires au calcul de l'adjoint, un travail non négligeable de compréhension et de traitement reste à effectuer sur ces codes issus de la différentiation automatique. Le plus souvent possible, il est impératif à procéder à l'initialisation des variables duales dans la routine de tête et éliminer celles correspondant à des variables constantes au cours de la simulation. Ensuite, il est toujours utile de programmer des formules plus simples et économiques pour s'affranchir de la sauvegarde des valeurs de passage de ces algorithmes itératifs à chaque itération.

Enfin, la partie du post-traitement du code construit par l'outil de différentiation la plus longue mais la plus importante est l'élimination des sauvegardes super opérées par les outils de différentiations qui rendent parfois le code brut inutilisable. En utilisant l'invariance de certaines variables au cours de la simulation, la linéarité des calculs et les propriétés de la plupart des boucles, il est possible de réduire considérablement le nombre des variables à sauvegarder.

La complexité de cette procédure de post traitement est l'une des principales raisons qui nous a poussé à abandonner cette technique qui devient très lourde en terme de mise en œuvre pour les cas traités ici.

### 5.2.5. Dérivation complexe

Dans cette partie, nous présentons une méthode de calcul du gradient d'une fonction réelle à base du calcul complexe. L'utilisation de ce type de calcul pour estimer des dérivées est originaire des travaux de J. N. Lyness et C. B. Moler [Lyn67a, Lyn67b] où plusieurs méthodes de calcul complexe sont exposées. Parmi ces méthodes, nous nous intéressons tout particulièrement à une technique assez générique permettant de calculer la  $n^{\text{ème}}$  dérivée d'une fonction analytique. Cette technique de calcul n'a été reprise et exploitée que récemment par W. Squire et G. Trapp dans [Squ98] et J. Martins et al. dans [Mar00, Mar01]. Elle présente une propriété numérique très intéressante.

#### 5.2.5.1. Principe de la méthode

Commençons tout d'abord par rappeler la notion de fonction *holomorphe sur un domaine* qui constitue le pilier central de l'étude de l'analyse complexe ; ce sont les fonctions à valeurs complexes, définies et dérivables en tout point d'un sous-ensemble ouvert du plan complexe  $X$ .

Cette notion est beaucoup plus forte que la dérivabilité réelle. Elle implique (via la théorie de Cauchy) que la fonction est analytique : elle est indéfiniment dérivable et est égale au voisinage de tout point de l'ouvert à la somme de sa série de Taylor. Un fait remarquable en découle : les notions de fonction analytique complexe et de fonction holomorphe coïncident.

Nous allons voir que l'estimée de la dérivée première d'une fonction réelle peut être simplement exprimée via une formule de calcul complexe.

Considérons une fonction  $f(z) = u + iv$  d'une variable complexe  $z = x + iy$ . Si  $f$  est analytique, elle vérifie donc les équations de Cauchy-Riemann ;

$$\begin{cases} \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \end{cases} \quad (5-106)$$

Ces équations aux dérivées partielles établissent la relation exacte entre la partie réelle  $u$  et la partie imaginaire  $v$  de la fonction  $f$ . Nous pouvons ainsi utiliser la définition de la dérivée (au sens complexe) de la partie droite de la première équation de Cauchy-Riemann pour obtenir,

$$\frac{\partial u}{\partial x} = \lim_{h \rightarrow 0} \frac{v(x + i(y + h)) - v(x + iy)}{h} \quad (5-107)$$

où  $h$  est un "petit" réel.

Les fonctions dont nous voulons calculer les dérivées sont réelles à variables réelles donc  $y = 0$ ,  $v(x) = 0$  et  $f(x) = u(x)$ . L'équation (5-107) devient

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{v(x + ih)}{h} = \lim_{h \rightarrow 0} \frac{\text{Im}[f(x + ih)]}{h} \quad (5-108)$$

Pour un petit pas  $h$ , cette dérivée peut être approximée par :

$$\frac{\partial f}{\partial x} \approx \frac{\text{Im}[f(x + i h)]}{h} \tag{5-109}$$

Cette approximation peut aussi être obtenue plus facilement à partir du développement en série de Taylor. En effet, pour une fonction  $f$  analytique réelle à variables réelles, le développement en série de Taylor autour d'un point  $x$  avec une perturbation imaginaire  $i h$  donne :

$$f(x + i h) = f(x) + i h f'(x) - h^2 \frac{f''(x)}{2!} - i h^3 \frac{f^{(3)}(x)}{3!} + o(h) \tag{5-110}$$

En prenant la partie imaginaire de part et d'autre, nous obtenons :

$$f'(x) = \frac{\text{Im}[f(x + i h)]}{h} + h^2 \frac{f^{(3)}(x)}{3!} + o(h^2). \tag{5-111}$$

Désormais, comme pour une différence finie de second ordre centrée, l'approximation est une estimée  $o(h^2)$  de la dérivée de la fonction  $f$ . En outre, cette estimée ne possède pas une erreur d'arrondi comme dans le cas des différences finies où cette erreur est produite à cause de la soustraction des termes du numérateur. Ceci constitue un avantage énorme par rapport à ces approches qui souffrent du problème de choix du pas  $h$ .

Il est possible de dériver plusieurs formules de calcul complexe où les erreurs d'arrondi sont inexistantes.

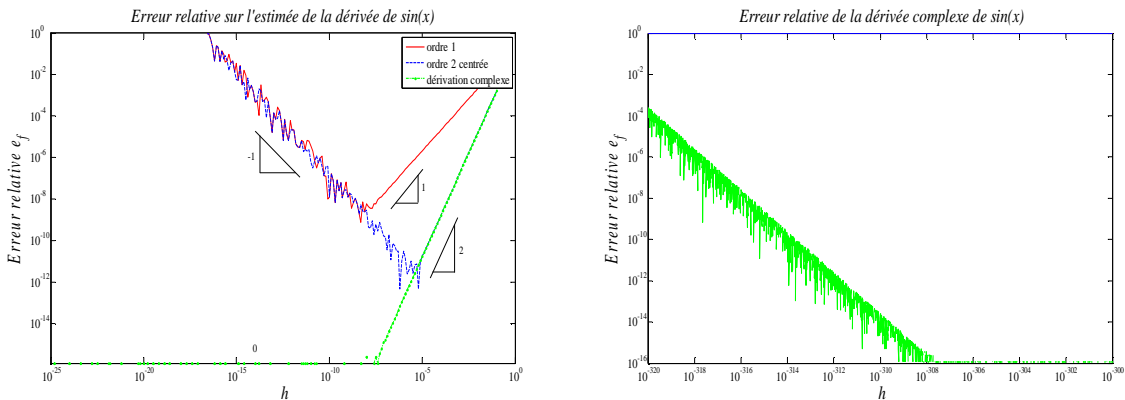


Figure 5.11 Erreur relative  $e_f = (\hat{f}' - f') / f'$  de la dérivée de  $f(x) = \sin(x)$  à  $x = 0.4$

La figure 5.11 illustre cette intéressante propriété avec le même exemple d'estimation de la dérivée de la fonction sinus. Nous remarquons que pour des pas supérieurs à  $10^{-5}$ , l'erreur de la méthode de dérivation complexe colle parfaitement avec celle de la différence finie d'ordre deux centrée ; il s'agit de la même erreur de troncation. Par contre, l'erreur relative de la méthode de dérivation complexe continue sa décroissance en fonction du pas  $h$  est atteinte une valeur numérique inférieure à la tolérance du compilateur (spacing of floating point numbers  $\approx 10^{-16}$ ).

Comme nous pouvons le constater, la taille du pas complexe peut être rendue extrêmement petite. Cependant, en utilisant une précision finie, il existe une limite inférieure à ce pas. La gamme des nombres réels qui peuvent être manipulés dans le calcul numérique dépend particulièrement du compilateur employé. Habituellement, quand des nombres complexes à double précision sont employés, le plus petit nombre différent de zéro qui peut être représenté est  $10^{-308}$  (Smallest positive

floating point number). Si un nombre est au-dessous de cette valeur, il est considéré comme nulle. Notons aussi que l'évaluation est encore très précise pour un pas de  $10^{-307}$ . Au-dessous de cette limite, l'estimée de la dérivée est noyée dans le bruit numérique et les résultats d'évaluation résultants sont indéfinies.

Le deuxième avantage de l'approximation (5-109) réside dans le fait qu'elle ne nécessite que  $n$  évaluations pour l'estimation d'un gradient d'une fonction  $f : \mathbb{R} \rightarrow \mathbb{R}^n$  avec une précision supérieure à celle de l'estimée du gradient par différences finies de second ordre centrée qui nécessite 2 fois plus d'évaluations. En terme de temps de calcul, les langages scientifiques traitent aujourd'hui les opérations sur des variables réelles comme un cas particulier du calcul complexe, les temps d'évaluation des fonctions étant les mêmes. Globalement cela revient à un temps de calcul deux fois moins important dans le cas d'une dérivation complexe.

En comparant la précision relative des calculs complexe et réel, l'analyse prouve qu'en utilisant des nombres complexes, il y a une erreur accrue au niveau des opérations arithmétiques de base, plus spécifiquement pour les opérations de division et de multiplication.

Ainsi, la dérivation complexe permet de construire une estimée assez précise et qui ne pose pas de problème du choix du pas. Il suffit de choisir un pas inférieur à  $10^{-8}$  pour obtenir une estimation de la dérivée de très bonne qualité.

### 5.2.5.2. Approximation des dérivées d'ordre supérieur

D'une manière générale, la dérivée de l'ordre  $n$  d'une fonction analytique donnée peut être calculée par l'Intégrale de Cauchy sous sa forme générale :

$$f^{(n)}(z) = \frac{n!}{2\pi i} \oint_{\Gamma} \frac{f(\xi)}{(\xi - z)^{n+1}} d\xi \tag{5-112}$$

où  $\Gamma$  est le contour orientée fermé contenant  $z$ . Cette intégrale peut être numériquement calculée au moyen d'une approximation trapézoïdale autour d'un cercle de rayon  $r$ ,

$$f^{(n)}(z) \approx \frac{n!}{mr} \sum_{j=0}^{m-1} \frac{f\left(z + r e^{i\frac{2\pi j}{m}}\right)}{e^{i\frac{2\pi j n}{m}}} \tag{5-113}$$

où si  $m$  est le nombre de points utilisés dans l'intégration, nous pouvons estimer les dérivées d'ordre  $n = 0, 1, \dots, m-1$ .

En comparant les méthodes conventionnelles de différences finies et l'intégration complexe exprimée dans l'équation (5-113), nous observons que les deux formules utilisent des sommations de type  $a_0 f(x_0) + \dots + a_n f(x_n)$  où les coefficients  $a_i$  ont signes différents. Cependant, il y a une différence significative entre les deux. Dans des méthodes conventionnelles le pas  $h$  doit être diminuée afin de réduire l'erreur de troncation de l'estimée, le rendant susceptible à l'erreur d'arrondi. Si nous voulons réduire toute l'erreur de troncation d'intégration de la méthode complexe, tout ce que nous devons faire est d'augmenter le nombre d'évaluations de fonction  $m$  dans l'équation (5-113). Ceci garde l'erreur d'arrondi constante et il est alors possible de calculer une limite sur l'erreur impliquée dans l'approximation.

La formule de dérivation complexe de la première dérivée (5-109) peut être trouvée à partir de l'équation (5-113). En effet, pour une fonction analytique réelle de variable réelle,

$$f(x + iy) = u + iv \Rightarrow f(x - iy) = u - iv \quad (5-114)$$

Pour le cas  $m = 2$  dans l'équation (5-113) et en sommant, nous obtenons l'approximation de second ordre que nous avons déterminée précédemment. C'est la seule approximation qui peut être obtenue à partir de l'équation (5-113) et qui n'implique pas la soustraction et elle est seulement valide pour les fonctions dont la partie imaginaire est nulle sur l'axe des réelles.

D'autres part, à partir du développement limité de la fonction analytique réelle  $f(x)$  à droite et à gauche du point  $x$ , nous pouvons déduire une approximation de la dérivée seconde de  $f(x)$

$$f''(x) = \frac{f(x) - \text{Re}[f(x + ih)]}{h^2/2} + \frac{h^2}{12} f^{(4)}(x) + o(h^2) \quad (5-115)$$

Nous pouvons facilement vérifier que cette formule implique une soustraction et donc des erreurs d'arrondi quand le pas est trop petit.

Par ailleurs, en utilisant comme paramètre de perturbation le terme  $Ih = \sqrt{2}/2(i+1)h$ , le développement en série de Taylor permet de déduire une nouvelle approximation :

$$f''(x) = \frac{\text{Im}[f(x + Ih) - f(x - Ih)]}{h^2} + \frac{h^4}{360} f^{(6)}(x) + o(h^4) \quad (5-116)$$

Cette approximation est toujours sujette à des erreurs d'arrondi, mais son erreur de troncation est  $h^4/360 f^{(6)}(x)$  tandis que l'erreur associée à l'équation (5-115) est  $h^2/12 f^{(4)}(x)$ . Nous pouvons montrer également par la simulation que l'approximation (5-116) est moins sensible aux erreurs d'approximation que (5-115).

L'obtention de la première et la deuxième dérivée en utilisant les équations (5-109) et (5-116) exige des évaluations des fonction  $f(x + ih)$ ,  $f(x + Ih)$  et  $f(x - Ih)$ . Toutefois, il est possible d'obtenir une approximation de première dérivée qui implique les évaluations  $f(x + Ih)$  et  $f(x - Ih)$  et qui ne génère aucune erreur d'arrondi. Cette dernière est donnée par :

$$f'(x) \approx \frac{\text{Im}[f(x + Ih) - f(x - Ih)]}{\sqrt{2} h} \quad (5-117)$$

Ces estimations peuvent être exploitées pour définir le gradient et la matrice Hessienne qui amorce l'algorithme AGU. Rappelons que ce dernier est basé, dans sa première phase, sur une descente de type quasi-Newton qui requiert une bonne estimation du Hessien.

Notons finalement que si les dérivées premières sont calculées analytiquement et mises en application dans un programme, la méthode de dérivation complexe peut alors être employée pour calculer les dérivées secondes avec d'autant plus de précision.

### 5.2.5.3. Procédure d'implémentation (fonctions complexes de base et opérateurs en Matlab)

Dans l'estimation de la dérivée d'une fonction  $f$  via la dérivation complexe (5-109), nous avons supposé que  $f$  était une fonction analytique, c'est-à-dire que les équations de Cauchy-Riemann

s’appliquent. Il est donc important d’examiner dans quelle mesure cette supposition est vérifiée quand la valeur de la fonction est calculée par un algorithme numérique. En outre, il est également utile d’expliquer comment nous pouvons convertir les fonctions et les opérateurs réels pour qu’ils puissent prendre des nombres complexes comme arguments. Actuellement, dans le cas de Matlab et d’autres langages scientifiques tels que C, C++ et Fortran, les nombres complexes sont un type de données standard et plusieurs fonctions mathématiques de base sont déjà définies pour les traiter.

Tout algorithme peut être décomposé en une série d’opérations de base. En convertissant un algorithme réel en un “algorithme complexe”, nous distinguons deux principaux types d’opérations :

- Les opérations relationnelles
- Les opérateurs et fonctions arithmétiques

Sur Matlab, les opérateurs logiques tels que “>” et “<” sont déjà définis pour des nombres complexes ; la comparaison se fait sur la partie réelle des arguments. Ces opérateurs sont souvent utilisés en même temps que des opérateurs de branchement ou d’orientation tel que les fonctions “if” et “switch” et “while” afin d’orienter le fil d’exécution du programme. Dans les autres langages tel que Fortran où les opérateurs logiques ne sont pas adaptés au calcul complexe, l’algorithme original et sa version “complexifiée” doivent bien évidemment suivre le même fil d’exécution. Par conséquent, il est impératif de redéfinir ces opérateurs pour comparer seulement les parties réelles des arguments.

Les fonctions sélectives qui retournent un argument tel que le maximum et le minimum sont basées sur les opérateurs logiques. Par conséquent, selon notre discussion précédente, nous devrions une fois de plus choisir un nombre basé sur sa partie réelle seulement. Sur Matlab ceci se traduit par les modifications suivantes :

$$\min(x_1 + i y_1, x_2 + i y_2) = \begin{cases} x_1 + i y_1 & \text{si } x_1 \leq x_2 \\ x_2 + i y_2 & \text{si } x_1 > x_2 \end{cases} \quad (5-118)$$

$$\max(x_1 + i y_1, x_2 + i y_2) = \begin{cases} x_1 + i y_1 & \text{si } x_1 \geq x_2 \\ x_2 + i y_2 & \text{si } x_1 < x_2 \end{cases} \quad (5-119)$$

En réalité, ces deux nouvelles fonctions sont toujours non analytiques en  $x_1 = x_2$  puisque aucune dérivée n’existe pour les deux fonctions “min” et “max” réelles en ce point. Toutefois, les états  $x_1 < x_2$  pour la fonction “min” et  $x_1 > x_2$  pour la fonction “max” seront substitués respectivement par  $x_1 \leq x_2$  et  $x_1 \geq x_2$  de sorte que nous obtenions non seulement une valeur de fonction pour  $x_1 = x_2$ , mais également nous pouvons calculer pour ce point les dérivées gauche et droite de ces deux fonctions respectivement.

Tout algorithme employant des conditions est susceptible d’être une fonction discontinue de ses entrées. Soit la valeur de la fonction elle-même est discontinue, soit la discontinuité est au niveau des dérivées. En utilisant une méthode de différences finies de premier ordre, l’estimée de la dérivée sera incorrecte si les deux évaluations de la fonction sont à une distance  $h$  de la discontinuité. Cependant, si la dérivation complexe est employée, l’évaluation dérivée résultante sera correcte jusqu’à la discontinuité. À la discontinuité, la dérivée n’existe pas par définition, mais si la fonction est définie en ce point, l’approximation retourne une valeur qui dépendra de la façon dont la fonction est définie en ce point.

Pour rappeler seulement quelques unes, les fonctions et les opérateurs arithmétiques incluent l'addition, la multiplication, et les fonctions trigonométriques. La plupart de ces derniers ont une définition complexe standard qui est analytique presque partout. Toutes ces définitions sont mises en application en Matlab. Elles en dépendent du compilateur et des bibliothèques de base. Pour d'autres langages, l'utilisateur doit vérifier la documentation du compilateur employé afin de déterminer quelles fonctions doivent être redéfinies.

Les fonctions de la variable complexe sont simplement des prolongements de leurs analogues réelles. En exigeant que la fonction prolongée satisfait les équations de Cauchy-Riemann, c'est-à-dire l'analyticité, et que ses propriétés soient identiques que ceux de la fonction réelle, nous pouvons obtenir une unique définition de la fonction complexe. Puisque ces fonctions complexes sont analytiques, l'approximation par dérivation complexe est valide et donnera le bon résultat.

Certaines des fonctions analytiques possèdent des singularités où des branches de courbes sur lesquelles elles ne sont pas analytiques. Ceci ne pose pas de problème si, comme précédemment mentionné, l'approximation par dérivation complexe renvoie une des dérivées correctes autour.

Quant au cas d'une fonction qui n'est pas définie en un point donné, l'algorithme ne renverra pas une valeur de fonction, ainsi la dérivée ne peut pas être obtenue. Cependant, l'évaluation de la dérivée sera correcte dans le voisinage de la discontinuité.

Une fonction non analytique commune à tous les langages est la fonction de valeur absolue ou module. Quand l'argument de cette fonction est une valeur complexe, la fonction renvoie un nombre réel positif,  $|z| = (x^2 + y^2)^{\frac{1}{2}}$ . La définition de cette fonction n'a pas été établie par imposition de l'analyticité et donc elle ne retournera pas la dérivée correcte lorsqu'elle est utilisée dans une estimation de dérivées par dérivation complexe. Afin d'établir une définition adaptée de cette fonction, nous commençons par satisfaire les équations de Cauchy-Riemann [Mar00]. De l'équation (5-106), comme nous connaissons ce que doit être la valeur de dérivée, nous pouvons écrire,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = \begin{cases} -1 & \text{si } x < 0 \\ +1 & \text{si } x > 0 \end{cases} \quad (5-120)$$

De l'équation (5-106), comme  $\partial v / \partial x = 0$  sur l'axe des réelles, nous obtenons  $\partial u / \partial y = 0$ , c'est-à-dire, la partie réelle du résultat doit être indépendante de la partie imaginaire de la variable. Par conséquent, le nouveau signe de la partie imaginaire dépend seulement du signe de la partie réelle du nombre complexe, et une fonction analytique "de valeur absolue" peut être définie comme :

$$abs(x + i y) = |x + i y| = \begin{cases} -x - i y & \text{si } x < 0 \\ x + i y & \text{si } x > 0 \end{cases} \quad (5-121)$$

Notons que cette fonction n'est pas analytique à  $x = 0$  puisque aucune dérivée n'existe pour la valeur absolue réelle. De nouveau, l'approximation par dérivation complexe donnera la valeur correcte de la dérivée première autour de la discontinuité.

Pour les calculs faits dans le cadre de l'optimisation par sous-gradient, nous pouvons effectuer la modification suivante : remplacer la condition  $x > 0$  par  $x \geq 0$  de sorte que nous obtenions non seulement une valeur de fonction pour  $x = 0$ , mais également nous pouvons calculer correctement la

dérivée droite en ce point. Ceci simplifie la gestion des cas d'exceptions sans pour autant changer les estimations des sous-gradients.

Il existe d'autres fonctions non analytiques spécifiques à Matlab :

- Les fonctions "min" et "max" qu'on peut leurs définir les fonctions analytiques équivalentes suivantes :

$$\min(z_1, z_2) = \min(x_1 + i y_1, x_2 + i y_2) = \begin{cases} z_1 & \text{si } x_1 \leq x_2 \\ z_2 & \text{si } x_1 > x_2 \end{cases} \quad (5-122)$$

$$\max(z_1, z_2) = \max(x_1 + i y_1, x_2 + i y_2) = \begin{cases} z_1 & \text{si } x_1 \geq x_2 \\ z_2 & \text{si } x_1 < x_2 \end{cases} \quad (5-123)$$

- L'opérateur transposé représenté par une apostrophe (') pose un problème car il prend le conjugué complexe des éléments de la matrice (non analytique), ainsi on doit employer à la place l'opérateur transposé non conjugué représenté par "point apostrophe" (.').

In fine, pour appliquer la méthode de dérivation complexe, on doit avoir accès au code source qui calcule la valeur de la fonction  $f$ . Le procédé d'exécution peut être récapitulé comme suit :

- Remplacement de toutes variables déclarées en type réel avec des déclarations en type complexe. Il n'est pas strictement nécessaire de déclarer toutes les variables en type complexe, mais il est beaucoup plus facile de faire ainsi. Cette étape est ignorée dans le cas d'un langage non typé comme Matlab.
- Définition de toutes les fonctions et les opérateurs qui ne sont pas définis pour des arguments complexes. Cette définition doit respecter les conditions d'analyticité presque partout.
- Les dérivées peuvent alors être estimées en utilisant l'équation (5-109).

Rappelons que cette méthode n'est valable que pour des fonctions réelles à variables réelles. En outre, l'utilisation de ce type d'approximation pour l'estimation de la dérivée d'une norme  $H_\infty$  ou d'un abscisse spectrale conduit à des résultats erronés. En effet, même si ces fonctions sont réelles leurs évaluations fait appel à un calcul de valeurs propres intermédiaires qui peuvent être imaginaires. L'apparition de tel calcul dans lors de l'exécution de la fonction fausse le calcul de la partie imaginaire de la fonction qui ne préserve plus son indépendance par rapport au reste du calcul intermédiaire de la fonction.

### 5.3. Conclusion

Dans ce chapitre, différents aspects d'analyse numérique ont été abordés dans le but de proposer une estimation efficace du  $\varepsilon$ -sous-différentiel  $\bar{\partial}_\varepsilon f(x)$  ne nécessitant pas le recours à des techniques de calcul très lourdes.

Dans un premier temps, une analyse de l'ensemble approximant le  $\varepsilon$ -sous-différentiel a été présentée. Alors que le calcul exact de cet ensemble est très difficile dans le cas général de critères implicites, mais en pratique il suffit d'un nombre fini de gradients échantillonnés pour construire un sous-



ensemble convexe fermé à partir duquel une bonne direction de descente et une bonne mesure de stationnarité sont déterminées.

Ainsi, au voisinage des zones non différentiables, il suffit de choisir un seul gradient de part et d'autre de ces zones pour construire une bonne approximation du  $\varepsilon$ -sous-différentiel. Ceci est réalisé dans notre cas par un échantillonnage uniforme à l'intérieur de l'hypersphère de rayon  $\varepsilon$  et de centre  $x$ . L'expérience montre aussi qu'il est souvent suffisant de tirer un nombre de gradients  $m$  égal au double de la dimension de l'espace paramétrique ( $m = 2 \cdot n$ ). En outre, l'analyse de deux types de fonctions non différentiables montre que l'opérateur estimé du  $\varepsilon$ -sous-différentiel  $G_\varepsilon(x)$  joue le rôle d'une bonne fonction de lissage.

Comme l'estimation du  $\varepsilon$ -sous-différentiel dépend du calcul des gradients échantillonnés, la deuxième partie de ce chapitre a été dédiée aux méthodes de différentiation et d'estimation du gradient. La liste des techniques existantes est longue. Nous nous sommes concentrés sur les classes adaptées aux problèmes de cahiers des charges en Automatique qui font intervenir des objets définis implicitement : solution d'une équation polynomiale ou d'une équation différentielle.

En particulier, nous nous sommes intéressés aux différences finies, au calcul des sensibilités, à la différentiation automatique à base du code adjoint et à la dérivation complexe. Nous avons analysé ces méthodes en termes de leur exactitude, leur facilité d'exécution, leur capacité de calculer des dérivées d'ordre supérieur et leur efficacité en ce qui concerne la charge de calcul. Nous avons trouvé que :

- Les estimateurs par différences finies sont le plus faciles à mettre en œuvre et à utiliser mais ils sont imprécis et inefficaces également pour des problèmes impliquant un grand nombre de variables.
- Les méthodes de fonctions de sensibilité temporelles et fréquentielles sont très efficaces. Leur calcul est théoriquement exact mais il nécessite, en pratique, soit une intégration soit une différentiation numérique. L'inconvénient majeur de ce type de méthodes est leur mise en place initiale qui est assez lourde.
- Les méthodes de l'état et du code adjoint permettent de produire du calcul exact pour n'importe quelle dérivée. Cependant, leur mise en œuvre est très lourde et fait appel à des routines spécifiques qui ne sont pas, au jour d'aujourd'hui, tout à fait efficaces. Il y a donc un choix stratégique à faire lors d'un projet pour justifier l'investissement dans une telle méthode. Pour des problèmes d'optimisation de taille moyenne, la méthode de dérivation complexe est fortement recommandée. Nous l'avons employée intensivement sur Matlab et nous l'avons trouvée très efficace et simple à mettre en œuvre pour répondre à la majorité de nos exigences pour des problèmes d'optimisation de cahiers des charges. Toutefois, cette méthode ne garantit pas une erreur d'arrondi nulle lorsque le calcul de la fonction réelle passe par un calcul intermédiaire complexe. Dans ce sens, il serait très intéressant d'étendre l'applicabilité de cette méthode afin qu'elle puisse être directement intégrée dans les compilateurs.

# Chapitre 6

## Analyse et validation de cahiers des charges pour des problèmes de commande

Dans ce chapitre, nous présentons les résultats d'application, de l'approche par optimisation non linéaire non différentiable développée dans ce manuscrit. Ces applications porteront sur des problèmes variés de commande et de retouche de correcteurs (linéaires et non linéaires).

Le but est de confirmer l'applicabilité de cette approche et son efficacité à travers un large panel de problèmes de commande et de cahiers des charges représentatifs et motivants l'application de l'optimisation non linéaire non différentiable en Automatique.

Nous avons pris le pari de présenter ici des problèmes de typiques, facilement appréhendables et qui pourront facilement servir de base à d'autres applications. Ces exemples suffisent pour illustrer la nécessité de l'approche de formulation non convexe présentée dans le chapitre 2 et les méthodes de résolutions et de calcul des chapitres 4 et 5.

Pour chaque application, nous présentons la problématique, le cahier des charges, le type de formulation, la méthode d'optimisation, les outils de calcul, les solutions et les analyses des résultats obtenus.

### 6.1. Problèmes de stabilisation linéaire

Peu après le développement de la théorie de la commande robuste dans les années 80, il a été réalisé que la plupart des problèmes d'Automatique d'analyse et de synthèse sont difficiles à résoudre en pratique et quelques uns sont même prouvés complexes au sens formel. Nous renvoyons le lecteur vers l'excellente expertise [Blo00] pour la compréhension des définitions et des classifications des problèmes difficiles, au sens de la complexité de résolution, en Automatique.

En particulier, le problème de synthétiser un contrôleur stabilisant un système linéaire est en général difficile quand l'ordre du contrôleur est fixé pour être strictement inférieur à celui du système. La synthèse d'une commande par retour de sortie statique est le cas particulier d'ordre 0 de ce problème. En dépit de plusieurs années de recherches, aucun algorithme de complexité polynomiale ou heuristique efficace pour la résolution du problème de commande par retour de sortie statique n'a pas été établi.

Une des sources de complexité de ce problème est sa non convexité. Mathématiquement, la stabilité est équivalente à localisation des valeurs propres d'une matrice (ou équivalentement les racines d'un polynôme) dans le demi-plan gauche. Quand elle est formulée dans l'espace Euclidien des matrices réelles (ou polynômes à coefficients réels), cette contrainte de stabilité n'est pas convexe. Il s'en suit que l'ensemble des contrôleurs stabilisants est typiquement un espace non convexe et parfois même non connexe, c'est-à-dire disjoint dans l'espace des paramètres [Ack02].

Une difficulté supplémentaire du problème de stabilisation est la non différentiabilité de ses critères associés. En effet, l'abscisse spectrale d'une matrice réelle non symétrique est non différentiable aux points de l'espace des matrices où plus d'une valeur propre réelle ou une paire conjuguée atteignent la partie réelle maximale, et est non-Lipschitz si la valeur propre qui achève la partie réelle maximale est multiple. Cependant, l'abscisse spectrale est différentiable aussi longtemps que la partie réelle maximale est achevée par une valeur propre simple (réelle ou paire conjuguée), et dans ce cas-ci son gradient est facilement calculé au moyen des vecteurs propres gauches et droits associés. Toutefois, la « rencontre » de la non différentiabilité se produit presque toujours lors de l'optimisation de l'abscisse spectrale d'un système paramétré [Bur03].

### 6.1.1. Problème de stabilisation pure

Souvent, la première étape importante d'une synthèse d'une loi de commande est de trouver un correcteur  $K(p)$  stabilisant le système en boucle fermée. Nous proposons dans ce qui suit deux formulations de ce problème qui peuvent être résolus efficacement avec les algorithmes non différentiables déjà développés.

Les applications présentées ci-dessous couvrent le cas de correcteurs  $K$  stabilisant statiques, mais il est clair que des problèmes de stabilisation comprenant des contraintes structurelles  $K \in \mathcal{K}$  peuvent bien être traités exactement de la même manière.

Dans ce qui suit, nous présentons brièvement deux formulations qui permettent de traiter ce problème efficacement.

La première formulation est basée sur la notion de la norme  $H_\infty$ . En effet, il est connu que sous les conditions de commandabilité et d'observabilité, un système linéaire à temps invariant est stable au sens de Lyapunov si et seulement si sa norme  $H_\infty$  est finie [Des75]. Plus particulièrement, sous l'hypothèse de commandabilité et d'observabilité, la loi de commande statique  $u = K y$  stabilise le système

$$G(p): \begin{cases} \dot{x} = Ax + B_2 u \\ y = C_2 x \end{cases} \quad (6-1)$$

si et seulement si la matrice de transfert en boucle fermée  $C_2 [pI - (A + B_2 K C_2)]^{-1} B_2$  possède une norme  $H_\infty$  finie.

En introduisant la généralisation de la norme  $H_\infty$  donnée par [Boy91], on définit la norme  $H_\infty$  décalée de  $a$  ( $a$ -shifted  $H_\infty$  norm) comme suit :

$$\|H\|_{a,\infty} = \sup_{\text{Re}(p) > -a} |H(p)| = \|H_a\|_\infty = \|H(p-a)\|_\infty \quad (6-2)$$

Cette norme généralisée est finie si et seulement si les parties réelles des pôles du transfert  $H$  sont inférieurs à  $-a$ .

Ainsi, pour  $a < 0$ , la norme la norme  $H_\infty$  décalée de  $a$  peut mesurer l'instabilité des fonctions de transferts. Par exemple :  $\|1/(p-1)\|_{-2,\infty} = 1$ , tandis que  $\|1/(p-1)\|_\infty = \infty$ . Autre part, si  $a > 0$ , la condition  $\|H\|_{a,\infty} < \infty$  garantit que les pôles du transfert  $H$  sont dans le demi-plan gauche limité par la droite  $\text{Re}(p) = -a$  du plan complexe.

Afin de synthétiser une commande par retour de sortie statique pour un système en boucle ouverte instable, la procédure suivante paraît alors naturelle.

Etant donné un correcteur initial  $K_0$  qui n'assure pas la stabilité du système en boucle fermée. Nous choisissons  $a_0 > 0$  telle que la norme  $H_\infty$  décalée de  $a_0$  du système en boucle fermée soit finie :

$$\|C_2[pI - (A + B_2KC_2)]^{-1}B_2\|_{-a_0,\infty} < \infty \quad (6-3)$$

Le problème de recherche du correcteur  $K$  stabilisant peut donc être formulé par le problème d'optimisation

$$\min_K \|C_2[pI - (A + B_2KC_2)]^{-1}B_2\|_{-a,\infty} \quad (6-4)$$

où le décalage  $a$  est soit maintenu fixe égal au décalage initial  $a_0$ , soit diminué graduellement à chaque itération afin d'accélérer la minimisation. Un correcteur stabilisant  $K$  est évidemment obtenu quand le décalage atteint  $a \leq 0$ , mais très souvent ceci se produit déjà avec la valeur initiale  $a_0$ , de sorte que le décalage ne soit pas même nécessaire en règle générale. Le processus d'optimisation est arrêté dès qu'un correcteur stabilisant est obtenu. Dans le cas où le minimum atteint n'assure pas la stabilité du système bouclé, la seule solution serait de relancer de nouveau une optimisation à partir d'un nouveau correcteur initial  $K_0$  ou de basculer vers une autre méthode d'optimisation comme dans le cadre des stratégies globales d'optimisation.

À première vue, une garantie de convergence si locale peut sembler faible, mais l'expérience prouve que les méthodes d'optimisation locales sont beaucoup plus performantes que les techniques globales. Ces dernières possèdent des garanties de convergence plus fortes, mais conduisent souvent à des problèmes numériques difficiles même pour des problèmes de petite taille. À l'opposé, l'approche proposée ici aboutit presque toujours dès le premier essai.

Le décalage initial pour la norme  $H_\infty$  est choisi selon la règle suivante :

$$a_0 = \max(1.05 \cdot \alpha(A), 10^{-2}) \quad (6-5)$$

où  $\alpha(A)$  est le rayon spectral de la matrice  $A$ .

Le correcteur est initialisé par  $K_0 = 0$ . Le calcul de la norme  $H_\infty$  est effectué par la fonction "hinfnorm" de la boîte à outils "Mu Analysis and Synthesis". Les gradients de l'algorithme AGU sont simplement estimés par une différence finie de second ordre centrée.

Les résultats d'optimisation obtenus en appliquant les algorithmes AGU et ASM sont donnés par le tableau 6.1. Ce tableau récapitule les résultats obtenus pour une sélection de problèmes de stabilisation

statique tirés de la littérature. Le triplé  $(n, m, p)$  renseigne la dimension de l'espace d'état, le nombre d'entrée et le nombre de sortie du système à commander. La colonne "Iter" correspond au nombre d'itérations nécessaires pour atteindre le test d'arrêt ( $a \leq 0$ ). La colonne  $\alpha$  représente l'abscisse spectrale finale. Une abscisse spectrale négative implique que le système en boucle fermée est stable. La colonne "Temps" donne le temps de l'unité central de calcul en seconde. La colonne "Réf" renseigne la référence du problème traité.

Problème	Réf	$(n, m, p)$	Formulation $H_\infty$					
			ASM			AGU		
			Iter	$\alpha$	CPU (s)	Iter	$\alpha$	CPU (s)
Transport airplane (AC8)	[Gan86]	(9, 5, 1)	3	-3.01e-3	0.15	1	-2.22e-2	0.11
Horisberger's example (NN6)	[Hor74]	(9, 1, 4)	13	-0.021	0.32	4	-0.01	0.98
VTOL helicopter (HE1)	[Kee88]	(4, 2, 1)	1	-1.31e-2	0.01	1	-6.00e-2	0.05
Chemical Reactor (REA2)	[Hun82]	(4, 2, 2)	2	-1.73	0.06	1	-1.73	0.13
Piezoelectric actuator (PAS)	[Che98]	(5, 1, 3)	2	-1.07	0.10	1	-9.95e-1	0.08
Boeing 767 (AC10)	[Dav90]	(55, 2, 2)	3	-5.61e-3	0.32	1	-2.33e-2	0.24

Tab. 6.1: Résultats des problèmes d'optimisation de type (6-4)

Nous observons qu'à l'exception du résultat du problème de Horisberger [Hor74], l'algorithme AGU s'arrête au bout d'une seule itération. Un correcteur stabilisant est alors obtenu très rapidement. Le choix du décalage initial  $a_0$  est suffisant. Comme nous l'avons déjà indiqué, nous n'avons pas besoin de poursuivre l'exécution de l'algorithme jusqu'à la convergence.

Dans le deuxième exemple, nous avons dû réduire le décalage trois fois avant que la stabilité soit atteinte. Ceci a été fait en appliquant la règle (6-5) à la dynamique du système en boucle fermée. Notons que dans cet exemple, le critère  $H_\infty$  est plat autour d'un optimum global qui sera d'ailleurs atteint parce qu'une norme  $H_\infty$  nulle est obtenue.

La deuxième alternative pour synthétiser des correcteurs statiques  $K$  consiste à utiliser la notion d'abscisse spectrale. Nous proposons alors de résoudre le problème d'optimisation non lisse

$$\min_K \alpha(A + B_2 K C_2) \tag{6-6}$$

Ce type de problème a été déjà traité et analysé dans le chapitre 4. Il s'agit ici de minimiser l'abscisse spectrale jusqu'à l'apparition du premier correcteur  $K$  stable.

Pour la même série de problèmes et pour le même correcteur initial, les résultats numériques des algorithmes AGU et ASM sont données par le tableau ci-dessous.

Problème	Réf	$(n, m, p)$	Formulation abscisse spectrale					
			ASM			AGU		
			Iter	$\alpha$	CPU (s)	Iter	$\alpha$	CPU (s)
Transport airplane (AC8)	[Gan86]	(9, 5, 1)	1	-1.01e-1	0.04	1	-5.23e-2	0.19
Horisberger's example (NN6)	[Hor74]	(9, 1, 4)	11	-0.02	0.32	3	-1.87e-2	1.04
VTOL helicopter (HE1)	[Kee88]	(4, 2, 1)	2	-6.74e-2	0.07	1	-0.05	0.03
Chemical Reactor (REA2)	[Hun82]	(4, 2, 2)	4	-0.98	0.16	2	-1.27	0.23
Piezoelectric actuator (PAS)	[Che98]	(5, 1, 3)	3	-0.87	0.09	1	-1.28	0.08
Boeing 767 (AC10)	[Dav90]	(55, 2, 2)	2	-1.04e-2	0.41	3	-4.66e-2	0.17

Tab. 6.2: Résultats des problèmes d'optimisation de type (6-6)

Ces résultats montrent que les algorithmes développés présentent le même degré d’efficacité vis-à-vis des six problèmes traités et ceci pour les deux formulations. Il est difficile de partager les deux approches, si ce n’est le choix de la valeur du décalage  $a$  et le calcul de la norme  $H_\infty$  et de son gradient qui sont très déterminants et nécessitent plus de précaution numérique.

Outre les résultats présentés, nous avons testé l’algorithme AGU pour des problèmes de stabilisation sur des systèmes Benchmark tirés de la librairie *COMPLieb* [Lei03]. Cette dernière a été proposée par F. Leibfritz pour des tester les algorithmes de programmation semi-définie. Elle réunit une grande collection de problèmes test industriels issus de la littérature. Le tableau ci-dessous résume les résultats numériques que nous avons obtenus :

Problème	Temps (s)	$\alpha(A+BKC)$	Problème	Temps (s)	$\alpha(A+BKC)$
AC1	1.9e-1	-7.24e-3	DIS4	7.8e-1	-1.91e-1
AC2	1.6e-1	-7.24e-3	DIS5	-	(instable)
AC4	3.1e-2	-5.00e-2	WEC1	3.1e-2	-2.43e-2
AC5	-	(instable)	BDT2	1.3e-1	-5.16e-3
AC7	3.1e-2	-4.82e-3	IH	6.3e-2	-1.01e-3
AC9	4.5e-1	-1.14e-3	TF1	3.4e+1	-1.28e-2
AC11	4.7e-2	-3.80e-4	TF2	1.0e+1	-1.00e-5
AC12	2.5e-1	-8.39e-3	TF3	2.3e+1	-1.92e-3
AC13	1.4e-1	-9.29e-5	NN1	1.1	-2.47e-2
AC14	1.1e-1	-9.29e-5	NN2	1.6e-2	-1.25e-1
AC18	3.3e-1	-3.10e-1	NN3	-	(instable)
HE3	1.2	-3.40e-3	NN5	3.1e-2	-4.39e-4
HE4	1.0	-7.57e-5	NN7	8.3e-1	-7.84e-3
HE5	4.8e-1	-3.09e-5	NN9	2.5	-1.67e-3
HE6	4.7e-2	-5.62e-4	NN10	-	(instable)
HE7	3.1e-2	-5.62e-4	NN12	-	(instable)
JE2	1.3	-1.33e-2	NN13	2.7e-1	-1.47e-1
JE3	1.0	-1.38e-2	NN14	2.5e-1	-1.47e-1
REA1	3.1e-2	-4.22e-2	NN15	4.7e-2	-3.28e-3
REA3	3.1e-2	-2.07e-2	NN16	3.1e-2	-3.50e-1
DIS2	1.2e-2	-8.83e-2	NN17	4.7e-2	-1.81e-3

Tab. 6.3: Résultats de la stabilisation pure d’une série de systèmes de la librairie *COMPLieb*

Ces résultats sont très satisfaisants. La stabilisation est achevée pour 37 systèmes sur les 42 étudiés. Le temps d’exécution de l’algorithme AGU est de moins d’une seconde pour 28 systèmes.

Ces performances sont largement supérieures à ceux obtenus avec les algorithmes de programmation semi-définie qui traitent des problèmes équivalents mais de tailles beaucoup plus grandes à cause de l’introduction des variables de Lyapunov (formulation LMI).

### 6.1.2. Optimisation de l’abscisse spectrale

Considérons le problème de stabilisation qui consiste à déterminer les coefficients du polynôme multiparamétrique  $P_x(\lambda)$  qui correspondent à une abscisse spectrale minimale

$$P_x(\lambda) = \lambda^n + x_1\lambda^{n-1} - x_1\lambda^{n-2} + \sum_{j=3}^n x_{j-1}\lambda^{n-j} \quad \text{avec} \quad x_1 \neq 0 \quad (6-7)$$

Ce problème est très intéressant pour tester des algorithmes d’optimisation car il présente de très nombreuses dégénérescences et il est suffisamment simple pour pouvoir déterminer des propriétés a priori [Bur04].

En particulier, la condition nécessaire de stabilité (critère de Routh) n’est pas vérifiée pour ce polynôme, car la liste de ses coefficients comporte un changement de signe entre les deux coefficients successifs  $x_1$  et  $-x_1$ . Par conséquent, il existe au moins une racine à partie réelle positive et le polynôme (6-7) est instable. On écrit :

$$f(x) = \max_i(\text{Re}(\lambda_i(x))) \geq 0 \quad \text{avec} \quad P_x(\lambda_i(x)) = 0 \quad (6-8)$$

La valeur  $f^* = 0$  est non seulement un minorant de la fonction  $f$  mais aussi un minimum car il suffit de poser  $x_{n-1} = 0$  pour qu’il existe au moins une racine réelle nulle et donc d’atteindre  $f^* = 0$ .

Le but de cette exemple est de tester l’efficacité des deux algorithmes d’optimisation développés (AGU et ASM) pour un problème d’optimisation de plus grande taille.

L’abscisse spectrale du polynôme  $P_x(\lambda)$  est équivalente à celle de sa matrice compagne  $A_x$  définie par :

$$A_x = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -x_{n-1} & -x_{n-2} & \cdots & x_1 & -x_1 \end{pmatrix} \quad (6-9)$$

Ainsi, nous avons préféré d’évaluer la fonction critère en utilisant les fonctions du calcul matriciel sous Matlab (calcul des valeurs propres) qui offrent une robustesse de calcul par rapport au celles du calcul polynomial (calcul des racines d’un polynôme).

Quant aux gradients utilisés dans l’algorithme AGU, ils ont été évalués en utilisant un calcul formel sur les polynômes sous Mathematica. Cette méthode exacte est très pratique mais elle nécessite un investissement en amont en terme de temps.

Pour un degré  $n = 6$ , le tableau suivant résume les résultats des optimisations effectuées avec les deux algorithmes ASM et AGU. Les deux algorithmes sont initialisés à partir d’un même point choisi aléatoirement. Dans ce cas, il est donné par :  $x_0 = [0.1731, 0.7931, 0.5137, 0.8128, 0.4539]$

Nous remarquons de ces résultats que l’algorithme ASM peine à atteindre le minimum et se bloque rapidement au bout de 155 itérations. Pour cet algorithme, le minimum est grossièrement approché avec un nombre d’itérations très petit comparant à celui de l’algorithme AGU.

Algorithme	ASM	AGU
Critère $\bar{f}$	4.5616e-01	8.5099e-04
Solution $\bar{x}$	[0.277, 1.330, 0.520, -0.475, 0.693]	(-1e-3, -6e-5, 2e-5, -8e-8, 1e-8)
Nb itérations	155	458
Nb éval. $f$	265	10110
Nb éval. $\nabla f(x)$	0	3680
$\ \nabla f(x)\ _2$	26.5159	2.6384e+04
Temps de calcul	5.2380	80.4409

Tab. 6.4: Résultats du problème de minimisation de l’abscisse spectrale (6-8)

La figure 6.1 représente quelques itérations de cet algorithme. Elle montre les difficultés auxquelles l'algorithme est confronté. En effet, à partir de l'itération 40, deux paires de valeurs propres complexes conjuguées se trouvent alignées autour d'une partie réelle minimale de 0,45615. Cette difficulté est accentuée ensuite lorsque ses deux paires se rapprochent pour se fusionner en une seule paire double, l'algorithme se bloque alors définitivement après avoir procédé aux différentes réinitialisations pour échapper à une éventuelle dégénérescence.

Ce résultat n'est pas très surprenant vu la complexité du problème traité et sa dimension qui ne facilitent pas la recherche d'une direction de descente. En effet, les méthodes de type simplexe perdent rapidement leurs facultés d'adaptation pour les problèmes de « grande » dimension en présence de dégénérescences.

A l'opposé, la solution de l'algorithme AGU est obtenue au bout de 458 itérations et démontre son efficacité et sa robustesse vis-à-vis du problème de valeurs propres. Nous signalons que ce résultat pouvait bien être plus précis si nous avions diminué le rayon d'échantillonnage  $\varepsilon$  davantage lors de l'estimation du  $\varepsilon$ -sous-différentiel. Ici, nous nous sommes contentés d'un rayon  $\varepsilon = 10^{-6}$ . L'algorithme termine son avancée avec 3 paires de valeurs propres complexes conjuguées alignées sur la même abscisse spectrale (cf. figure 6.2).

Même si nous ne pouvons pas représenter ce critère multidimensionnel, géométriquement, le minimum est atteint en un point très difficile à atteindre : le cas bidimensionnel était déjà très difficile pour des algorithmes d'optimisations classiques (cf. section 4.3.2.2.5). Ceci peut être expliqué comme suit : les zones non différentiables naissent d'une déformation brusque de l'espace paramétrique causée par une perte de dimension dans l'espace des valeurs propres de la matrice  $A_x$ . Plus la dimension de cet espace est réduite, plus le problème est difficile à résoudre.

Dans notre exemple, le minimum de la fonction abscisse spectrale (6-8) correspond à une matrice en forme d'un bloc de Jordan Nilpotent où le degré de multiplicité algébrique est égal à 4 et le degré de multiplicité géométrique est égal à 1. En d'autres termes, la seule valeur propre de la matrice  $A_x$  est quadruple et son espace propre est engendré par un seul vecteur propre  $v = [1, 0, 0, 0]^T$ .



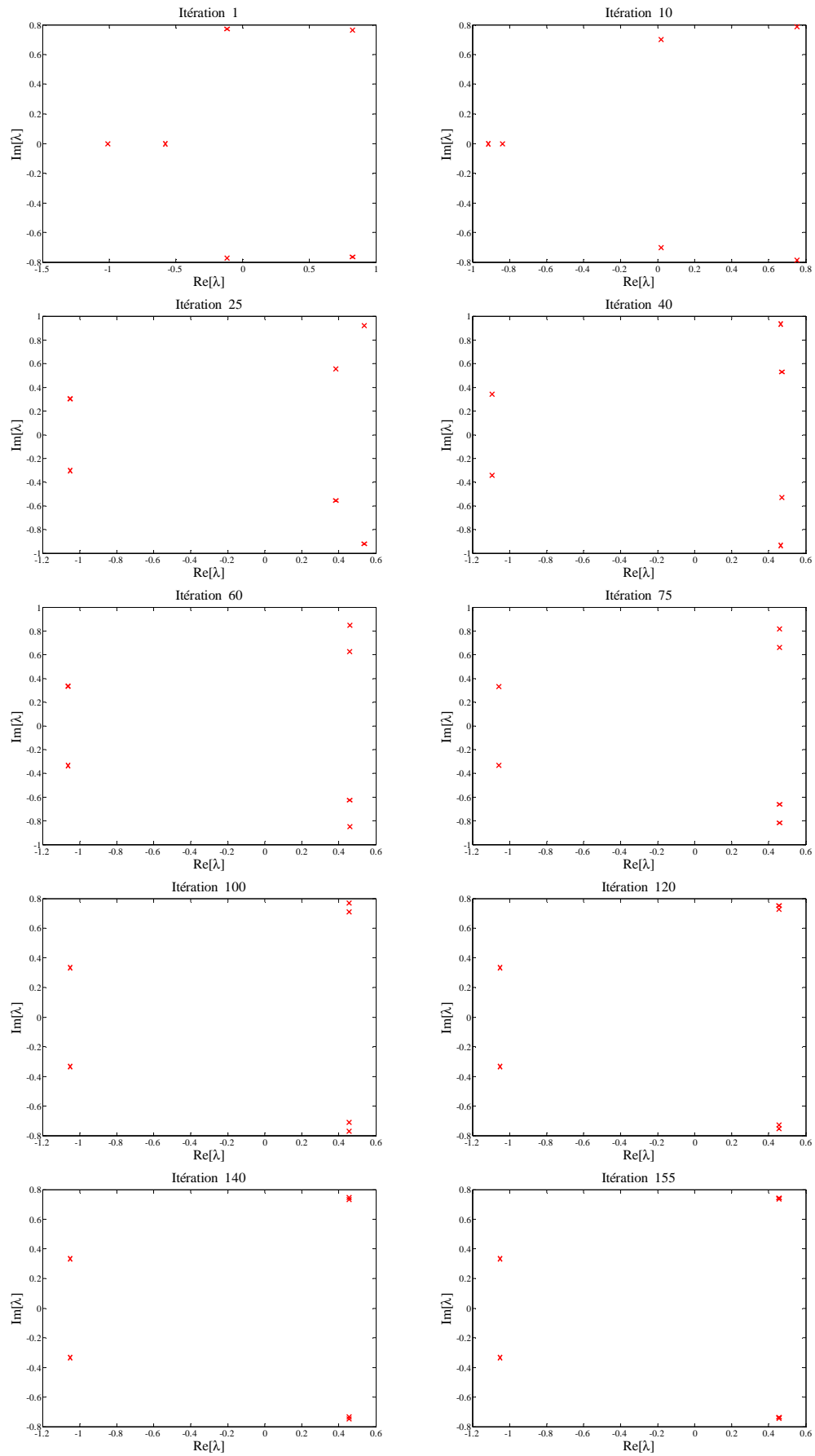


Fig. 6.1 – Itérations de l’algorithme ASM

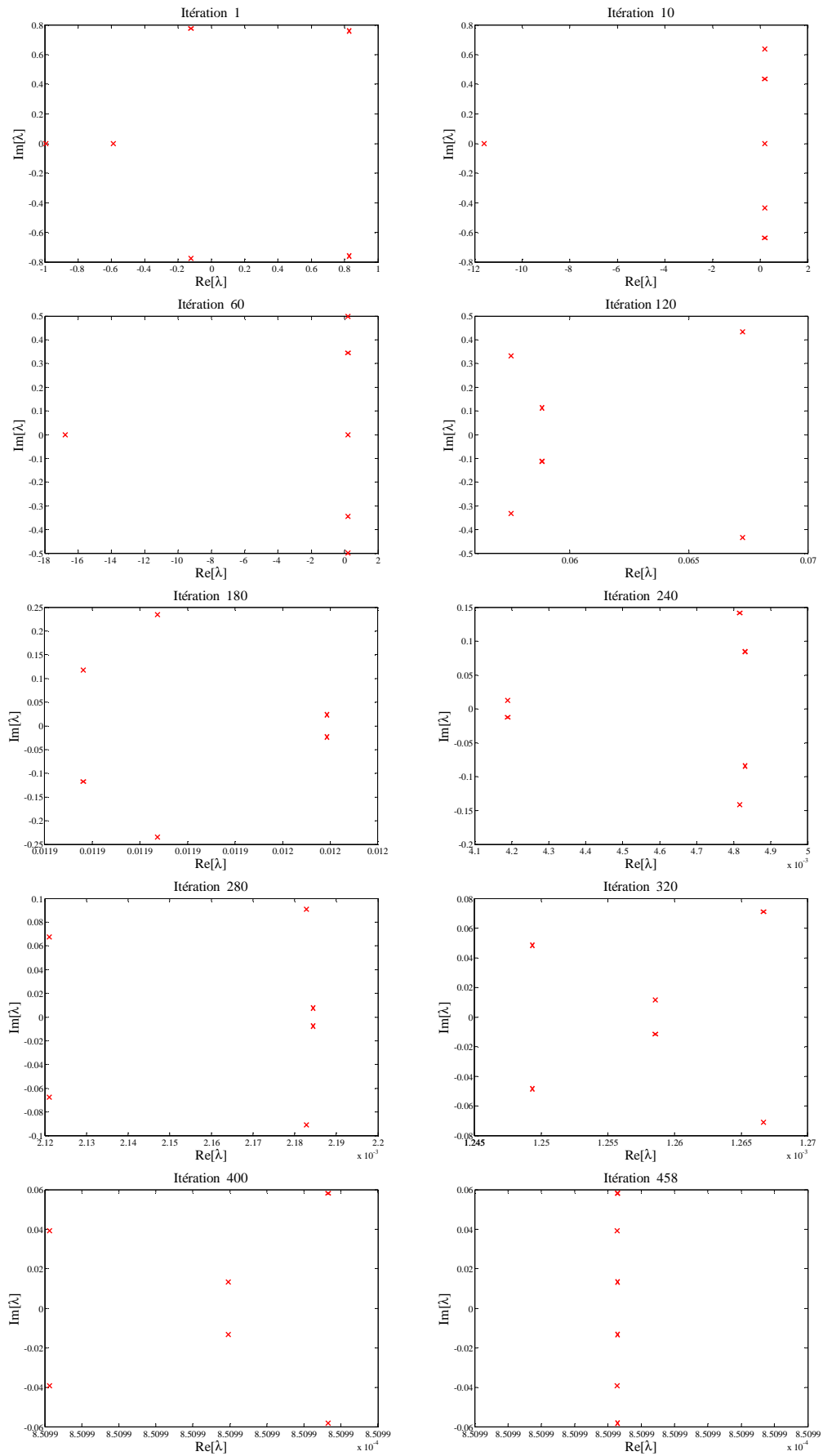


Fig. 6.2 – Itérations de l’algorithme AGU

### 6.1.3. Problème de stabilisation simultanée (Application au problème du chocolat belge)

Le problème de stabilisation simultanée consiste à trouver un contrôleur qui stabilise un ensemble fini de systèmes. Il est souvent formulé comme suit :

*Etant donné  $n$  systèmes monovariables propres  $P_1(p), P_2(p), \dots, P_n(p)$ , existe-t-il un correcteur unique  $K(p)$  pour lequel le système en boucle fermée à retour unitaire est stable pour chacun des systèmes  $P_i(p)$  ?*

Ce problème est particulièrement intéressant car il est au cœur de plusieurs applications. Une des applications les plus citées est la commande d'un système en fonctionnement nominal et sous plusieurs modes de défaillances, chacun représentés par une description différente de ce même système. Par exemple, le pilotage d'un avion totalement opérationnel et sous plusieurs combinaisons de défaillances de capteurs et actionneurs. Ou bien aussi, un système possédant plusieurs modes de fonctionnement nominal différents. Par exemple, la réponse d'attitude d'un avion aérospatial serait représentée par un modèle mathématique différent selon s'il est hypersonique, supersonique, subsonique ou dans l'espace. De même, en utilisant une heuristique très courante, un contrôleur linéaire pourrait potentiellement stabiliser un système non linéaire s'il stabilise simultanément ses linéarisés tangent autour de plusieurs points de fonctionnement. Ces exemples et d'autres illustrent que les problèmes de stabilisation simultanée sont plus que de simples problèmes académiques.

La stabilisation simultanée est une branche de commande robuste. La commande robuste stabilise simultanément une gamme continue de systèmes dont les paramètres varient dans des régions prédéfinies. La principale différence entre la stabilisation robuste et simultanée est dans le nombre de système que chacune des approches essaye de stabiliser. La stabilisation robuste fait face à un nombre (non dénombrable) infini de systèmes tandis que la stabilisation simultanée traite seulement un nombre fini. Néanmoins, la stabilisation simultanée d'un nombre fini de systèmes est paradoxalement très difficile. À la différence de la stabilisation robuste dans laquelle le continuum de systèmes ne doit pas varier trop loin du système nominal, il ne peut y avoir aucune prétention sur la corrélation du nombre fini de systèmes distincts. Au jour d'aujourd'hui, il y a eu une seule solution complète au problème de stabilisation simultanée et ceci quand il n'y a pas plus de deux systèmes [Blo94a, Blo94b].

L'étude proposée dans cette application est inspirée des travaux de Burke et al. [Bur06a] concernant la résolution d'un problème de commande challenge proposé par Blondel [Blo99]. Ce problème est communément appelé problème du chocolat belge.

Le but de cette étude est de mesurer les performances de l'algorithme AGU que nous avons développé par rapport à l'ensemble des résultats récents exposés dans [Bur06a]. N'ayant pas accès à l'algorithme utilisé dans [Bur06a], nous n'avons pas pu effectuer une étude comparative exhaustive. Ainsi, nous n'avons pas comparé les performances en terme de déroulement de l'algorithme, par contre nous avons pu tout de même comparer les optima obtenus.

Nos résultats sont très satisfaisants. Ils témoignent de l'efficacité de l'algorithme AGU développé qui permet de résoudre tous les problèmes traités et d'atteindre des résultats similaires voir souvent meilleurs que ceux exposés dans [Bur06a].

Le problème benchmark en question est formulé comme suit :

Soit le système linéaire invariant dans le temps (LTI) décrit par la fonction de transfert suivante

$$G(p) = \frac{p^2 - 1}{p^2 - 2\delta p + 1} \quad (6-10)$$

où  $\delta$  est un paramètre réel appartenant à l'intervalle  $[0,1]$ . La forme particulière de cette fonction de transfert provient des travaux de thèse de Vincent Blondel [Blo94a], qui offrait un kilogramme de chocolat belge pour la solution de chacun des problèmes de commande suivants :

**Problème 1 :** trouver la gamme de valeurs du paramètre  $\delta$  pour laquelle il existe un système linéaire invariant dans le temps stable et un contrôleur à minimum de phase stabilisant le système (6-10).

**Problème 2 :** trouver un système linéaire invariant dans le temps stable et un contrôleur à minimum de phase stabilisant le système (6-10) quand  $\delta = 0.9$ .

Le problème 2 a été résolu seulement récemment : un contrôleur d'ordre onze a été trouvé [Pat02] en utilisant une méthode de recherche aléatoire. Le problème 1 est toujours non résolu. Cependant, nous remarquons qu'une stabilisation est impossible dans le cas où  $\delta = 1$ , car il y a une élimination d'un pôle par un zéro qui se produit dans le système  $G(p)$ . Selon [Pat02] et Prachant Batra, les résultats de l'analyse complexe peuvent être utilisés pour démontrer qu'il existe un nombre  $\delta^* < 1$  tel que la stabilisation est possible pour tout  $\delta < \delta^*$ , mais impossible pour  $\delta \geq \delta^*$ .

On peut arguer que les problèmes de chocolat de Blondel sont principalement d'intérêt académique et mathématique. Cependant, une meilleure compréhension mathématique de tels problèmes, même à travers des exemples académiques, peut aider dans la compréhension de problèmes pratiques et réels. Par exemple, on peut rappeler que les problèmes de contrôle avec une proche élimination de pôles instables par des zéros (comme dans le problème du chocolat Belge) surgissent dans des problèmes appropriés d'ingénierie, tel que le problème de conception d'avion du prototype X-29 ou le problème de conception de bicyclette de Klein décrit dans [Ast00].

Dans ce qui suit, nous allons essayer d'apporter une réponse au problème 2 et tenter d'approcher au mieux  $\delta^*$  pour répondre au premier problème. L'approche que nous proposons ici est basée sur une optimisation non lisse d'un critère d'abscisse spectrale que nous allons définir.

Soient  $a(p) = p^2 - 2\delta p + 1$  et  $b(p) = p^2 - 1$ . Le but est de trouver un correcteur  $K(p) = y(p)/x(p)$  stable et d'inverse stable qui stabilise la boucle fermée  $b y / (a x + b y)$ .

Ce problème se traduit alors par une exigence de stabilisation simultanée des transferts monovariabiles  $by/(ax+by)$ ,  $y/x$  et  $x/y$  que nous allons formuler en une minimisation de l'abscisse spectrale du produit des dénominateurs :  $x(p)$ ,  $y(p)$  et le polynôme caractéristique  $a(p)x(p)+b(p)y(p)$ .

$$\min_z \alpha(xy(ax+by)) \quad (6-11)$$

où  $z$  est le vecteur de coefficients des polynômes  $y(p)$  et  $x(p)$ .

Nous fixons l'ordre du correcteur  $K(p)$  à 3 et nous imposons une structure telle que  $\deg(x) = n_k = 3$  et  $\deg(y) = m_k = 0$  avec  $x(p)$  monique.  $\delta$  est initialement fixé à 0.9. Il sera ensuite augmenté au fur et à

mesure, tant que le problème est faisable. Les résultats d’optimisation obtenus par l’algorithme AGU sont donnés ci-dessous.

$\delta$	$y(p)$	$x(p)$
0.9000	1.8868	$p^3 + 2.3629p^2 + 3.3979p + 1.8868$
0.9025	1.4726	$p^3 + 1.8940p^2 + 2.6737p + 1.4729$
0.9050	1.5125	$p^3 + 1.9624p^2 + 2.7461p + 1.5127$
0.9075	1.3989	$p^3 + 1.8339p^2 + 2.5505p + 1.3989$
0.9100	1.4008	$p^3 + 1.8551p^2 + 2.5564p + 1.4008$
0.9125	1.4065	$p^3 + 1.8836p^2 + 2.5709p + 1.4065$
0.9130	1.4168	$p^3 + 1.9012p^2 + 2.5900p + 1.4186$
0.9135	1.3593	$p^3 + 1.8280p^2 + 2.4886p + 1.3593$

Tab. 6.5: Polynômes des correcteurs stabilisants  $(n_k, m_k) = (3, 0)$  en fonction de  $\delta$

L’évaluation du gradient de la fonction critère a été réalisée formellement en utilisant les vecteurs propres.

Les résultats, présentés ci-dessus, correspondent à une série de tests similaire à celle utilisée dans [Bur06a] où l’algorithme AGU est exécuté pour une gamme de  $\delta$  à partir de plusieurs conditions initiales aléatoirement choisies jusqu’à ce que la stabilisation simultanée soit établie. Dans le cas contraire, ce processus est arrêté au bout de 100 exécutions et le problème est considéré comme non faisable.

Comme dans [Bur06a], les premiers résultats obtenus montrent que le problème 2 de Blondel est faisable avec  $b(p) = 1.8868$  et  $a(p) = p^3 + 2.3629p^2 + 3.3979p + 1.8868$ . Ce résultat est de loin meilleur que celui obtenu dans [Pat02] où l’ordre du correcteur est égal 11. Avec la structure du correcteur choisie  $(n_k, m_k) = (3, 0)$ , nous avons même réussi d’atteindre la stabilité pour une valeur de  $\delta$  égale 0.9135 qui est meilleure que celle obtenue dans [Bur06a]. Cette valeur pratique correspond en réalité à une approximation d’une limite théorique très difficile à atteindre où le  $\delta$  est égale à  $\bar{\delta} = (2 + \sqrt{2})^{\frac{1}{2}} / 2$ .

En effet, nous observons des différents lieux des racines de la figure 6.3 que les pôles de la boucle fermée (en rouge) sont les plus dominants (par rapport à ceux du correcteur) et ils sont de plus en plus regroupés autour de l’origine. Ils tendent tous vers un pôle quintuple nulle. Ce constat est similaire à ce que nous avons déjà vu avec l’exemple de minimisation de l’abscisse spectrale (cf. section 6.1.2). Les pôles du correcteur sont, dans tous les cas testés, plus rapides et ne sont pas actives lors des différentes itérations de l’algorithme AGU.

En partant de ce constat, le polynôme caractéristique en boucle fermée tend à vérifier :

$$a(p)x(p) + b(p)y(p) = (p^2 - 2\delta p + 1)(p^3 + \sum_{i=0}^2 x_{\delta,i} p^i) + (p^2 - 1)y_{\delta,0} = (p - z_\delta)^5 \quad (6-12)$$

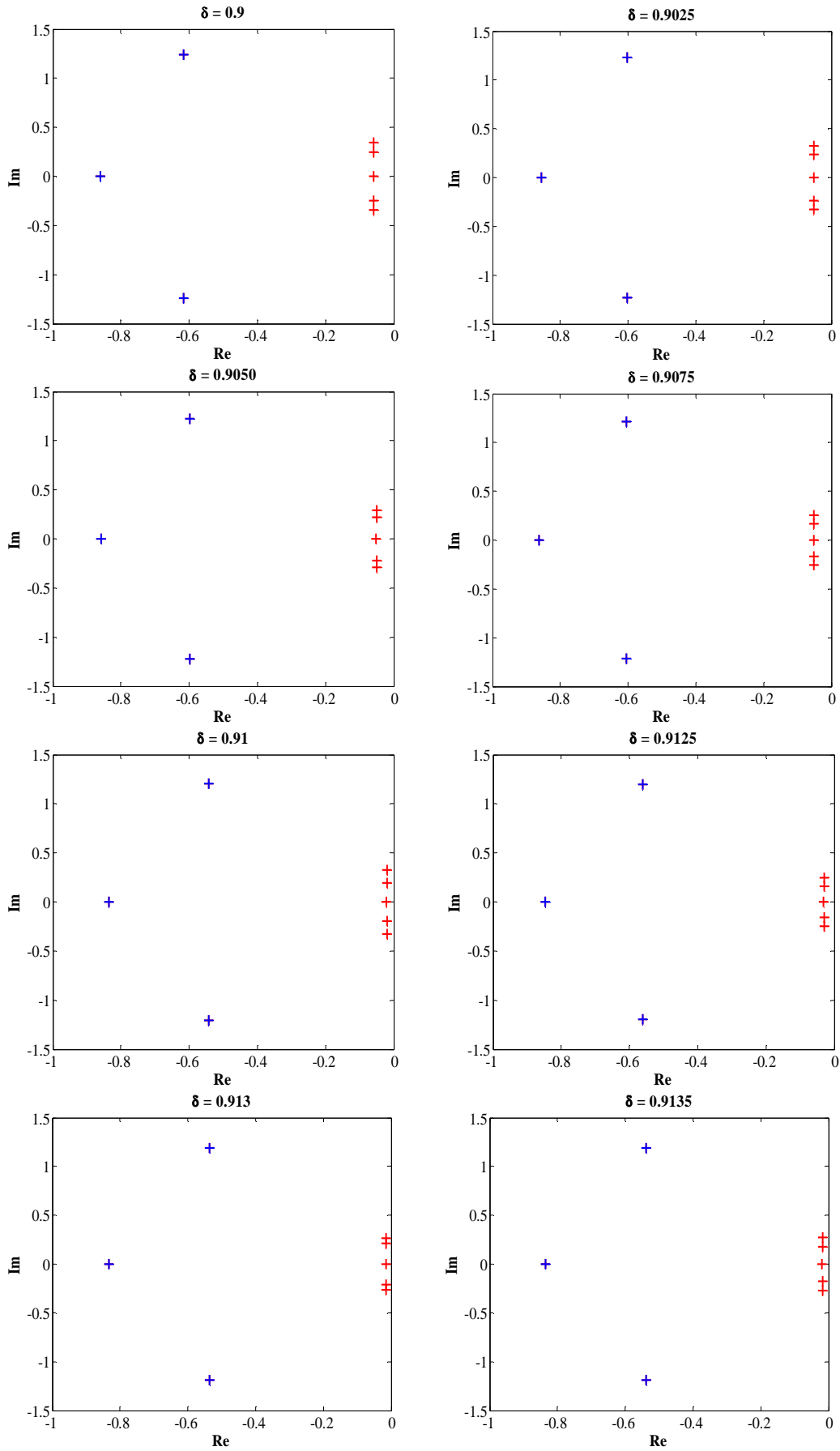


Fig. 6.3 – Lieu des pôles optimaux : + pôle du correcteur et + pôle de la boucle fermée

Ce qui est équivalent au système d'équations :

$$\begin{cases} -2\delta + x_{\delta,2} + 5z_{\delta} = 0 \\ -2\delta x_{\delta,2} + x_{\delta,1} + 1 - 10z_{\delta}^2 = 0 \\ -2\delta x_{\delta,1} + x_{\delta,2} + x_{\delta,0} + y_{\delta,0} + 10z_{\delta}^3 = 0 \\ -2\delta x_{\delta,0} + x_{\delta,1} - 5z_{\delta}^4 = 0 \\ x_{\delta,0} - y_{\delta,0} + z_{\delta}^5 = 0 \end{cases} \quad (6-13)$$

Après résolution, nous obtenons l'équation polynomiale suivante :

$$\delta(5z_{\delta} - 2\delta)(-3 + 4\delta^2) - (1 - 10z_{\delta}^2 + 5z_{\delta}^4 - 10z_{\delta}^3\delta - z_{\delta}^5\delta - 2\delta^2 + 20z_{\delta}^2\delta^2) = 0 \quad (6-14)$$

En choisissant comme pôle quintuple  $z_{\delta} = 0$ , nous obtenons la nouvelle équation :

$$-8\delta^4 + 8\delta^2 - 1 = 0 \quad (6-15)$$

qui admet deux solutions réelles positives dont la maximale est égale à  $\bar{\delta} = (2 + \sqrt{2})^{\frac{1}{2}} / 2$ .

Ainsi, pour  $\delta$  proche de  $\bar{\delta}$ , l'équation polynomiale (6-14) admet une solution  $z_{\delta}$  autour de 0. Cette solution croissante en fonction de  $\delta$  vérifie  $z_{\bar{\delta}} = 0$ . En outre, il existe des fonctions dépendantes de  $\delta$  pour lesquelles l'égalité (6-12) est vérifiée et le polynôme  $x(p)$  est stable avec  $x_{\delta} = [2\delta, -1 + 4\delta^2, 2\delta - 1/2\delta]$  et  $y_{\delta} = [2\delta - 1/2\delta]$ .

En appliquant le même principe, nous allons essayer de voir si ce raisonnement reste valable pour les correcteurs de même structure mais d'ordre plus élevé.

Les résultats d'optimisation obtenus avec  $(n_k, m_k) = (4,0)$  sont résumés dans le tableau et la figure qui suivent :

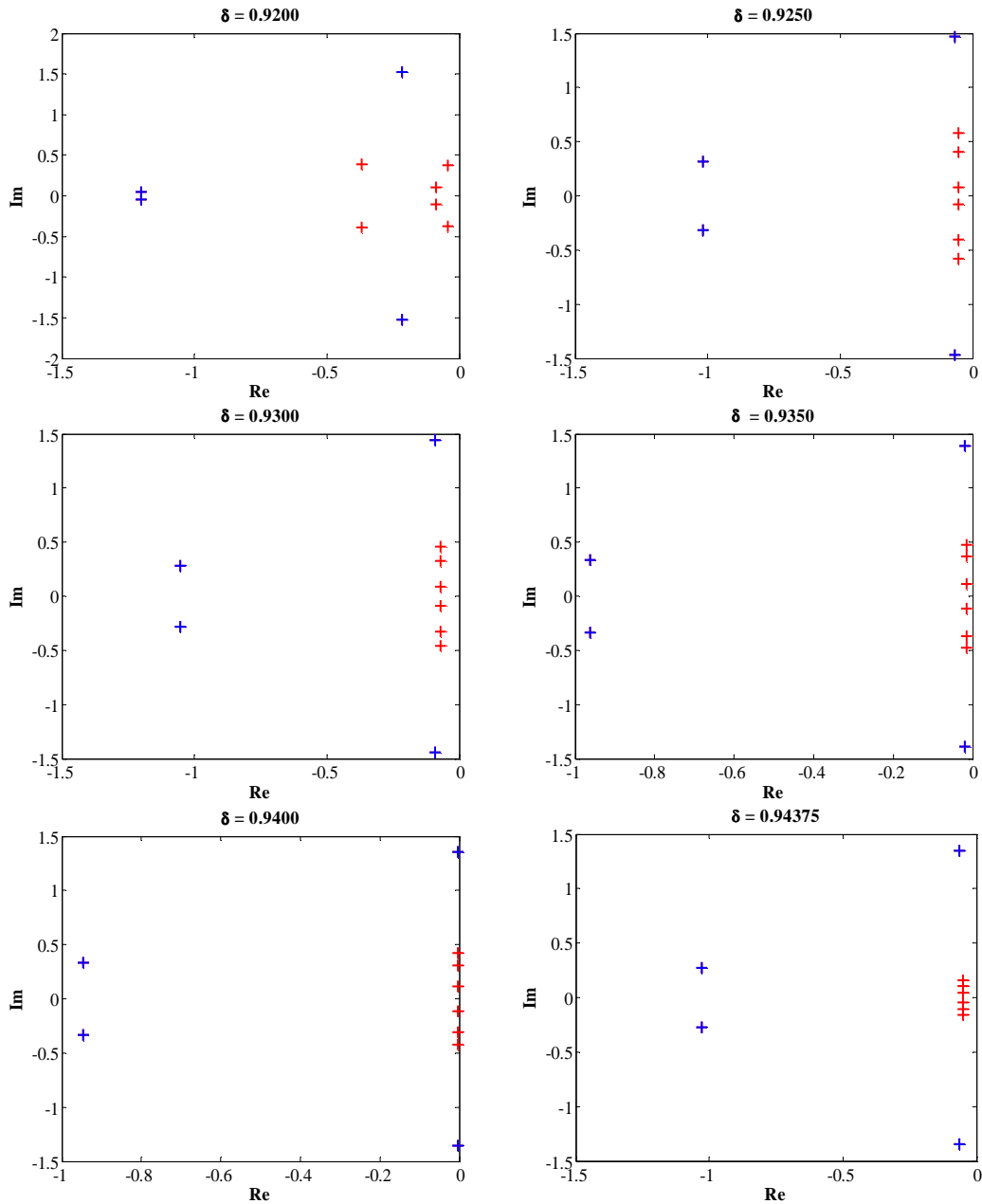


Fig. 6.4 – Lieu des pôles optimaux : + pôle du correcteur et + pôle de la boucle fermée

Nous observons dans ce cas que l’abscisse spectrale du système global dépend de plus en plus de l’emplacement des pôles du correcteur. En effet, une paire de pôles complexes conjugués du correcteur devient active autour des solutions obtenues.

$\delta$	$y(p)$	$x(p)$
0.9200	3.4551	$p^4 + 2.8390p^3 + 4.8790p^2 + 6.3684p + 3.4559$
0.9250	2.4495	$p^4 + 2.1686p^3 + 3.5655p^2 + 4.5391p + 2.4500$
0.9300	2.4707	$p^4 + 2.2858p^3 + 3.6491p^2 + 4.6001p + 2.4710$
0.9350	2.0113	$p^4 + 1.9633p^3 + 3.0497p^2 + 3.7630p + 2.0117$
0.9400	1.8461	$p^4 + 1.8977p^3 + 2.8541p^2 + 3.4713p + 1.8464$
0.94375	2.0466	$p^4 + 2.1853p^3 + 3.1991p^2 + 3.8629p + 2.0466$

Tab. 6.6: Polynômes des correcteurs stabilisants  $(n_k, m_k) = (4, 0)$  en fonction de  $\delta$



En répétant le même calcul que précédemment, nous arrivons à une valeur limite  $\bar{\delta} = (5/8 + \sqrt{5}/8)^{\frac{1}{2}}$  et à des coefficients  $x_{\delta} = [-3\delta + 4\delta^2, -1 + 2\delta^2, \delta, 1/2]/(1 - 8\delta^2 + 8\delta^4)$  et  $y_{\delta} = [1/2/(1 - 8\delta^2 + 8\delta^4)]$ .

Nous vérifions facilement que pour cette solution le correcteur est à la limite de la stabilité et possède une paire de pôles imaginaires pure.

Les résultats d'optimisation réalisés montrent que la stabilité a pu être assurée jusqu'à une limite  $\delta$  égale à 0.94375. Cette fois, cette valeur est similaire à la valeur optimale obtenue dans [Bur06a] même si elle reste loin de la limite théorique. Ceci est surtout dû à la complexité du critère optimisé et à la notion d' $\varepsilon$ -stationnarité qui peut être toujours améliorée en réitérant davantage en diminuant le rayon  $\varepsilon$ .

Même en augmentant l'ordre  $n_k$  à 5, les résultats de simulation montrent qu'il est extrêmement difficile d'augmenter la limite  $\bar{\delta}$ . En effet, pour cette structure du correcteur, les cinq pôles de la boucle fermée se joignent autour de l'origine et deux paires de pôles complexes conjugués du correcteur deviennent actives (cf. figure 6.4)

$\delta$	$y(p)$	$x(p)$
0.95	39.4262	$p^5 + 23.6561p^4 + 45.7167p^3 + 63.4771p^2 + 74.9098p + 39.4262$
0.9502	44.9564	$p^5 + 27.0940p^4 + 52.2482p^3 + 72.4326p^2 + 85.4161p + 44.9464$

Tab. 6.7: Polynômes des correcteurs stabilisants  $(n_k, m_k) = (5, 0)$  en fonction de  $\delta$

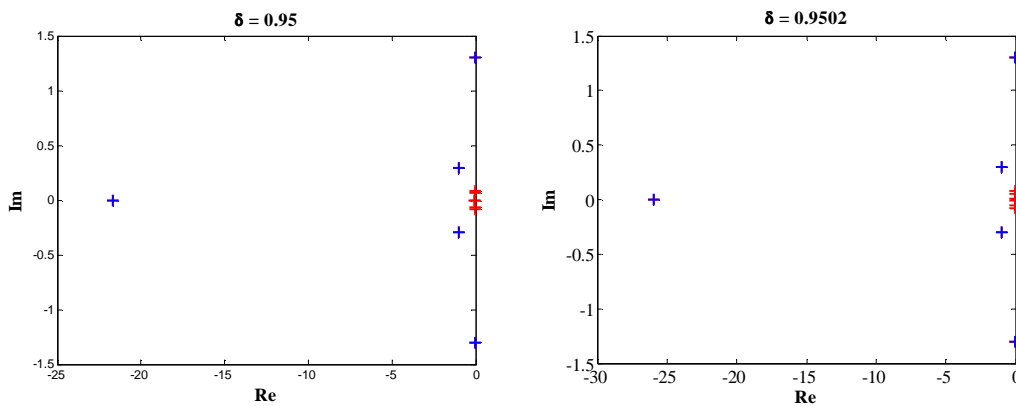


Fig. 6.6 – Lieu des pôles optimaux : + pôle du correcteur et + pôle de la boucle fermée

La meilleure valeur que nous avons pu atteindre avec l'algorithme AGU avec ses paramètres par défaut est  $\bar{\delta} = 0.9502$ . Elle est légèrement meilleure que celle exposée dans [Bur06a].

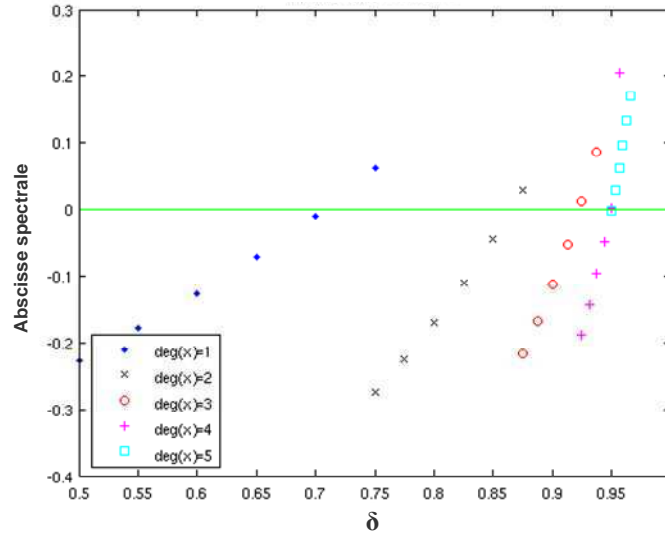


Fig. 6.7 – Variation de l’abscisse spectrale en fonction de l’ordre du correcteur et de  $\delta$

La figure ci-dessus résume l’ensemble des résultats obtenus en conservant la même structure du correcteur et en variant l’ordre de son dénominateur  $x(p)$  seulement. Nous observons que pour  $\text{deg}(x) = n_k = 5$ , la stabilisation du système devient quasi impossible et que l’augmentation de l’ordre du correcteur n’améliorera pas la limite  $\bar{\delta}$  significativement (saturation autour de 0.95).

Il serait donc plus judicieux d’explorer la dynamique des zéros du correcteur. Dans ce cas, le problème est structurellement plus compliqué (contrainte sur les pôles et les zéros d’un système) mais ne changera rien de notre approche qui reste générique.

En conservant l’ordre du correcteur  $K(p)$  à 5 tout en augmentant le degré du numérateur et dénominateur à  $(n_k, m_k) = (7, 2)$ , la limite  $\bar{\delta}$  a été améliorée. La meilleure valeur limite que nous avons pu atteindre est de 0.965. Elle est, une fois de plus, meilleure que la limite 0.96 obtenue par Burke et al. [Bur06a].

Le tableau qui suit donne les polynômes optimaux du correcteur en fonction de  $\delta$ .

$\delta$	$y(p)$	$x(p)$
0.96	$6.9032p^2 + 0.4668p + 7.7717$	$2.2338p^7 + 7.3226p^6 + 12.7877p^5 + 19.1596p^4 \dots$ $+ 24.3325p^3 + 20.9162p^2 + 15.3890p + 7.7717$
0.962	$7.7464p^2 + 0.7148p + 7.5937$	$2.4158p^7 + 7.9613p^6 + 14.3898p^5 + 21.0346p^4 \dots$ $+ 26.3827p^3 + 22.0452p^2 + 15.3252p + 7.5937$
0.965	$6.0243p^2 + 0.4507p + 5.9481$	$2.2830p^7 + 6.8955p^6 + 11.9469p^5 + 16.8668p^4 \dots$ $+ 20.7307p^3 + 17.1545p^2 + 11.9305p + 5.9481$

Tab. 6.8: Polynômes des correcteurs stabilisants  $(n_k, m_k) = (7, 2)$  en fonction de  $\delta$

La figure ci-dessous représente les lieux des pôles de ces trois systèmes en boucle fermée.

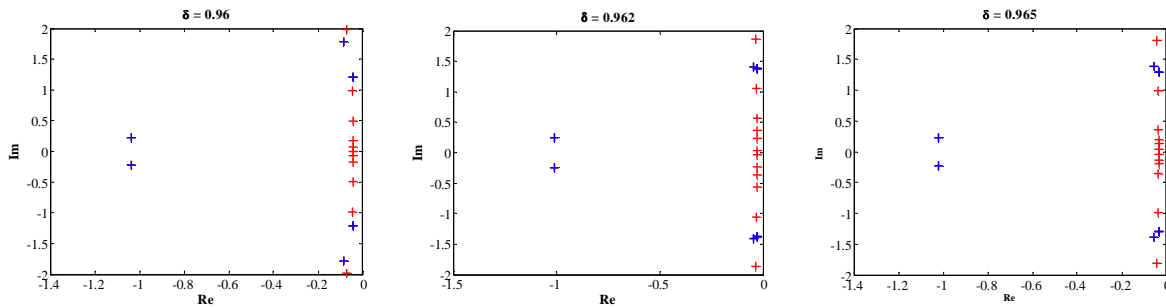


Fig. 6.8 – Lieu des pôles optimaux : + pôle du correcteur et + pôle de la boucle fermée

En pariant sur l'efficacité de l'algorithme d'optimisation utilisé, nous pouvons affirmer qu'une telle limite  $\bar{\delta}$  peut être améliorée davantage en augmentant les degrés de liberté du correcteur. Toutefois, le choix de la structure adéquate du correcteur reste le l'obstacle fondamental à surmonter dans ce type de problème de stabilisation.

## 6.2. Synthèse $H_\infty$ à structure fixe

Le problème  $H_\infty$  consiste à trouver une matrice de retour de sortie  $K$  stabilisante et qui minimise la norme  $H_\infty$  de certains transferts de la boucle fermée.

Dans cette section, nous étudions le problème de synthèse  $H_\infty$  avec ajout de contraintes de structure sur le correcteur. Ceci peut inclure la synthèse  $H_\infty$  par retour de sortie statique, la synthèse  $H_\infty$  à ordre réduit, la stabilisation, la stabilisation robuste, la stabilisation simultanée et la synthèse  $H_\infty$  multi-objectifs.

Dans une synthèse  $H_\infty$  nominale, les correcteurs sont solution d'un problème de programmation semi-définie (SDP) [Gah93, Apk95] d'un ensemble d'équations algébriques de Riccati [Doy89a]. En ajoutant les contraintes structurelles du correcteur, le problème  $H_\infty$  ne devient plus convexe et certain des problèmes précités deviennent même d'une complexité non polynomiale [Nem94, Blo97] ou rationnellement indécidables [Blo94b]. Les concepts mathématiques en relation avec ce problème montrent bien la difficulté incontournable de la synthèse  $H_\infty$  sous des contraintes de correcteur.

Les méthodes numériques existantes pour la synthèse  $H_\infty$  à structure fixe sont souvent basées sur une reformulation du problème en un problème LMI (Inégalités Matricielles Linéaires) et une contrainte de rang non convexe ou une contrainte d'inégalité non convexe. Les méthodes numériques de telles formulations du problème contient ceux à base de linéarisation [Elg97, Iba01, Lei01] ; projection alternée [Gri96, Gri99, Ors04] ; les méthodes de Lagrangien augmenté [Apk03, Apk04, Far01, Nol04] ; et la programmation semi-définie [Far02].

Le problème de synthèse  $H_\infty$  peut aussi être formulé en un problème BMI (Inégalités Matricielles Bilinéaires). En effet, sous contraintes structurelles, le lemme réel borné reste toujours applicable. Cependant, son utilité dans le contexte d'une synthèse  $H_\infty$  n'est plus la même car il n'aboutit plus à un problème convexe de type LMI mais un problème d'optimisation BMI, non convexe. Des études [Far02, Lei02, Kan04, The04] concernant la résolution des problèmes BMI montrent que les algorithmes développés souffrent de plusieurs problèmes numériques même pour des problèmes

d'ordre modéré. Ceci est essentiellement dû à la présence des variables de Lyapunov : leur nombre croît quadratiquement avec le nombre d'état du système. Ainsi, le nombre total des variables de décision devient très grand ce qui engendre des difficultés numériques même avec des problèmes de petite dimension.

Dans cette étude, la synthèse  $H_\infty$  est posée comme un problème de minimisation non convexe non différentiable sans contrainte. L'approche proposée, ici, n'utilise pas le lemme réel borné et ainsi elle évite l'utilisation des variables de Lyapunov ce qui conduit à des problèmes d'optimisation de dimension moyenne même avec des grands systèmes. En revanche, les fonctions coût deviennent non lisses et requièrent des techniques d'optimisation spécifiques similaires à ce que nous avons développé. L'évaluation de la norme  $H_\infty$  est accomplie par la méthode de bisection Hamiltonienne et elle est employée davantage pour le calcul des sous-gradients. Ces derniers sont ainsi utilisés pour le calcul des directions de descente à chaque itération. Notons également que l'approche proposée n'est pas purement fréquentielle. En effet, elle permet une paramétrisation à la fois de la matrice des fonctions de transferts et de l'espace d'état pour un correcteur inconnu. Ceci fait de notre approche un outil très flexible pour plusieurs situations d'intérêt pratique.

### 6.2.1. Problème de synthèse $H_\infty$

Rappelons le problème de synthèse  $H_\infty$  par retour de sortie statique.

**Problème 1** Considérant le système linéaire invariant (LTI) suivant

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ \omega \\ u \end{bmatrix} \quad (6-16)$$

où  $x \in \mathfrak{R}^n$  est le vecteur d'état,  $u \in \mathfrak{R}^m$  est le vecteur de commande,  $y \in \mathfrak{R}^{p_2}$  est le vecteur des sorties mesurées,  $\omega \in \mathfrak{R}^m$  est le vecteur des entrées exogènes et  $z \in \mathfrak{R}^{p_1}$  est le vecteur des sorties à commander.

Trouver le retour de sortie  $u = K y$  telle que la norme  $H_\infty$  de  $T_{\omega \rightarrow z}(p, K)$ , transfert en boucle fermée entre  $\omega$  et  $z$ , soit minimale sur l'ensemble de correcteurs  $K$  qui stabilise le système en boucle fermée donné par

$$\begin{bmatrix} \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} A + B_2 K C_2 & B_1 + B_2 K D_{21} \\ C_1 + D_{12} K C_2 & D_{11} + D_{12} K D_{21} \end{bmatrix} \begin{bmatrix} x \\ \omega \end{bmatrix} = \begin{bmatrix} A_{BF} & B_{BF} \\ C_{BF} & D_{BF} \end{bmatrix} \begin{bmatrix} x \\ \omega \end{bmatrix} \quad (6-17) \blacksquare$$

Le problème de synthèse  $H_\infty$  par retour de sortie dynamique peut être posé comme problème de synthèse  $H_\infty$  par retour de sortie statique pour un système augmenté. En effet, pour le système (6-16) et pour un correcteur dynamique d'ordre  $k \leq n$  de la forme

$$\begin{bmatrix} \dot{x}_k \\ u \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \begin{bmatrix} x_k \\ y \end{bmatrix} \quad (6-18)$$

où  $x_k \in \mathfrak{R}^k$  est le vecteur d'état du correcteur.

Le problème de synthèse  $H_\infty$  par retour de sortie dynamique est équivalent au problème 1 avec les substitutions suivantes

$$K \rightarrow \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}, A \rightarrow \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix}, B_1 \rightarrow \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, B_2 \rightarrow \begin{bmatrix} 0 & B_2 \\ I_k & 0 \end{bmatrix}, C_1 \rightarrow [C_1 \ 0], C_2 \rightarrow \begin{bmatrix} 0 & I_k \\ C_2 & 0 \end{bmatrix}, D_{11} \rightarrow D_{11}, \\ D_{12} \rightarrow [0 \ D_{12}], D_{21} \rightarrow \begin{bmatrix} 0 \\ D_{21} \end{bmatrix}.$$

$I_k$  et  $0_k$  dénotent la matrice d'identité et la matrice nulle de dimensions  $k \times k$  respectivement.

Notons que la le correcteur  $K \in \mathfrak{R}^{m_2 \times p_2}$  est remplacé dans ce cas par une matrice de dimension  $(k + m_2) \times (k + p_2)$ .

### 6.2.2. Problème de stabilisation robuste (optimisation du rayon de stabilité complexe)

Avant d'introduire ce problème, nous rappelons la notion préliminaire de rayon de stabilité complexe. Pour une matrice carrée complexe  $X \in \mathbb{C}^n$ , le rayon de stabilité complexe  $\beta(X)$  est défini par [Hin86]

$$\beta(X) = \min \{ \sigma_{\max}(E) \mid E \in \mathbb{C}^n, \alpha(X + E) \geq 0 \} \quad (6-19)$$

où  $\alpha(\cdot)$  est l'opérateur abscisse spectrale et  $\sigma_{\max}(E)$  la valeur singulière maximale de la matrice  $E$ .

Le rayon de stabilité complexe est nulle si et seulement si le  $X$  est instable. Dans le cas stable, il représente un indice de robustesse de la stabilité de la matrice  $X$  vis-à-vis à une perturbation additive  $E$  : pour toute matrice  $X$  stable,  $\beta(X)$  mesure la distance entre cette matrice  $X$  et les matrices instables de même dimension.

Le problème de stabilisation robuste peut être annoncé comme suit :

**Problème 2** Considérant le système linéaire invariant (LTI) suivant

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (6-20)$$

où  $x \in \mathfrak{R}^n$ ,  $u \in \mathfrak{R}^m$  et  $y \in \mathfrak{R}^p$  sont, respectivement, l'état, la commande et la sortie du système.

Trouver la commande par retour de sortie statique  $u = Ky$  qui maximise le rayon de stabilité complexe de la matrice du système en boucle fermée  $A + BKC$  :

$$\max_{K \in \mathfrak{R}^{m \times p}} \beta(A + BKC) \quad (6-21) \blacksquare$$

Ce problème est un cas particulier du problème 1, car comme nous venons de le voir si  $X$  est une matrice stable à qui on associe une fonction de transfert  $H(p) = (pI - X)^{-1}$ , alors  $\beta(X)$  est l'inverse de la norme  $H_\infty$  du transfert  $H$

$$\beta(X) = \|H(p)\|^{-1} \quad (6-22)$$

Ainsi, le problème 2 est équivalent à la minimisation de la norme  $H_\infty$  de la fonction de transfert en boucle fermée  $(pI - (A + BKC))^{-1}$  sous la contrainte que  $A + BKC$  soit stable.

En choisissant  $A = A$ ,  $B_1 = I$ ,  $B_2 = B$ ,  $C_1 = I$ ,  $C_2 = C$ ,  $D_{11} = 0$ ,  $D_{12} = 0$  et  $D_{21} = 0$ , nous pouvons facilement montrer que le problème 1 se réduit à un problème 2.

En utilisant les notations du problème générique 1, nous définissons la fonction coût

$$f(K) = \begin{cases} \|T_{\omega \rightarrow z}(p, K)\|_\infty & \text{si } \alpha(A + B_2KC_2) < 0 \\ \alpha(A + B_2KC_2) & \text{si } \alpha(A + B_2KC_2) \geq 0 \end{cases} \quad (6-23)$$

Le problème 1 peut être reformulé en un problème d'optimisation sans contrainte

$$\min_{K \in \mathbb{R}^{n_1 \times n_2}} f(K) \quad (6-24)$$

Cette fonction coût présente une discontinuité à  $\alpha(A + B_2KC_2) = 0$ . En effet, quand  $\alpha(A + B_2KC_2) \rightarrow 0^+$ , la valeur de  $\|T_{\omega \rightarrow z}(p, K)\|_\infty$  tend vers une valeur infinie alors que  $\alpha(A + B_2KC_2)$  tend vers 0. Cette discontinuité risque de compliquer davantage la minimisation de la fonction coût qui contient déjà des opérateurs non différentiables (norme  $H_\infty$  et abscisse spectrale). Il serait donc judicieux de modifier cette fonction en remplaçant la minimisation de  $\|T_{\omega \rightarrow z}(p, K)\|_\infty$  par la minimisation équivalente de  $-\|T_{\omega \rightarrow z}(p, K)\|_\infty^{-1}$ . Cette dernière assure bien la continuité à  $\alpha(A + B_2KC_2) = 0$ . La nouvelle fonction coût proposée est donnée par :

$$f(K) = \begin{cases} -\|T_{\omega \rightarrow z}(p, K)\|_\infty^{-1} & \text{si } \alpha(A + B_2KC_2) < 0 \\ \alpha(A + B_2KC_2) & \text{si } \alpha(A + B_2KC_2) \geq 0 \end{cases} \quad (6-25)$$

Dans le cas où le problème traité est celui de la stabilisation robuste (problème 2), la fonction coût  $f(K)$  est équivalente à

$$f(K) = \begin{cases} -\beta(A + B_2KC_2) & \text{si } \alpha(A + B_2KC_2) < 0 \\ \alpha(A + B_2KC_2) & \text{si } \alpha(A + B_2KC_2) \geq 0 \end{cases} \quad (6-26)$$

Toutefois, le nouveau problème formulé ne diffère pas des approches de synthèse  $H_\infty$  que nous avons précitées : il est global, non convexe et même non différentiable. Son unique avantage consiste dans le non recours à des variables de Lyapunov ce qui procure à nos problèmes d'optimisation formulés une dimension modérée.

### Remarque :

En s'inspirant du problème de stabilisation pure à base de la norme  $H_\infty$  décalée, le problème (6-25) peut être traité via une approche utilisant les normes  $H_\infty$ . Le critère à minimiser sera dans ce cas :

$$g(K) = \begin{cases} -\|T_{\omega \rightarrow z}(p, K)\|_\infty^{-1} & \text{si } \alpha(A + B_2KC_2) < 0 \\ \|C_{BF} [pI - A_{BF}]^{-1} B_{BF} + D_{BF}\|_{-a, \infty} & \text{si } \alpha(A + B_2KC_2) \geq 0 \end{cases} \quad (6-27)$$

Dans ce qui suit, nous avons choisi de présenter les résultats numériques de la première formulation (minimisation de la fonction  $f(K)$ ) sur quelques problèmes tirés de la littérature. Nous ne détaillerons

pas tous les problèmes qui pourront être retrouvés facilement dans la littérature. Nous présentons et analysons les résultats numériques afin de montrer l'applicabilité et l'habileté de notre approche pour résoudre des problèmes complexes.

Les normes infinies sont calculées en utilisant la fonction "hinfnorm". Les gradients de la fonction coût sont calculés à base des vecteurs droite et gauche de la décomposition en valeurs singulières de la matrice d'état en boucle fermée. Le correcteur  $K$  est initialisé à zéro ( $K = 0$ ).

### Stabilisation robuste d'un turbogénérateur (TG1)

Le modèle de cette application est tiré de la librairie *COMpleib* [Lei03], il a été initialement proposé dans [Hun82]. Ce modèle linéaire de 10 états est multivariable (2 entrées et 2 sorties) décrit la dynamique d'un turbogénérateur à énergie nucléaire de type 1072 MVA. Il est caractérisé par un rayon de stabilité complexe en boucle ouverte  $\beta(A) = 0.00767$ .

Le but de cet exemple est de trouver la matrice  $K$  qui permet de minimiser le rayon de stabilité complexe de la boucle fermée  $\beta(A + BKC)$ .

Les résultats d'optimisations issus de l'algorithme AGU sont résumés dans le tableau suivant :

Itération	$\beta$	Temps (s)
1	1.52e-2	1.31
4	6.89e-2	2.02
7	7.19e-2	3.74
12	7.82e-2	4.18
19	7.84e-2	7.37
27	7.87e-2	13.10

Fig. 6.9 – Résultats de stabilisation robuste ( $k = 0$ ) pour le système TG1

Le meilleur rayon de stabilité complexe achevé est  $\beta(A + BKC) = 0.0787$ . Il sensiblement meilleur que celui de la boucle ouverte  $\beta(A)$ . Le meilleur correcteur obtenu est

$$K = \begin{bmatrix} -0.99935 & -1.08070 \\ -0.09941 & -0.15920 \end{bmatrix} \quad (6-28)$$

Ce résultat est aussi très proche de celui trouvé dans [Bur03]. La solution atteinte dans cette étude est donnée par :

$$K = \begin{bmatrix} -0.7763 & -0.7193 \\ -0.0935 & -0.1515 \end{bmatrix} \quad (6-29)$$

avec  $\beta(A + BKC) = 0.0785$ .

### Stabilisation robuste d'un Boeing 767 (AC10)

Le présent problème traite la stabilisation robuste d'un avion de type Boeing 767 en conditions de vol définies dans [Dav90]. Le modèle aéroélastique de ce système est nommé AC10 dans la collection *COMpleib*. Ce modèle déjà étudié pour une stabilisation pure (cf. section 6.1.1) comporte 55 états, deux entrées et deux sorties.

En boucle ouverte, ce système est instable. Nous essayerons, dans cette étude, de concevoir un correcteur assurant la stabilité la plus robuste possible.

Dans [Bur03], la stabilisation robuste a été traitée pour le cas de correcteurs statiques ( $k = 0$ ) et dynamiques d'ordre faible ( $k = 1$  et  $k = 2$ ). Nous allons montrer que l'algorithme AGU que nous avons développé est capable d'atteindre de résultats meilleurs par rapport à ceux exposés dans [Bur03]. Ces résultats ont été jusqu'à ici les meilleurs obtenus sur ce système.

Pour  $k = 0$ , les résultats sont donnés par le tableau 6.10.

Itération	$\beta$	Temps (s)
1	9.09e-6	1.11
5	7.04e-5	3.25
12	9.20e-5	7.34
34	9.23e-5	19.23
54	9.32e-5	55.17
79	9.33e-5	102.34

Fig. 6.10 – Résultats de stabilisation robuste ( $k = 0$ ) pour le système AC10

Le rayon de stabilité achevé par l'algorithme AGU est donné par  $\beta(A + BKC) = 9.33 \cdot 10^{-5}$  avec

$$K = \begin{bmatrix} -1.1179 & -1.6880 \cdot 10^{-5} \\ 4.3558 \cdot 10^1 & 1.9930 \cdot 10^{-4} \end{bmatrix} \quad (6-30)$$

En comparant avec l'étude [Bur03] où le correcteur solution produisait un rayon de stabilité  $\beta(A + BKC) = 7.91 \cdot 10^{-5}$ , nous concluons que notre correcteur est meilleur (18% d'amélioration).

Pour  $k = 1$ , les résultats sont donnés par le tableau suivant.

Itération	$\beta$	Temps (s)
1	3.05e-6	1.52
4	1.22e-4	6.40
10	1.24e-4	20.88

Fig. 6.11 – Résultats de stabilisation robuste ( $k = 1$ ) pour le système AC10

Le rayon de stabilité obtenu par l'algorithme AGU est donné par  $\beta(A + BKC) = 1.24 \cdot 10^{-4}$  avec

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} -2.0566 \cdot 10^{-1} & -1.2201 \cdot 10^2 & 5.6638 \cdot 10^{-3} \\ -1.5569 \cdot 10^{-2} & -5.2806 \cdot 10^{-1} & 2.5267 \cdot 10^{-6} \\ -1.4663 \cdot 10^{-2} & 5.8900 & 3.9470 \cdot 10^{-5} \end{bmatrix} \quad (6-31)$$

De même, pour cette structure du correcteur, notre résultat reste meilleur puisque le rayon de stabilité optimal réalisé dans [Bur03] est de  $9.89 \cdot 10^{-5}$ .

Pour  $k = 2$ , les résultats sont donnés par le tableau 6.12.

Itération	$\beta$	Temps (s)
1	2.43e-5	2.43
6	1.86e-4	13.07
13	1.99e-4	22.81
19	2.01e-4	147.44

Fig. 6.12 – Résultats de stabilisation robuste ( $k = 2$ ) pour le système AC10



Le rayon de stabilité obtenu par l’algorithme AGU est donné par  $\beta(A+BKC) = 2.01 \cdot 10^{-4}$  avec un correcteur  $K$  défini par le quadruple suivant :

$$\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} -4.1368 \cdot 10^1 & -4.6368 \cdot 10^1 & 4.7884 \cdot 10^1 & -1.3239 \cdot 10^{-2} \\ -3.7889 \cdot 10^1 & -4.2488 \cdot 10^1 & -3.3479 \cdot 10^1 & -8.5480 \cdot 10^{-3} \\ -1.3890 \cdot 10^{-1} & -2.1165 \cdot 10^{-1} & 1.1020 & -3.6489 \cdot 10^{-5} \\ -5.6430 \cdot 10^{-1} & -5.2074 \cdot 10^{-1} & 5.7206 \cdot 10^{-2} & -1.2360 \cdot 10^{-4} \end{bmatrix} \quad (6-32)$$

Ce correcteur est de loin plus robuste que celui obtenu dans [Bur03], le rayon de stabilité complexe est double.

### Synthèse $H_\infty$ d’un Boeing 767 (AC10)

Dans cette section, le système précédent est réétudié en vue d’une synthèse  $H_\infty$  d’ordre fixé. La formulation utilisée est celle définie par le problème d’optimisation (6-24) et la fonction coût (6-25).

Les résultats obtenus sont résumés comme suit :

Pour  $k = 0$ , la norme  $H_\infty$  minimale du transfert en boucle fermée  $T_{\omega \rightarrow z}$  est  $\|T_{\omega \rightarrow z}(p, K)\|_\infty = 13.11$  (cf. tableau 6.13).

Itération	$\ T_{\omega \rightarrow z}(p, K)\ _\infty$	Temps (s)
1	546.07	1.67
8	1.323	6.30
17	13.11	15.92

Fig. 6.13 – Résultats de synthèse  $H_\infty$  ( $k = 0$ ) pour le système AC10

Le correcteur correspondant est donné par :

$$K = \begin{bmatrix} -8.9180 \cdot 10^{-1} & 2.1578 \cdot 10^{-5} \\ 4.2989 & 2.0538 \cdot 10^{-4} \end{bmatrix} \quad (6-33)$$

Pour comparaison, le meilleur résultat obtenu dans la littérature pour ce problème est celui qui a été établi dans [Apk06]. Il très proche du notre ( $\|T_{\omega \rightarrow z}(p, K)\|_\infty = 13.1$ ).

Pour  $k = 1$ , les résultats sont récapitulés dans le tableau ci-dessous :

Itération	$\ T_{\omega \rightarrow z}(p, K)\ _\infty$	Temps (s)
1	7.33e+2	2.01
4	1.31e+1	8.17
9	1.20e+1	63.79
18	1.03e+1	121.20
24	1.01e+1	168.34

Fig. 6.14 – Résultats de synthèse  $H_\infty$  ( $k = 1$ ) pour le système AC10

La plus petite norme  $H_\infty$  obtenue est meilleure que celle trouvée dans [Apk06]. Elle est égale à  $\|T_{\omega \rightarrow z}(p, K)\|_\infty = 10.1$ .

Le correcteur optimal d'ordre 2 est donné par :

$$\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} -2.4176 & -2.6371 \cdot 10^{-1} & 2.6845 \cdot 10^{-4} \\ -2.9232 & 3.1731 \cdot 10^{-1} & 1.4342 \cdot 10^{-5} \\ -3.4393 \cdot 10^{-1} & 2.4001 \cdot 10^{-1} & 4.9838 \cdot 10^{-5} \end{bmatrix} \quad (6-34)$$

### Synthèse $H_\infty$ d'un avion de transport (AC8)

Ce système porte le label AC8 dans la librairie *COMpleib*. Il a été introduit par [Gan86]. Il modélise un avion de transport à une altitude de 35000 Pieds et à une vitesse de 0.8 Mach avec un centre de gravité en arrière de l'avion. Ce système multivariable présente neuf états, une commande et 5 sortie de mesure. Sa matrice d'état est instable.

Quelques résultats du problème d'optimisation sont donnés par le tableau suivant :

Itération	$\ T_{\omega \rightarrow z}(p, K)\ _\infty$	Temps (s)
1	13.2	1.31
8	2.13	5.23
21	2.04	23.01
33	2.02	31.45
45	2.01	53.09
57	2.01	126.47

Fig. 6.15 – Résultats de synthèse  $H_\infty$  ( $k=0$ ) pour le système AC8

Le minimum atteint est donné par  $\|T_{\omega \rightarrow z}(p, K)\|_\infty = 2.01$  et ceci pour un correcteur statique  $K = [1.3018 \quad -0.9867 \quad -1.4861 \quad 0.0631 \quad 1.4209]$ .

### Remarque :

La formulation à base de la norme  $H_\infty$  peut être exploitée pour résoudre d'autre type de problèmes en Automatique. Une des applications envisageables est la réduction des modèles.

Ce problème se pose souvent lorsqu'on souhaite asservir un système  $G(p)$  d'ordre élevé difficile à commander. Si nous supposons la décomposition du modèle  $G(p) = G_{inst}(p) + G_{stab}(p)$  en deux parties une stable et l'autre instable, alors nous pouvons considérer le problème de minimisation

$$\min_{\tilde{G}_{stab} \in \chi} \|G_{stab}(p) - \tilde{G}_{stab}(p)\| \quad (6-35)$$

où  $\tilde{G}_{stab}(p)$  est un transfert stable d'ordre réduit d'une classe  $\chi$  définissant sa structure.

Si la norme utilisée est celle de Hankel, une solution explicite  $\tilde{G}_{stab}(p)$  de ce problème est donnée dans [Glo84]. Cependant, il serait aussi intéressant d'utiliser la norme  $H_\infty$ , ce qui mènera à un problème similaire à ceux déjà traités dans ce manuscrit. Une fois la solution du problème (6-35) obtenue, le nouveau modèle  $\tilde{G}(p) = G_{inst}(p) + \tilde{G}_{stab}(p)$  devient moins difficile à manipuler tout en gardant des caractéristiques très proches du modèle original.

### 6.3. Asservissement PID d'un moteur à courant continu

Dans cette section, nous appliquons les outils de synthèse d'une loi de commande optimale présentés dans les chapitres précédents, sur le cas d'un asservissement de position d'un moteur à courant continu. Nous proposons de traiter un exemple simple d'une synthèse de loi de commande PID mais suffisant pour montrer l'apport de l'approche par optimisation non linéaire proposée pour les applications de type « réglage rapide ». Les résultats de cette application sont validés expérimentalement sur un banc d'essai.

#### 6.3.1. Présentation du système

Souvent posé dans les applications industrielles, l'asservissement de position consiste à positionner une charge mécanique, avec une précision et une rapidité données. Des exemples à cette application seront : le braquage des gouvernes d'un avion, l'ouverture (ou la fermeture) d'une vanne, le positionnement d'un axe de machine à outil.

Pour illustrer la classe de ces problèmes, on se propose ici d'asservir la position rectiligne  $x$  d'un curseur à une grandeur électrique de référence (cf. figure 6.9).

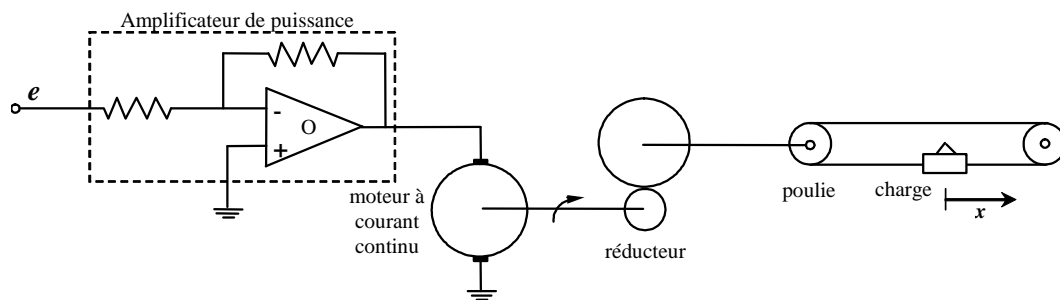


Fig. 6.9 – Dispositif du moteur à courant continu

Le curseur, solidaire d'une courroie, est déplacé grâce à un moteur à courant continu via une poulie et un réducteur. Le rôle de la poulie est de transformer en déplacement rectiligne le mouvement de rotation de l'arbre moteur, celui du réducteur est d'une part de faciliter l'obtention de la précision requise dans le positionnement du curseur, d'autre part de ramener au niveau de l'arbre moteur une inertie plus faible.

Afin de positionner convenablement le curseur, on dispose de deux sources d'information : un capteur de la position rectiligne du curseur et une génératrice tachymétrique qui fournit une tension proportionnelle à la vitesse de rotation du moteur. Le schéma fonctionnel du système est donné par la figure ci-dessous.

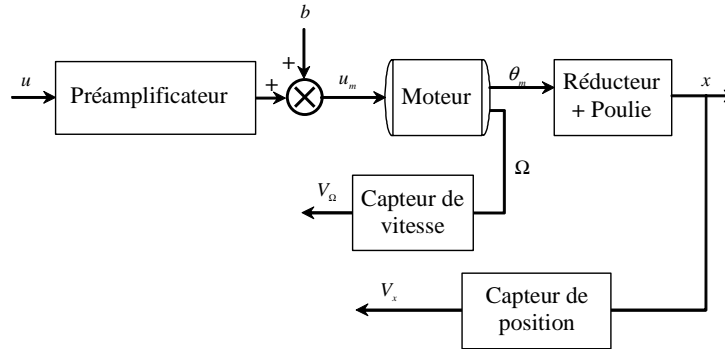


Fig. 6.10 –Schéma fonctionnel du système à commander

La tension de décalage  $b$  en amont du moteur permet de simuler une perturbation appliquée au système (offset en sortie de l’amplificateur de puissance ou couple perturbateur).

Les différents éléments de ce système sont modélisés comme suit :

- Le préamplificateur est assimilé à un gain  $K_A$  ;
- Le moteur est modélisé par le transfert  $K_M/p(1+\tau p)$  pour le modèle de position angulaire;
- L’ensemble (Réducteur+Poulie) est équivalent à un gain  $\rho/N$  ;
- Le capteur de position (potentiomètre) est identifié à un gain  $K_x$  ;
- Le capteur de vitesse (génératrice tachymétrique) est modélisé par un gain  $K_\Omega$ .

Afin d’identifier les différents paramètres du modèle, nous avons adopté une approche “boite noire” fréquentielle. Nous avons ainsi utilisé une solution logicielle graphique qui a été développée sous MATLAB, en employant la boîte à outils Real Time Windows Target. Cette dernière permet d’effectuer une analyse harmonique de la boucle ouverte en position et puis en vitesse pour une gamme de fréquences allant de 3 Hz à 30 Hz. Les tracés fréquentiels expérimentaux et paramétriques des transferts position et vitesse obtenus sont donnés par les figures ci-dessous :

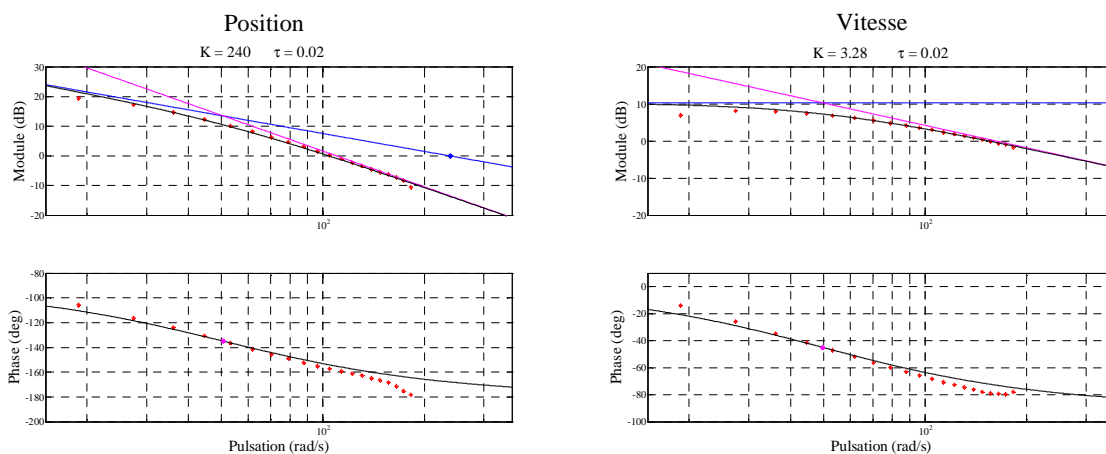


Fig. 6.11 –Résultats de l’identification fréquentielle du modèle en position et en vitesse

Les modèles paramétriques résultants permettent d’estimer les principaux paramètres du modèle. Ces derniers sont résumés par le tableau 6.16 :

Paramètre	Valeur (Unité)
$K_A$	10
$K_M$	40 (rad/s/V)
$\tau$	0,02 (s)
$K_\theta = K_x \rho / N$	0,6 (V/rad)
$K_\Omega$	$8,2 \cdot 10^{-3}$ (V/rad/s)

Tab. 6.16: Paramètres du modèle

### 6.3.2. Cahier des charges

Contrairement à la plus part des méthodes classiques de synthèse de commande linéaire qui se basent sur une formulation totalement fréquentielle des cahiers des charges, nous proposons dans cette application de traiter les spécifications temporelles sans aucune formulation équivalente. Dans un premier temps, nous considérons un asservissement de position par un correcteur PID avec une action dérivée filtrée (cf. figure 6.12).

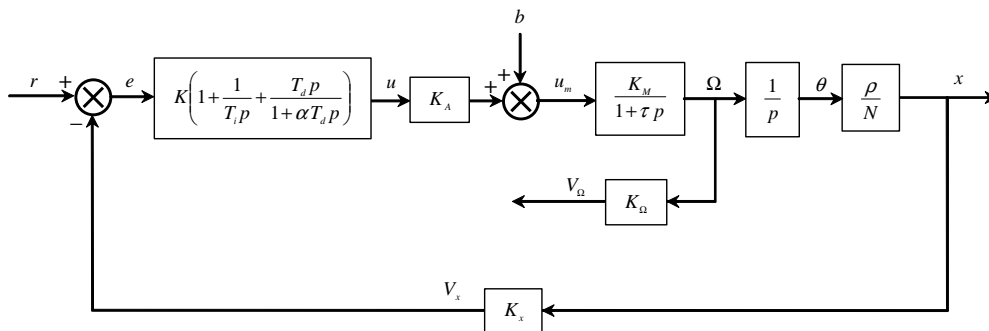


Fig. 6.12 – Schéma fonctionnel de l'asservissement PID de position

Le but de cette boucle d'asservissement est de respecter les spécifications suivantes :

Pour une entrée de référence  $r$  de type échelon :

- Une erreur statique nulle ;
- Un dépassement maximal de 20% ;
- Un temps de réponse à 50% inférieur à 0,006(s) ;
- Un temps d'établissement à 5% inférieur à 0,012(s) ;
- Un temps d'établissement à 2% inférieur à 0,020(s) ;
- Une amplitude de commande maximale de 5(V) ;
- Un rejet des perturbations  $b$  constantes ;
- Une marge de phase supérieure à  $60^\circ$  .

### 6.3.3. Formulation du problème

Tout d'abord, nous constatons que la structure du système de commande permet déjà de répondre à quelques exigences du cahier des charges. En effet, le terme intégral dans le transfert du système à

commander garantit une erreur statique nulle. Tandis que l'action intégrale du correcteur PID assure un rejet des perturbations  $b$  constantes.

Afin de traiter le reste des exigences temporelles du cahier des charges, nous avons choisi d'utiliser la formulation sous forme de gabarits. Les paramètres d'optimisation sont ceux du correcteur  $\theta = (K, T_i, T_d, \alpha)^T$  et le problème de faisabilité des spécifications temporelles est traduit par la minimisation du critère (somme pondérée de pénalisations exactes) suivant :

$$J_1(\theta) = \sum_{i=1}^{n_t} \left( \eta_{11} |y(\theta, t_i) - \bar{y}(t_i)|_+ + \eta_{12} |y(t_i) - y(\theta, t_i)|_+ + \eta_{21} |u(\theta, t_i) - \bar{u}(t_i)|_+ + \eta_{22} |u(t_i) - u(\theta, t_i)|_+ \right) \quad (6-36)$$

avec :  $|f|_+ = \max(f, 0)$  et  $\eta$  des pondérations.

La contrainte fréquentielle est également prise en compte dans le critère par l'ajout d'un terme de pénalisation exacte :

$$J(\theta) = J_1(\theta) + \eta_3 |\Delta\phi - \Delta\phi(\theta)|_+ \quad (6-37)$$

Ce critère mixte (temporel et fréquentiel) est minimisé par rapport au vecteur de paramètres  $\theta$  et le cahier des charges sera faisable si et seulement si le coût optimal atteint  $\bar{J}$  est nul.

### 6.3.4. Résolution, simulation et validation expérimentale

L'évaluation du critère (6-37) est partiellement basée sur le calcul des trajectoires E/S du système. Ces dernières sont estimées via une résolution différentielle du système en boucle fermée. La méthode utilisée, dans ce cas, est celle de Runge-Kutta d'ordre 4 à pas fixe (égal à  $10^{-4}$ ). Elle est implémentée sur Matlab sous la fonction "ode45".

Pour l'algorithme AGU, l'évaluation du gradient de la partie  $J_1$  du critère est faite avec celle des trajectoires du système en boucle fermée : nous résolvons simultanément dans un même système augmenté les équations différentielles du système en boucle fermée et des fonctions de sensibilité correspondant aux paramètres du vecteur  $\theta$ .

Nous avons choisi de simuler les réponses temporelles du système sur une grande plage de temps comparant au temps de réponse demandé (10 fois plus grande). Ce choix est en effet crucial pour une bonne évaluation du critère (6-37). La réduction d'une telle plage peut compromettre l'évolution des algorithmes et donc leurs résultats.

La partie fréquentielle du critère est évaluée via une résolution numérique d'un problème polynomial (détermination de la pulsation de coupure  $\omega_c$ ). Ses variations sont ensuite estimées par la technique de dérivation complexe.

Les algorithmes testés pour ce problème de minimisation sont initialisés à partir d'un correcteur stable arbitrairement choisi  $\theta_0 = (0.1, 0.1, 0.6, 0.2)$ . Les performances de ces différents algorithmes d'optimisation sont résumées par le tableau ci-dessous.

Algorithme	fminsearch	ASM	fminunc	AGU
Critère $\bar{J}$	41.9256	0	2.0540e-02	0
Solution $\bar{\theta}$	(0.1983, 0.0950, 0.1106, 0.0402)	(0.5850, 0.2594, 0.0239, 0.1349)	(0.6418, 0.4416, 0.0220, 0.1471)	(0.6036, 0.3052, 0.0227, 0.1402)
Nb itérations	179	135	55	41
Nb éval. $J$	308	256	402	68
Nb éval. $\nabla J(x)$	0	0	55	240
Temps de calcul (s)	330.6050	58.5040	444.7090	57.1030

Tab. 6.17: Résultats des algorithmes d’optimisation testés

Dans ce tableau, nous comparons les résultats obtenus par les algorithmes ASM et AGU avec les fonctions Matlab “fminsearch” et “fminunc” respectivement. Ces deux fonctions ont été déjà introduites dans le chapitre 4 sous les noms algorithme 4 et algorithme 7.

Les résultats montrent que seuls les algorithmes que nous avons développés permettent d’établir la faisabilité du cahier des charges exigé. La performance des algorithmes ASM et AGU est nettement établie par les courbes de décroissance du critère de la figure 6.13.

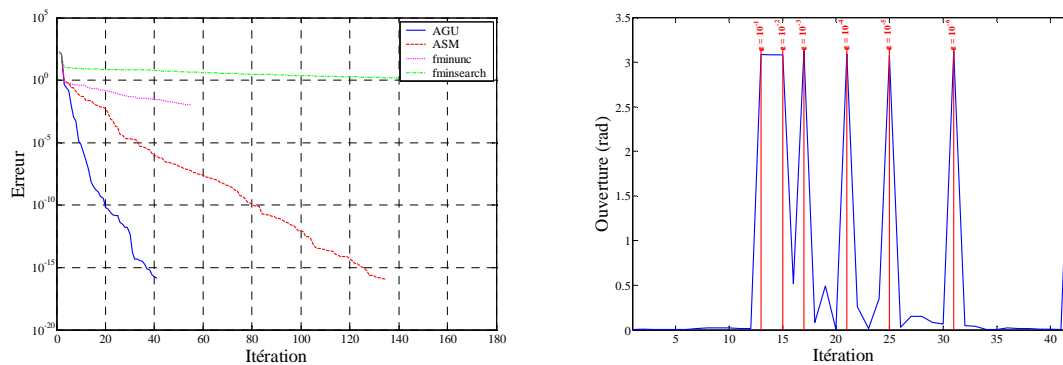
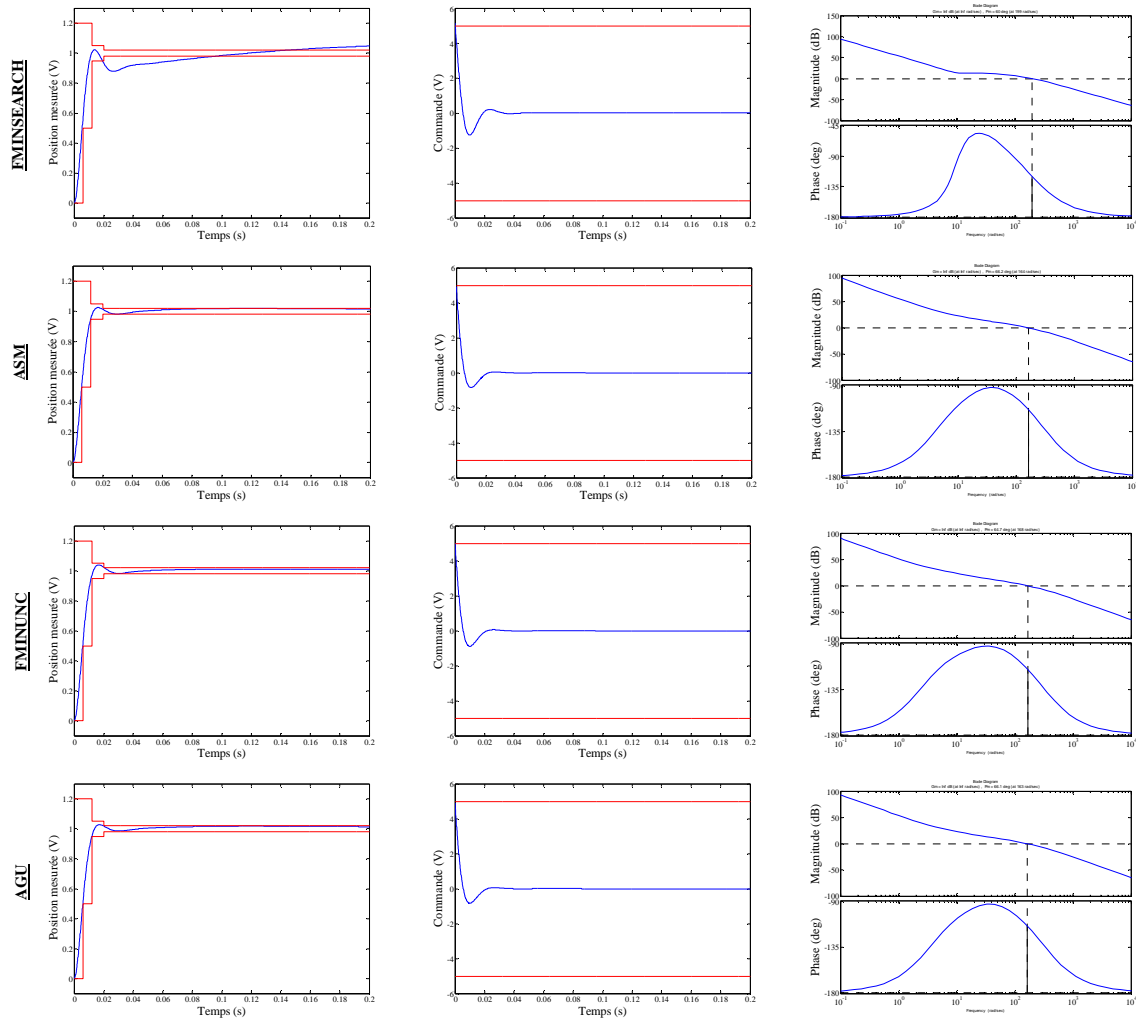


Fig. 6.13 – Evolution des algorithmes testés

Les deux autres algorithmes convergent au bout de quelques itérations vers des régions “difficiles” et où leur évolution est vraiment ralentie. Ces régions de l’espace paramétrique correspondent à des zones non différentiables vu la structure multi-objectif du problème d’optimisation traité. La mesure de la dispersion des gradients du critère à minimiser dans un voisinage de l’espace paramétrique permet de discerner facilement ces régions.

En effet, l’évolution de l’angle maximal entre ces différents gradients permet de détecter les régions du critère où il est difficile de déterminer une direction de descente. Ces zones sont atteintes lorsque l’ouverture (l’angle maximal entre les différents gradients d’une même itération) s’éloigne de la valeur 0 degré. Dans le cas où cet angle avoisine 180°, ceci correspond à une  $\varepsilon$ -stationnarité du critère pour l’algorithme AGU et donc à un passage vers une nouvelle itération où le rayon  $\varepsilon$  est réduit (cf. figure 6.13).

Pour chaque correcteur optimal, les réponses temporelles et fréquentielles des systèmes commandés sont donnés par la figure 6.14.



**Fig. 6.14 – Résultats de simulation par les correcteurs PID optimaux**

Ces courbes montrent que l'exigence fréquentielle de marge de phase est largement remplie par tous les correcteurs obtenus. Cependant, seuls les algorithmes ASM et AGU réussissent à réaliser simultanément toutes les exigences du cahier des charges.

Nous observons aussi qu'il existe deux contraintes plus difficiles à satisfaire : le temps d'établissement à 2% et à l'amplitude du signal de commande. D'ailleurs, lors de l'optimisation, nous constatons que c'est sur ces deux exigences que la fonction d'optimisation "fminunc" bute définitivement.

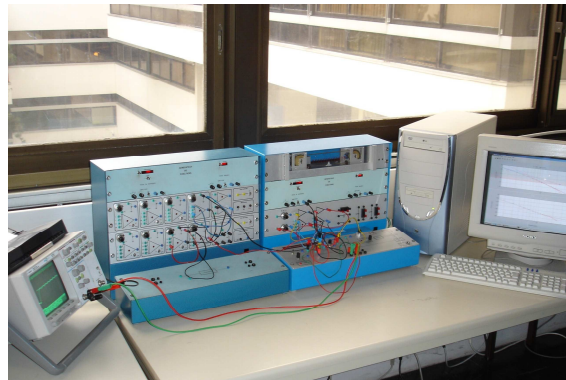
Nous notons aussi que les correcteurs optimaux obtenus pour les algorithmes ASM et AGU sont légèrement différents. Ceci est particulièrement dû à la différence de leurs stratégies de descente. Le tableau ci-dessous fait une synthèse de spécifications exigées par le cahier des charges et obtenues par ces deux algorithmes.

Spécification	Exigées	ASM	AGU
Dépassement	< 20 %	3.360 %	2.760 %
Temps de montée à 50%	< 6 ms	5.725 ms	5.835 ms
Temps d'établissement à 5%	< 12 ms	11.500 ms	11.900 ms
Temps d'établissement à 2%	< 20 ms	19.990 ms	19.890 ms
$\ u(t)\ _{\infty}$	< 5 V	4.989 V	4.910 V
$\Delta\phi$	> 60°	66.2 °	66.1°

*Tab. 6.18: Comparaison des performances réalisées*

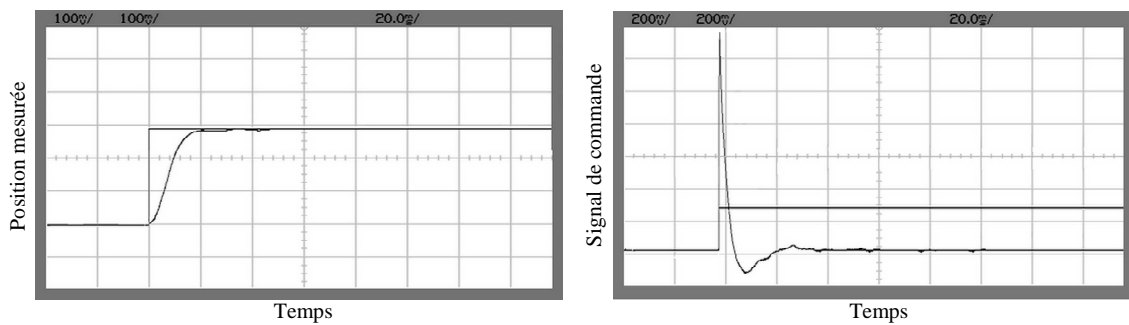


L'implémentation expérimentale des correcteurs PID optimaux obtenus avec les algorithmes ASM et AGU a été effectuée analogiquement sur le banc d'essai disponible au Département Automatique de Supélec (cf. figure 6.15).



**Fig. 6.15 –Banc expérimental de l'asservissement de position**

Les paramètres des deux correcteurs PID optimaux ne sont pas trop différents, c'est pourquoi les réponses temporelles sont très similaires. Nous nous contentons ici de présenter celles obtenues par l'algorithme AGU.



**Fig. 6.16 –Réponses temporelles expérimentales**

Même si les paramètres optimaux du correcteur n'ont pas été scrupuleusement respectés lors de l'implémentation (faute de composants bien calibrés), les relevés temporels présentent une grande concordance avec les résultats de simulation : mise à part une légère différence dans la première oscillation, l'ordre de grandeur du temps de réponse et des amplitudes est bien respecté.

### 6.3.5. Minimisation du temps de réponse

Nous considérons à présent le problème de minimisation du temps de réponse (temps d'établissement à 5%) du système en boucle fermée sous les mêmes conditions du cahier des charges précédent. Il s'agit de déterminer la limite de rapidité que le système peut atteindre avec une telle structure de correcteur (PID avec action dérivée filtrée).

Pour ce faire, nous relâchons la contrainte sur l'amplitude du signal de commande jusqu'à 10 volts (maximum possible sur la maquette) et nous proposons de minimiser le critère suivant :

$$J(\theta) = T_r + \sum_{i=1}^n \left( \eta_{11} |y(\theta, t_i) - \bar{y}(t_i)|_+ + \eta_{12} |y(t_i) - y(\theta, t_i)|_+ + \eta_{21} |u(\theta, t_i) - \bar{u}(t_i)|_+ + \eta_{22} |u(t_i) - u(\theta, t_i)|_+ \right) + \eta_3 |\Delta\phi - \Delta\phi(\theta)| \quad (6-38)$$

Dans ce critère, les formules des gabarits  $\bar{y}$  et  $\underline{y}$  ne tiennent pas compte l’ancienne spécification du temps de réponses.

Les différents termes de ce critère sont évalués selon les mêmes techniques déjà précitées à l’exception du terme  $T_r$  qui nous estimons numériquement à base des estimées du signal de sortie  $y$ . La variation de ce terme est aussi estimée par la technique de dérivation complexe.

Les résultats d’optimisation obtenus avec les algorithmes ASM et AGU sont donnés par les tableaux ci-dessous :

Algorithme	ASM	AGU
Critère $\bar{f}$	7.996 e-03	7.924e-03
Solution $\bar{x}$	(0.9117, 0.1989, 0.0225, 0.1016)	(0.8699, 0.2140, 0.0243, 0.0962)
Nb itérations	43	19
Nb éval. $f$	98	28
Nb éval. $\nabla f(x)$	0	82
Temps de calcul (s)	128.5850	121.7910

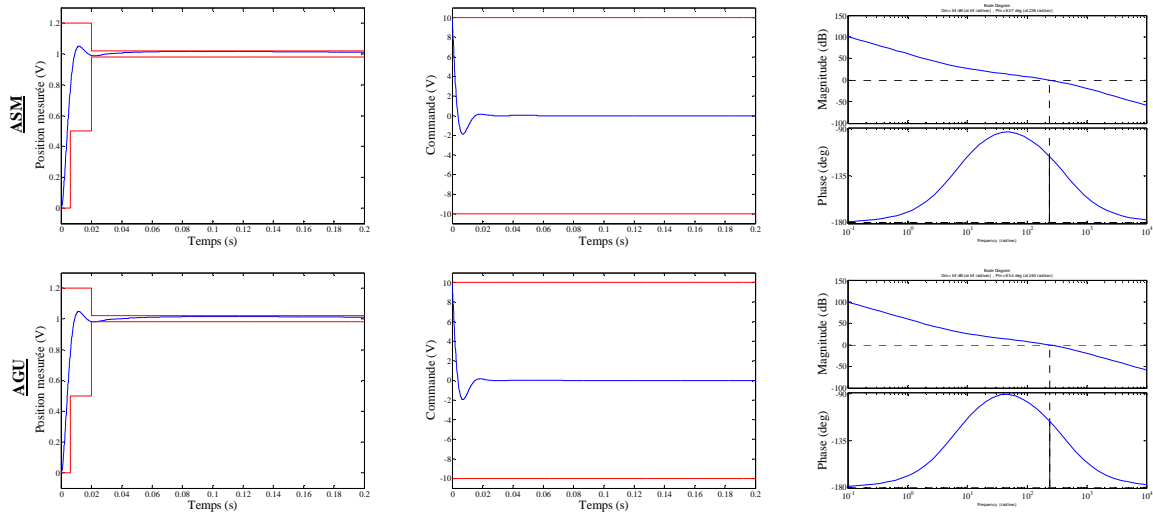
Tab. 6.19: Résultats des algorithmes d’optimisation

Spécification	Exigées	ASM	AGU
Dépassement	< 20 %	4.980 %	4.805 %
Temps de montée à 50%	< 6 ms	4.050 ms	4.030 ms
Temps d’établissement à 5%	à minimiser	7.996 ms	7.924 ms
Temps d’établissement à 2%	< 20 ms	15.530ms	15.000 ms
$\ u(t)\ _\infty$	< 10 V	9.886 V	9.907 V
$\Delta\phi$	> 60°	63.7 °	63.4 °

Tab. 6.20: Comparaison des performances réalisées

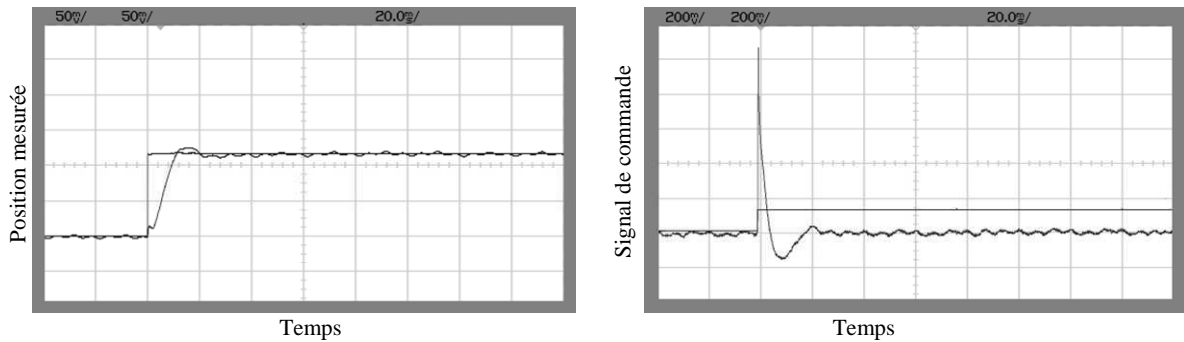
Ces deux algorithmes permettent d’atteindre un temps de réponse de l’ordre de 8 millisecondes toute en respectant les autres demandes du cahier des charges. Les performances du correcteur optimal de l’algorithme AGU sont légèrement meilleures. En réalité, cet algorithme est par construction plus adapté à des critères non différentiables (échantillonnage local), éventuellement avec des sauts (ce qui est une propriété du temps de réponse).

Les réponses temporelles et fréquentielles du système correspondant aux correcteurs optimaux sont données par la figure 6.17. Elles sont presque identiques.



**Fig. 6.17 – Résultats de simulation des correcteurs PID optimaux**

Les réponses temporelles de l'implémentation du correcteur PID optimal sont données par la figure ci-dessous :



*Fig. 6.18 – Réponses temporelles expérimentales*

Ces réponses temporelles présentent un régime oscillatoire autour de la nouvelle position de référence. Ces oscillations sont essentiellement dues à la saturation du préamplificateur de commande et aux dynamiques hautes fréquences qui n'ont pas été prises en compte lors de la modélisation du système. En effet, le modèle considéré dans cette étude n'est valable qu'au dessous de 180 (rad/s) (cf. figure 6.11), alors que les pulsations de coupure obtenues sont de l'ordre de 240 (rad/s).

De même, nous remarquons l'apparition d'un effet de phase non minimale sur la sortie, près de l'instant d'application de l'échelon. Ce phénomène est certainement dû aux dynamiques hautes fréquences qui n'ont pas été considérées dans notre modèle (en particulier l'élasticité d'une partie de la transmission).

### 6.3.6. Modification de la structure PID

Nous proposons dans cette section d'échanger la structure du correcteur PID à action dérivée filtrée par une nouvelle structure où la dérivée du signal erreur  $e$  est calculée à partir de la vitesse mesurée par la génératrice tachymétrique.

La nouvelle boucle fermée est alors composée de deux boucles en cascade qui permettent l’implantation d’une correction PID pure. Le schéma de cette boucle d’asservissement est donné par la figure suivante.

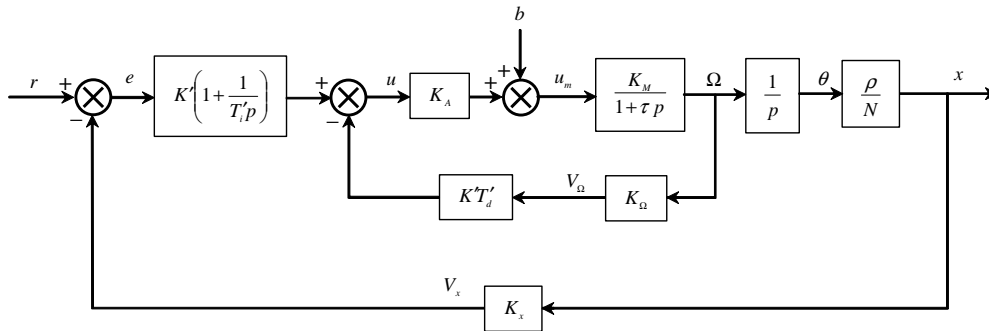


Fig. 6.19 –Schéma fonctionnel d’un asservissement de position en cascade

Le but est d’ajuster les anciens paramètres de commande optimaux  $\theta$  afin d’adapter cette nouvelle structure et d’évaluer ses performances.

Le nouveau gain dans la contre-réaction tachymétrique est donné par  $K'T'_d$  et le correcteur PID est défini par la loi de commande :

$$u(p) = K' \left( 1 + \frac{1}{T'_i p} \right) \left( r(p) - K_\theta \theta(p) \right) - K'T'_d K_\Omega p \theta(p) \quad (6-39)$$

En se basant sur le correcteur optimal obtenu par l’algorithme AGU (cf. tableau. 6.19) et en conservant les gains statiques des transferts, nous proposons d’initialiser les paramètres  $\theta' = (K', T'_i, T'_d)^T$  de la nouvelle structure PID comme suit :  $K'_0 = K_{opt}$ ,  $T'_{i,0} = T_{i,opt}$  et  $T'_{d,0} = T_{d,opt} K_\theta / K_\Omega$ .

Nous conservons le même critère (6-38) que nous minimisons en employant l’algorithme AGU. Les résultats de ce nouveau problème d’optimisation sont résumés par les deux tables suivantes.

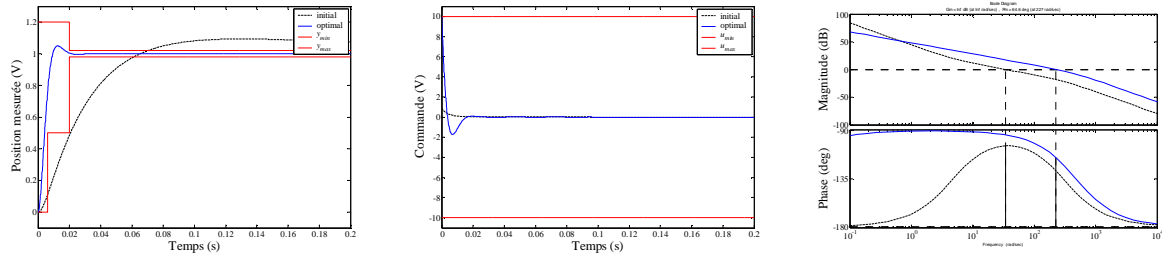
Algorithme	AGU
Critère $\bar{f}$	12.750e-3
Solution $\bar{x}$	(9.9999, 116.8487,
Nb itérations	14
Nb éval. $f$	43
Nb éval. $\nabla f(x)$	54
Temps de calcul (s)	71.4850

Tab. 6.21: Résultats d’optimisation

Spécification	Exigées	PID
Dépassement	< 20 %	5.02 %
Temps de réponse à 50%	< 6 ms	4.110 ms
Temps d’établissement à 5%	à minimiser	12.750 ms
Temps d’établissement à 2%	< 20 ms	17.300 ms
$\ u(t)\ _\infty$	< 10 V	9.999 V
$\Delta\phi$	> 60°	64.6°

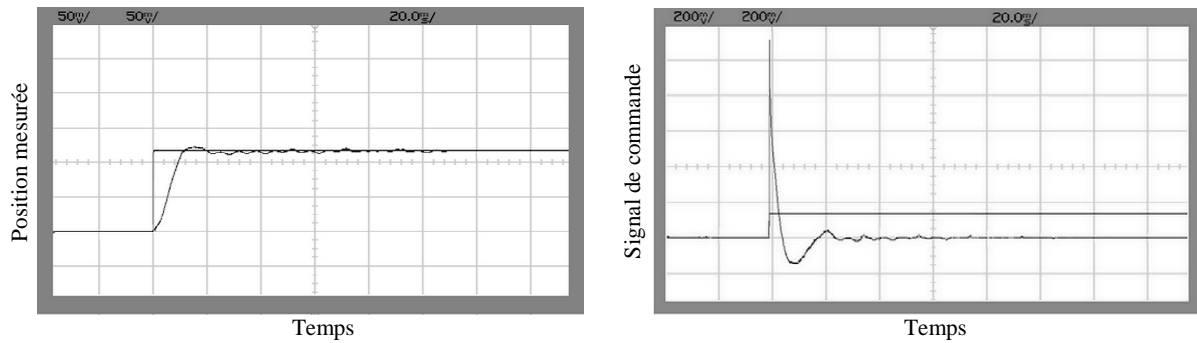
Tab. 6.22: Performances réalisées

Nous observons que le temps de réponse minimal obtenu pour ce correcteur PID est largement supérieur (12,750 ms) à celui d’un correcteur PID avec une action dérivée filtrée (7,924 ms). Les réponses temporelles et fréquentielles, avant et après la retouche, sont représentées sur la figure 6.20.



**Fig. 6.20 – Résultats de simulation du correcteur PID optimal**

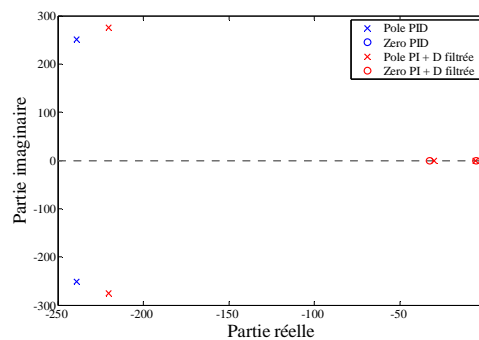
Ces résultats de simulation temporels ont été validés expérimentalement sur le banc d’essai. Les relevés de cette validation sont donnés par la figure ci-dessous.



*Fig. 6.21 – Réponses temporelles expérimentales*

Nous remarquons que les réponses expérimentales obtenues sont moins oscillatoires que les précédentes et qu’elles respectent parfaitement le cahier des charges fixé.

Le paramètre  $\alpha$  du filtre passe-bas de l’action dérivée apparaît donc comme un paramètre décisif qui permet d’atteindre des régimes en sortie très rapides avec de faibles efforts de commande. Le lieu des racines de la sortie de la boucle fermée pour les deux structures du correcteur PID (cf. figures 6.12 et 6.19) est donné pas la figure ci-dessous.



*Fig. 6.22 – Lieu des racines des boucles fermées*

Pour les deux structures de commande PID optimales, les pôles dominants sont compensés par des zéros. La boucle de sortie correspondante au correcteur PID en cascade se réduit alors à un système de second ordre rapide bien amorti, tandis que celle du correcteur PID filtré est ramenée grâce au paramètre  $\alpha$  à un système de troisième ordre avec un zéro stable proche du pôle réel. Ce transfert en

boucle fermée est plus riche en dynamiques : il permet d’atteindre des régimes oscillatoires plus adaptées aux gabarits temporels et d’améliorer ainsi le temps de réponse du système (cf. figures 6.17 et 6.20). D’autres parts, la paire des pôles complexes conjugués dans le cas du correcteur PID avec action dérivée filtrée est plus à droite que celle du correcteur PID en cascade. Ceci implique une amplitude de commande moins importante.

## 6.4. Stabilisation robuste d’un système oscillatoire

Le but de cet exemple est de montrer la difficulté à laquelle un ingénieur automatique peut être confronté pour asservir un simple système oscillatoire avec un correcteur PI.

### 6.4.1. Présentation du problème

Considérant le modèle linéaire suivant, proposé et étudié par Åström et al. dans [Ast98].

$$G(p) = \frac{9}{(p+1)(p^2 + \alpha p + 9)} \quad (6-40)$$

Ce système benchmark possède deux pôles complexes conjugués ayant un petit coefficient d’amortissement  $\xi = \alpha/6$  engendrant un régime oscillatoire. Quand le paramètre  $\alpha$  est diminué, le système devient très difficile à commander avec un contrôleur PI [Ast98].

Nous proposons dans cette application d’utiliser l’approche par optimisation non linéaire non différentiable, décrite dans les chapitres précédents, pour ajuster les paramètres du correcteur PI afin d’atteindre de bonnes marges de robustesse en stabilité.

### 6.4.2. Cahier des charges

En considérant une correction PI en série et pour  $\alpha = 0.2$ , les spécifications de robustesse du système sont exprimées en fonctions des marges de stabilité classiques suivantes :

- Une marge de phase de  $40^\circ$  ;
- Une marge de module égale à 0.5 .

### 6.4.3. Formulation en un problème d’optimisation

Contrairement au cas précédent, nous proposons de formuler les spécifications fréquentielles par le critère quadratique d’égalité suivant :

$$J(\theta) = \eta_1 (\Delta\phi(\theta) - 40)^2 + \eta_2 (\Delta M(\theta) - 0.5)^2 \quad (6-41)$$

où  $\eta_1$  et  $\eta_2$  sont des pondérations.

En utilisant les outils de calcul polynomial disponibles sur le logiciel Mathematica, le calcul de ce critère fréquentiel  $J$  est fait formellement : nous calculons respectivement la pulsation de coupure  $\omega_c$  et la pulsation  $\omega_0$  correspondante au point du lieu de Nyquist le plus proche du point critique  $(-1,0)$ . En revanche, le gradient du critère est estimé par la méthode de dérivation complexe.

Nous utilisons le critère de Routh pour établir les conditions de stabilité du système en boucle fermée. Le tracé de l'espace paramétrique correspondant aux correcteurs PI stabilisants est donné par la figure 6.23.

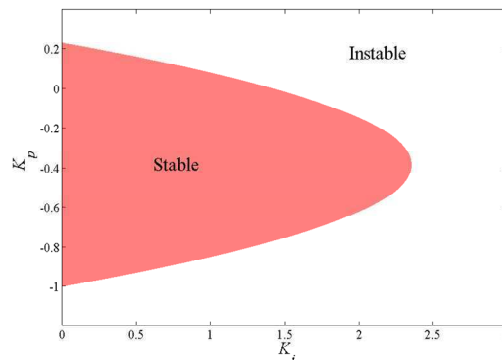


Fig. 6.23 –Espace paramétrique stable

Cet espace paramétrique est utilisé conjointement avec les règles temporelles d'autoréglage de Ziegler-Nichols [Gar04] pour amorcer les algorithmes d'optimisation avec un bon correcteur initial stable. Ce dernier est donné par :  $K_0(p) = k_p + k_i/p = -0.1 + 0.1/p$ .

Par la suite, nous employons les algorithmes ASM et AGU pour minimiser le critère (6-41). Le contrôleur PI optimal obtenu est donné par :  $K(p) = -0.2076 + 0.9459/p$ .

Les performances des deux algorithmes d'optimisation ne sortent pas des observations préétablies : leurs résultats sont similaires à ceux des problèmes déjà traités. Nous présentons et analysons maintenant les résultats de simulation obtenus.

Dans la figure 6.24, le lieu de Nyquist de la boucle ouverte et les réponses temporelles du système en boucle fermée sont représentées.

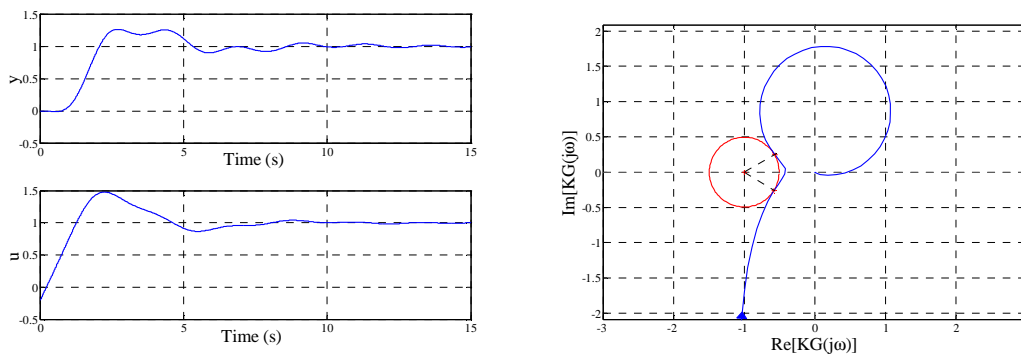


Fig. 6.24 –Réponses temporelles et fréquentielles du système

Bien que le critère à minimiser ait une structure quadratique, il n'est pas lisse (différentiable) partout. Ceci est essentiellement dû à la non différentiabilité des spécifications elles-mêmes ( $\Delta\phi$  et  $\Delta M$ ) qui peuvent présenter variations brusques sous forme de sauts. Ces dernières surgissent suite à l'apparition ou la disparition des racines lors de la résolution du problème polynomial correspondant. Dans notre cas, le minimum est atteint simultanément à deux pulsations comme le montre le tracé de Nyquist.

La figure 6.25 montre les courbes de niveau du critère (6-41) avec deux zones non différentiables. Le minimum obtenu par les algorithmes ASM et AGU se situe à l'intersection de ces deux zones.

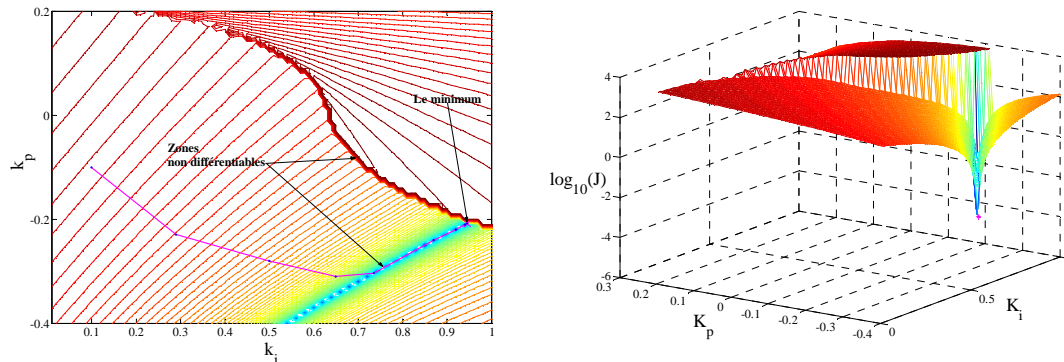


Fig. 6.25 – Courbes de niveau du critère montrant les zones non différentiables

Les différentes itérations de l'algorithme AGU sont représentées sur le graphe des courbes de niveau du critère  $J$ . Nous observons que ces dernières convergent vers une des deux zones non différentiables avant de continuer son avancée dans la direction descendante vers l'intersection de ces deux zones.

## 6.5. Commande Backstepping d'une suspension magnétique

Les systèmes de suspension magnétique pour machines tournantes ont suscité une attention particulière ces dernières années. Suite à la baisse du prix des actionneurs et de l'électronique associée, ce système est de plus en plus employé dans des applications industrielles et plus seulement pour des applications spécifiques "hautes technologie". Ils offrent un certain nombre d'avantages pratiques telles une basse consommation d'énergie, une capacité pour le déplacement linéaire, une vitesse de rotation élevée ; ils peuvent également fonctionner à des températures extrêmes et leur durée de vie est très longue. De plus, l'absence de contacts mécaniques, qui sont présents dans les systèmes traditionnels élimine le problème de la lubrification et réduit ainsi le coût et la fréquence de maintenance.

Le système de suspension magnétique étudié est instable en boucle ouverte. Afin de garantir sa stabilité, une commande en boucle fermée est alors indispensable. Pour traiter le problème de stabilisation de tels systèmes, il est souvent utilisé, dans la littérature, le modèle linéarisé autour d'un point de fonctionnement. D'un point de vue pratique, les régulateurs classiques sont encore les plus largement utilisés, ceci est principalement dû à leur simplicité d'implémentation. En outre, dans de nombreuses applications, il est montré que la robustesse est relativement bonne. Des méthodes linéaires ont donc été employées pour commander ce système, soit sur la base de la représentation classique en fréquentiel, soit sur celle de la représentation en espace d'état [Mat90, Swa90, Tsu89]. La commande par les méthodes  $H_\infty$  et  $\mu$  synthèse ont également été utilisées [Fuj90, Non94].

Cependant, ces techniques linéaires limitent le fonctionnement de l'actionneur dans une petite plage de variation autour de la position désirée. Parmi les commandes non linéaires déjà appliquées, on cite la commande par modes glissants [Run96], la commande par Backstepping [Que96, Hen04], la



commande basée sur la passivité [Rod00a, Rod00b Cor02], la commande basée sur la platitude [Lev96], les techniques de commande floue [Wei94] et enfin dans [Tor98], une comparaison entre la commande basée sur la linéarisation par bouclage, la commande par Backstepping et celle basée sur la passivité a été effectuée.

Pour satisfaire les demandes industrielles qui exigent un rendement de plus en plus élevé et donc de meilleures performances, on peut appliquer des commandes (non linéaires) **prenant en compte les aspects non linéaires présents dans certains modèles**. Pour ce faire, nous nous proposons de traiter le problème avec la méthode de Backstepping (cf. annexe 1). Si cette méthode permet de définir systématiquement une structure de commande prenant en compte l'aspect non linéaire du modèle, elle ne permet pas de réglage des paramètres qu'elle introduit. Elle nécessite donc un complément, indispensable, permettant le réglage des paramètres en fonction d'un cahier des charges spécifié.

Nous allons montrer que la combinaison du Backstepping avec des méthodes de formulation et d'optimisation de cahier des charges ouvre la voie à d'intéressantes réalisations.

Notons enfin que la mise en œuvre de la commande a été faite sur le dispositif de suspension magnétique disponible dans notre laboratoire.

### 6.5.1. Modélisation du système

#### 6.5.1.1. Principe d'une suspension magnétique

Brièvement, le dispositif de suspension magnétique, représenté sur la figure 6.26 est composé d'un électroaimant fixe, alimenté par une source de courant variable et d'un pendule mobile, en fer aimanté.

Ce système de lévitation magnétique peut être comparé à un ressort mécanique, mettant donc en jeu des forces fonctions du déplacement. Cependant, contrairement au problème à une dimension du ressort, il faut tenir compte ici d'un système à symétrie cylindrique, où les forces magnétiques intervenant se décomposent en composantes radiales et axiales.

Le dispositif la figure 6.26 présente une symétrie radiale parfaite (au niveau des pièces et de la structure des aimants) et la réalisation physique est conçue pour conserver le pendule sur l'axe central de symétrie. De ce fait, on s'intéresse uniquement au comportement résultant des petits déplacements verticaux du pendule.

Outre la force de gravitation, le pendule est soumis à la force exercée par l'électroaimant, permettant le déplacement vertical du pendule. Des résultats expérimentaux montrent que cette force  $F_m$  est inversement proportionnelle au carré de la distance entre le pendule et l'électroaimant  $x$ .

$$F_m = k \frac{i^2}{x^2} \quad (6-42)$$

où  $i$  est le courant circulant dans l'électroaimant et  $k$  une constante.

Ce système est naturellement instable en boucle ouverte. Pour un courant d'excitation constant donné, il y a attraction en dessus d'une certaine distance minimale et le pendule vient au contact de la bobine

d'excitation, et inversement, au delà d'une certaine distance, la force de la pesanteur l'emporte et le pendule tombe.

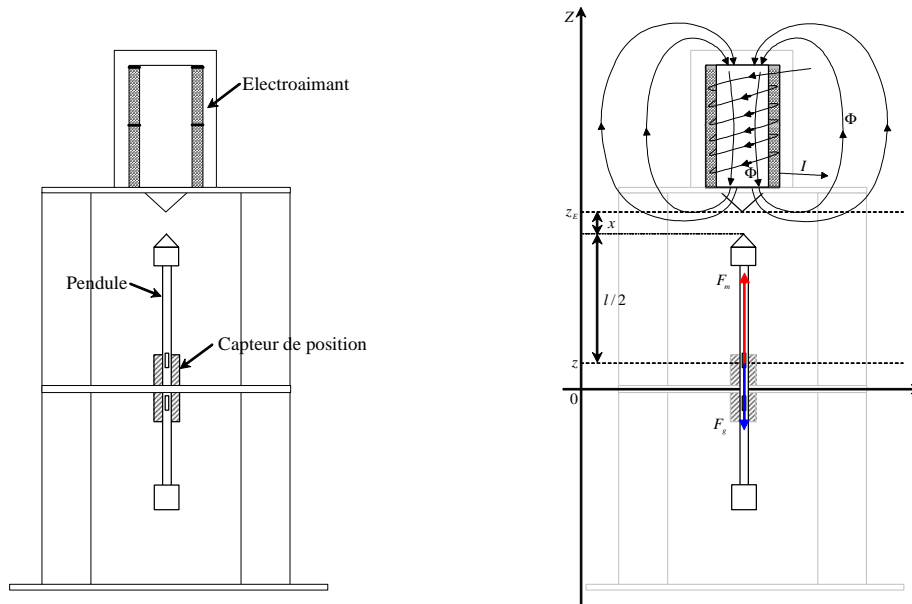


Fig. 6.26 – Système de suspension magnétique

### 6.5.1.2. Schéma fonctionnel de la boucle fermée

La commande d'un tel système exige la mesure de la position et de la vitesse de la partie mobile de la suspension et du courant qui traverse l'actionneur. Pour le dispositif de suspension magnétique considéré, disponible au sein de notre laboratoire, seuls le courant et la position du pendule (partie mobile) sont mesurables. La vitesse doit être estimée, si nécessaire, par construction d'un observateur.

Le schéma bloc du système en boucle fermée classique est représenté par la figure 6.27, où  $z$  est la position du pendule mesurée par le capteur,  $i$  est le courant en sortie de l'actionneur et  $u$  est la commande qui est la tension à l'entrée de l'actionneur.

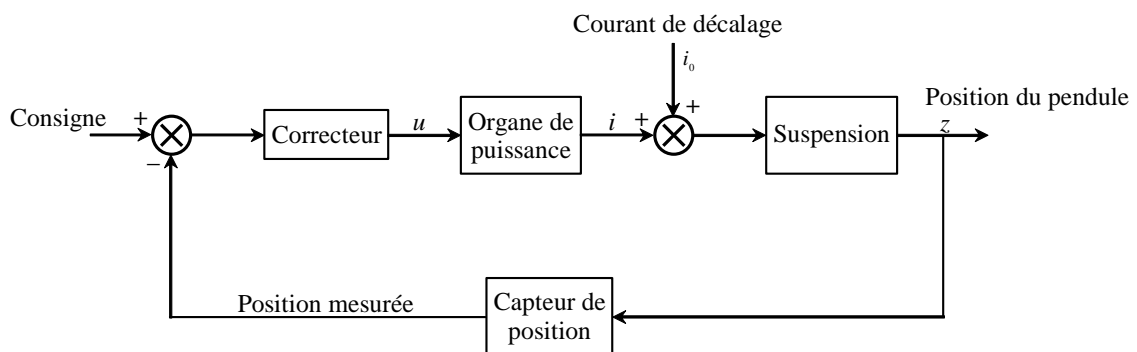


Fig. 6.27 – Boucle d'asservissement classique du système de suspension magnétique

L'électronique utilisée dans le capteur de position offre une très grande linéarité et une bonne précision, que l'on peut modéliser par un simple gain  $\beta$ .

L'organe de puissance permet, à partir d'une tension de commande  $u$  de fournir un courant variable  $i$  à la bobine de l'électroaimant. Ce courant  $i$  peut être mesuré via la mesure d'une tension aux bornes d'un pont de résistances (shunt).

### 6.5.1.3. Modèle dynamique

Dans un premier temps, le courant à la sortie de l'organe de puissance  $i$  et la tension de commande  $u$  sont considérées proportionnelles et leur relation entrée/sortie est donnée par :

$$i = k_v u \quad (6-43)$$

où  $k_v$  est le facteur d'amplification de l'actionneur.

Un courant de décalage  $i_0$  permet de créer un champ magnétique de référence. Ce courant constant  $i_0$  est réglé directement par un potentiomètre sur la maquette. Dans le modèle,  $i_0$  joue ainsi un rôle de perturbation constante à l'entrée du processus à commander. Bien réglé, il permet de « soulager » la partie dynamique de l'organe de puissance qui peut alors fonctionner autour de valeurs faibles et éviter des phénomènes de saturations.

Un bilan des forces appliquées au pendule donne :

$$m \ddot{z} = F_m - F_g = k \frac{(i + i_0)^2}{x^2} - mg = k \frac{(i + i_0)^2}{((z_E - l/2) - z)^2} - mg = k \frac{(i + i_0)^2}{(x_0 - z)^2} - mg \quad (6-44)$$

où

- $m$  et  $l$  sont la masse et la longueur du pendule respectivement,
- $z$  et  $z_E$  représente respectivement, la position du pendule et la position où l'électroaimant est actif, mesurées par rapport au centre du capteur, dans un repère absolu (cf. figure 6.26),
- $x_0$  représente l'entrefer.

En choisissant  $x = [z, \dot{z}]^T$  comme vecteur d'état, le système non linéaire s'écrit sous la forme d'état suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \frac{k (k_v u + i_0)^2}{m (x_0 - x_1)^2} \\ y = \beta x_1 \end{cases} \quad (6-45)$$

Pour toute position  $z = x_{1*} \leq x_0$ , l'équilibre du système est défini par le vecteur d'états  $x_* = [x_{1*}, 0]^T$  et la commande  $u_* = \sqrt{(mg/k)(x_0 - x_{1*})} / k_v$ .

### 6.5.2. Cahier des charges

Le système de suspension magnétique étant instable en boucle ouverte, Nous nous intéressons donc ici à stabiliser le pendule et à asservir sa position verticale.

Les spécifications temporelles considérées sont données par :

- Une entrée de référence  $r$  de type échelon d'une amplitude comprise entre  $+4.10^{-3}$  (m) et  $-4.10^{-3}$  (m) ,
- Un temps de réponse du système commandé inférieur à 0,1(s) ,
- Un dépassement maximal de 20% ,
- Une amplitude de la tension de commande limitée à 5 (V) ,
- Une erreur statique nulle,
- Un rejet de perturbation pour  $i_0 = cste$  .

Afin de garantir la stabilité de la boucle fermée une commande appropriée est nécessaire. Le système non linéaire étudié est sous forme triangulaire. C'est pourquoi la commande par Backstepping est choisie. Le principe et les techniques relatives à cette dernière ont été brièvement rappelés dans l'annexe 1.

### 6.5.3. Synthèse de la commande Backstepping en vue d'une validation expérimentale

Dans un premier temps, l'objectif de la synthèse par Backstepping est d'asservir la position  $z = x_1$  afin d'assurer le suivi à une entrée de référence  $r$  de type échelon en absence de la perturbation  $i_0$ . Afin qu'elle puisse être validée sur la maquette, la loi de commande synthétisée doit tout d'abord être adaptée pour respecter le cahier des charges imposé.

Pour ce faire, nous procédons en plusieurs étapes. Pour le modèle de suspension magnétique, nous développons différentes lois de commandes qui, au fur et à mesure du processus de synthèse, prennent en compte des résultats antérieurs.

Pour chacune de ces étapes, nous détaillons le calcul de la loi de commande, l'analyse de la boucle fermée et les résultats de simulation correspondants afin d'en déduire les futures améliorations à apporter pour que la loi de commande devienne de plus en plus réaliste et implémentable.

#### 6.5.3.1. Hypothèses

Avant que nous commençons la synthèse de la commande par Backstepping et afin de simplifier le calcul, nous faisons les hypothèses suivantes :

- Le signal de référence  $r$  vérifie :  $\dot{r} = 0$  (la dynamique des transitions rapides entre les échelons est négligée),
- Le bruit de mesure de la position du pendule suit une distribution normale  $N(0,10^{-9})$  .

#### 6.5.3.2. Calcul formel de la loi de commande Backstepping sous Mathematica

Le calcul des lois de commande par Backstepping a été effectué avec un programme que nous avons développé sous le logiciel de calcul formel Mathematica (conformément à la théorie rappelée en annexe 1). Ce dernier permet un interfaçage du calcul avec Matlab pour effectuer les simulations et l'implémentation en toute souplesse. La procédure de synthèse récursive du Backstepping est rendue complètement automatique. Le programme conçu ne nécessite que les équations du système et la définition de la paramétrisation des fonctions de Lyapunov choisie. En sortie du programme, il est possible d'obtenir la fonction de Lyapunov assignable globale et sa loi de commande stabilisante, le changement de variable plaçant le système dans les coordonnées d'erreur, l'estimateur du paramètre incertain et le code de Matlab pour la simulation.

Dans la littérature, un travail similaire a été développé en utilisant Maple et Matlab [Zin99]. Le programme résultant est similaire au notre, il offre le même type d'opérations avec une possibilité de couplage avec la synthèse par mode glissant. Toutefois, notre programme se distingue par le fait de permettre le choix et la paramétrisation des fonctions de Lyapunov ce qui représente, dans cette méthode, un degré de liberté fondamental pour concevoir des lois de commande plus adaptées aux différents cahiers des charges.

Dans la suite de cette section, les étapes de synthèse présentées sont les résultats directs du programme Mathematica développé.

### 6.5.4. Synthèse de la commande Backstepping

Dans un premier temps, la perturbation  $i_0$  est considérée nulle. Afin d'alléger le développement des équations, nous introduisons de nouvelles notations dans le modèle :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \theta \lambda(x_1) u^2 \\ y = \beta x_1 \end{cases} \quad (6-46)$$

avec  $\theta = k k_v^2 / m$  est un nouveau paramètre constant connu et  $\lambda(x_1) = (x_0 - x_1)^{-2}$ .

Le processus de calcul de la loi de commande se fait en deux étapes :

#### Étape 1 :

Considérons le sous-système

$$\dot{x}_1 = x_2 \quad (6-47)$$

Soient la première variable d'erreur  $e_1 = x_1 - r$  et la fonction de Lyapunov assignable :

$$V(e_1) = \frac{1}{2} e_1^2 \quad (6-48)$$

Sa dérivée temporelle est donnée par :

$$\dot{V} = e_1 \dot{e}_1 = e_1 x_2 \quad (6-49)$$

Si nous choisissons la première commande virtuelle stabilisante

$$x_2^{des} = -c_1 e_1 = \alpha(x_1) \quad \text{avec} \quad c_1 > 0 \quad (6-50)$$

nous obtenons :  $\dot{V} = -c_1 e_1^2 \leq 0$ .

Il est évident que ce correcteur stabilise la première équation qui devient  $\dot{e}_1 + c_1 e_1 = 0$  ou  $e_1(t) = e_1(0) e^{-c_1 t}$ . Le choix de  $c_1$  permet d'atteindre l'équilibre plus ou moins rapidement.

#### Étape 2 :

La deuxième variable d'erreur est alors :  $e_2 = x_2 - \alpha$ . Le système se réécrit alors sous la nouvelle forme :

$$\begin{cases} \dot{e}_1 = e_2 + \alpha \\ \dot{e}_2 = -g + \theta \lambda(e_1) u^2 + c_1 (e_2 + \alpha) \end{cases} \quad (6-51)$$

où  $\lambda(e_1) = (x_0 - r - e_1)^{-2}$ .

Nous augmentons la fonction la fonction de Lyapunov assignable de la première étape par un terme pénalisant la déviation de la variable d'état  $x_2$  de sa valeur désirée  $x_2^{des} = \alpha$  :

$$V_1(e_1, e_2) = \frac{1}{2}(e_1^2 + e_2^2) \quad (6-52)$$

Le choix de la commande  $u$  telle que :

$$u^2 = \frac{1}{\theta \lambda(e_1)} (-c_2 e_2 - e_1 + g - c_1 (e_2 + \alpha)) \quad \text{avec } c_2 > 0 \quad (6-53)$$

permet d'assurer la non positivité de la dérivée  $\dot{V}_1$  qui vérifiera :  $\dot{V}_1 = -c_1 e_1^2 - c_2 e_2^2 \leq 0$ . Ce qui implique la stabilité asymptotique globale du point d'équilibre.

#### 6.5.4.1. Analyse du système en boucle fermée

Le système en boucle fermée résultant de la commande par Backstepping est donné par :

$$\begin{cases} \dot{e}_1 = e_2 - c_1 e_1 \\ \dot{e}_2 = -c_2 e_2 - e_1 \end{cases} \quad (6-54)$$

Ce système linéaire équivalent est stable ; il présente deux valeurs propres à partie réelle négative,

$$\lambda_{1,2} = -\frac{1}{2}(c_1 + c_2 \pm \sqrt{(c_1 - c_2)^2 - 4}) \quad (6-55)$$

Une analyse des dynamiques de la boucle fermée montre que le choix des paramètres de commande  $c_1$  et  $c_2$  ne permet pas de couvrir la totalité des dynamiques d'un système de second ordre.

En effet, les expressions du facteur d'amortissement  $\xi$  et de la pulsation propre  $\omega_n$  correspondants au système bouclé (6-54) sont données par :

$$\begin{cases} \xi = \frac{(c_1 + c_2)}{2\sqrt{1 + c_1 c_2}} \\ \omega_n = \sqrt{1 + c_1 c_2} \end{cases} \quad (6-56)$$

Le tracé du domaine des dynamiques atteignables par la commande (6-53) dans le plan  $(\xi, \omega_n)$  est donné par la figure ci-dessous,

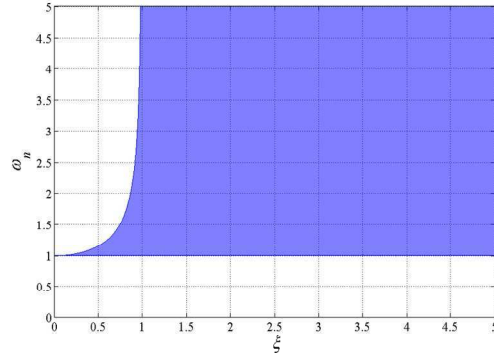


Fig. 6.28 – Domaine de variation de  $\xi$  et  $\omega_n$

Cette figure montre que la loi de commande synthétisée ne permet ni d’atteindre un régime oscillatoire rapide avec un coefficient d’amortissement  $\xi$  proche de 0,5, ni d’asservir le système à des pulsations propres  $\omega_n$  inférieures à 1 (rad/s).

Ces limites résultent d’un mauvais choix de la structure et/ou de la paramétrisation de la loi de commande synthétisée. Dans le cas d’une synthèse pas Backstepping, ce choix dépend originairement de la sélection des fonctions de Lyapunov à chaque étape de synthèse. C’est pour cette raison que nous avons développé un logiciel permettant d’avoir toute liberté dans le choix des fonctions de Lyapunov. Nous avons pu constater que contrairement aux démarches habituelles qui ne visent qu’à la stabilisation via un raisonnement algébrique, il faut introduire suffisamment de paramètres libres dans les fonctions de Lyapunov pour pouvoir également prendre en compte des demandes de performances et de robustesses.

Ainsi, nous proposons une nouvelle paramétrisation de la fonction de Lyapunov au niveau de la deuxième étape de synthèse. On introduit alors un nouveau paramètre  $c_3 > 0$  dans la définition de la fonction de Lyapunov augmentée. On remplace donc  $V_1$  dans la deuxième étape de synthèse par :

$$V_1(e_1, e_2) = \frac{1}{2}(e_1^2 + c_3 e_2^2) \tag{6-57}$$

Cette fonction est définie positive et sa dérivée temporelle est donnée par :

$$\begin{aligned} \dot{V}_1 &= e_1 \dot{e}_1 + c_3 e_2 \dot{e}_2 \\ &= e_1(e_2 + \alpha) + c_3 e_2(-g + \theta \lambda(e_1)u^2 + c_1(e_2 + \alpha)) \\ &= e_1 \alpha + c_3 e_2(e_1/c_3 - g + \theta \lambda(e_1)u^2 + c_1(e_2 + \alpha)) \end{aligned} \tag{6-58}$$

Si on choisit

$$u^2 = \frac{1}{\theta \lambda(e_1)}(-c_2/c_3 e_2 - e_1/c_3 + g - c_1(e_2 + \alpha)) \text{ avec } c_2 > 0 \tag{6-59}$$

on obtient  $\dot{V}_1 = -c_1 e_1^2 - c_2 e_2^2 \leq 0$ , d’où la stabilité asymptotique globale de l’équilibre  $x_* = [x_{1*}, 0]^T$ .

Le nouveau système équivalent en boucle fermée est donné par :

$$\begin{cases} \dot{e}_1 = e_2 - c_1 e_1 \\ \dot{e}_2 = -(c_2 e_2 + e_1) / c_3 \end{cases} \quad (6-60)$$

C'est un système linéaire stable avec une nouvelle paire de valeurs propres à partie réelle négative :

$$\lambda_{1,2} = -\frac{1}{2c_3} \left( c_1 c_3 + c_2 \pm \sqrt{(c_1 c_3 - c_2)^2 - 4c_3} \right) \quad (6-61)$$

Les nouvelles expressions du facteur d'amortissement et de la pulsation propre sont données par :

$$\begin{cases} \xi = \frac{c_1 c_3 + c_2}{2\sqrt{c_3(1 + c_1 c_2)}} \\ \omega_n = \sqrt{(1 + c_1 c_2) / c_3} \end{cases} \quad (6-62)$$

Comparant à la commande (6-53), le domaine des dynamiques atteignables par la nouvelle commande (6-59) est complet. En effet, plus le paramètre  $c_3$  est choisi grand, plus le système peut atteindre des dynamiques lentes ( $< 1$  rad/s), et réciproquement, plus  $c_3$  est petit, plus le système peut atteindre des régimes oscillatoires rapides (cf. figure 6.29).

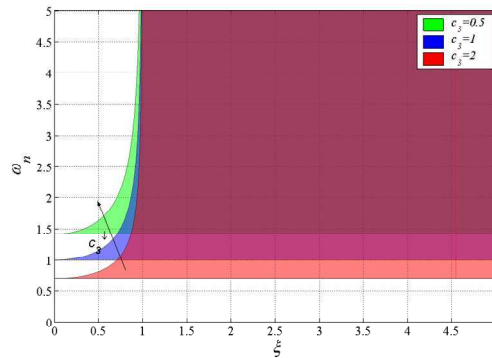


Fig. 6.29 – Domaine de variation de  $\xi$  et  $\omega_n$  en fonction de  $c_3$

Ainsi, le paramètre  $c_3$  de la loi de commande (6-59) joue le rôle d'un degré de liberté supplémentaire qui peut s'avérer très pratique lors d'une procédure d'optimisation pour la validation d'un gabarit temporel par exemple.

**Remarques :**

◆ Même si, théoriquement, la dynamique de la sortie du système en boucle fermée coïncide avec celle d'un système linéaire du second ordre, il est pratiquement impossible d'atteindre toutes ces dynamiques car la commande (6-59) présente une singularité. Ce type de problèmes est fréquemment posé par les méthodes de synthèse algébriques qui se basent sur une inversion du modèle (et dont le Backstepping fait partie).

Dans le cas de notre étude, la singularité peut être contrecarrée, par l'ajout de la contrainte supplémentaire sur la commande :

$$u^2 \geq 0 \Leftrightarrow (1/c_3 - c_1^2)e_1 + (c_2/c_3 + c_1)e_2 \leq g \quad (6-63)$$



Contrairement aux contraintes déjà imposées par la structure de la loi de commande ( $c_{i=1,2,3} > 0$ ), cette dernière dépend du signal de référence  $r$  et des états du système  $x = [z, \dot{z}]^T$  ce qui rend le choix des paramètres de commande plus laborieux. La plage de variation des paramètres de commande  $c_{i=1,2,3}$  devient plus restreinte et l'ensemble des dynamiques atteignables dans le plan  $(\xi, \omega_n)$  est limité ce qui risque d'atténuer considérablement la performance de la loi de commande (6-59).

◆ D'autre part, la loi de commande synthétisée dépend de l'état  $x_1 = z$  qui n'est disponible qu'à travers un capteur de position de gain  $\beta$ . En pratique (cf. figure 6.27), l'asservissement de la position du pendule se fait à travers la position mesurée  $y = \beta x_1$ . Par conséquent, deux solutions sont envisageables :

Une première solution consiste à respecter le schéma de la boucle et à asservir la position du pendule via la position mesurée. Dans ce cas, l'erreur  $e_1$  utilisée dans la première étape de synthèse est remplacée par  $e'_1 = y - y_r = \beta(x_1 - r) = \beta e_1$ . La suite consiste à choisir les mêmes fonctions de Lyapunov et de refaire toute la procédure de synthèse pour calculer de nouvelle loi de commande par Backstepping. On trouve :

$$u^2 = (-c'_2/c'_3 e'_2 - \beta/c'_3 e'_1 + g - c'_1(e'_2 + \alpha')) / (\theta \lambda(e'_1)) \quad (6-64)$$

où  $\alpha' = -c'_1 / \beta e'_1$ ,  $e'_2 = x_2 - \alpha'$ ,  $\lambda(e'_1) = (x_0 - (e'_1 + y_r) / \beta)^{-2}$  et  $c'_{i=1,2,3} > 0$ .

Cette relation qui définit la loi de commande  $u$  est équivalente à celle donnée par (6-59). En effet, il suffit de choisir le vecteur de paramètres de commande  $c' = [c'_1, c'_2, c'_3]^T$  tel que  $c'_1 = c_1$ ,  $c'_2 = \beta^2 c_2$  et  $c'_3 = \beta^2 c_3$  pour obtenir exactement la même boucle fermée.

Toutefois, comme il s'agit d'un capteur dont le transfert est un simple gain  $\beta$ , il existe une deuxième solution plus simple qui consiste à remplacer  $x_1$  dans l'expression de la commande  $u$  (cf. équation (6-59)) par le terme  $y / \beta$ . Cette solution est bien plus pratique car elle permet d'éviter le parcours des différentes étapes de calcul. Par commodité, c'est cette solution qui sera retenue dans la suite de cette étude.

### 6.5.4.2. Formulation de la faisabilité du cahier des charges en un problème d'optimisation

La stabilité du système en boucle fermée et l'erreur statique nulle étant garanties par construction de la loi de commande, il s'agit maintenant de trouver le jeu de paramètres de commande  $c_{i=1,2,3}$  pour lequel le reste des spécifications du cahier des charges seront vérifiées.

Selon le principe décrit dans le chapitre 3, le problème de faisabilité du cahier des charges se formule par le problème d'optimisation paramétrique sur les signaux de la boucle fermée  $y(t, c)$  et  $u(t, c)$  :

$$\begin{aligned} &\text{Existe-t-il } c = [c_1, c_2, c_3]^T \text{ tel que :} \\ &\begin{cases} T_r(y(t, c)) \leq 0,1 \\ D_{\%}(y(t, c)) \leq 0,2 \\ -5 \leq u(t, c) \leq 5 \end{cases} \end{aligned} \quad (6-65)$$

A ce problème s'ajoute des contraintes liées à la structure de la loi de commande synthétisée. Le problème global s'écrit :

Existe – t – il  $c = [c_1, c_2, c_3]^T$  tel que :

$$\begin{cases} T_r(y(t, c)) \leq 0,1 \\ D_{\%}(y(t, c)) \leq 0,2 \\ 0 \leq u(t, c) \leq 5 \\ c_i > 0 \quad \text{pour } i = 1, 2, 3 \end{cases} \quad (6-66)$$

En utilisant la formulation par gabarits et l'approche de pénalisation exacte, ce problème de faisabilité est transformé en un problème d'optimisation équivalent sous des contraintes de bornes seulement :

$$\min_{c_i > 0} \int_{t_i}^{t_f} [\eta_{11} |y(t, c) - \bar{y}(t)|_+ + \eta_{12} |y(t) - y(t, c)|_+ + \eta_{21} |u(t, c) - \bar{u}(t)|_+ + \eta_{22} |u(t) - u(t, c)|_+] dt \quad (6-67)$$

où,

- $\underline{y}$  et  $\bar{y}$  sont, respectivement, les gabarits minimal et maximal sur la sortie  $y$ ,
- $\underline{u}$  et  $\bar{u}$  sont, respectivement, les gabarits minimal et maximal sur la commande  $u$ ,
- $\eta_{11}$ ,  $\eta_{12}$ ,  $\eta_{21}$  et  $\eta_{22}$  sont des pondérations,
- $[t_i, t_f]$  représente l'intervalle de temps sur lequel le critère est calculé.

Le cahier des charges sera faisable si et seulement si le critère minimal obtenu est nul et les contraintes de bornes sont vérifiées.

L'évaluation du critère 6-67 (qui représente l'aire située en dehors des enveloppes temporelles) peut se faire à partir du calcul de la commande et de la sortie du système en boucle fermée. Ces grandeurs sont estimées numériquement par une intégration du système d'équations différentielles correspondant. A partir de cette intégration numérique, une deuxième intégration est effectuée pour évaluer la fonction critère à minimiser. En fait, ce calcul est généralement imprécis car il s'effectue à base d'une intégration sur des grandeurs estimées et donc il peut être fortement bruité. Dès lors, cette méthode rudimentaire ne convient pas à notre problématique d'optimisation où le critère doit être évalué précisément et rapidement.

Une solution plus efficace consiste à augmenter le système en boucle fermé par une équation différentielle décrivant la variation du critère. Dans notre cas, l'équation à ajouter s'écrit :

$$\dot{J}_c = \eta_{11} |y(t, c) - \bar{y}(t)|_+ + \eta_{12} |y(t) - y(t, c)|_+ + \eta_{21} |u(t, c) - \bar{u}(t)|_+ + \eta_{22} |u(t) - u(t, c)|_+ \quad (6-68)$$

Cette formulation permet de calculer simultanément les différents signaux de la boucle fermée, le critère d'optimisation  $J_c$  et même son gradient à travers la méthode des fonctions de sensibilité. Elle présente l'avantage d'allouer un pas d'intégration adaptatif à la dynamique du système augmenté complet ce qui permet d'obtenir des estimées plus précises. Le problème d'optimisation substitué est alors :

$$\min_{c_i > 0} (J_c(t_f) - J_c(t_0)) \quad (6-69)$$

Comme il s'agit d'inégalités strictes et afin d'utiliser la reparamétrisation quadratique (cf. équation (4-8)), ces contraintes de bornes sont légèrement durcies selon l'implication suivante :

$$(c_i \geq \kappa \text{ avec } \kappa \in \mathfrak{R}_+^* \text{ et } \kappa \text{ très petit}) \Rightarrow (c_i > 0) \quad (6-70)$$

Le changement de variables consiste alors à poser  $c_i = (c'_i)^2 + \kappa$  et de minimiser le nouveau critère  $J_c$  par rapport au vecteur de paramètres  $c' \in \mathfrak{R}^3$

$$\min_{c \in \mathfrak{R}^n} (J_c(t_f) - J_c(t_0)) \quad (6-71)$$

### 6.5.4.3. Résultats de simulation

Les paramètres du modèle de suspension sont identifiés en laboratoire. Les valeurs sont :  $m = 0.0844$  (kg),  $g = 9.81$  (ms<sup>-2</sup>),  $k_v = 0.1$  (AV<sup>-1</sup>),  $k = 0.005$  et  $x_0 = 0.011$  (m).

Le schéma bloc de la boucle fermée (modèle+contrôleur Backstepping) est donné par la figure 6.30. Ce dernier est augmenté par une structure d'optimisation des paramètres de commande  $c$ .

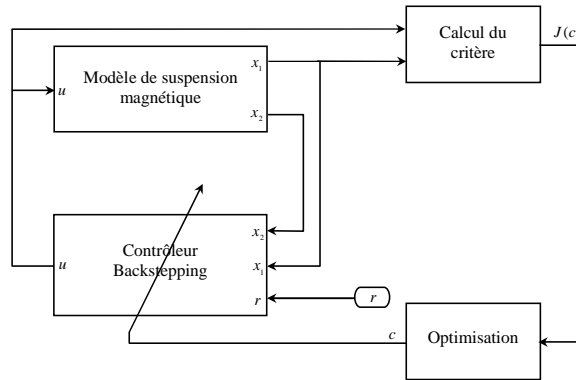


Fig. 6.30 – Structure d'optimisation de la commande Backstepping

Les algorithmes AGU et ASM avec reparamétrisation de la fonction coût sont utilisés pour résoudre le problème d'optimisation (6-71). Ces versions correspondent, respectivement, aux algorithmes 15 et 12 du chapitre 4.

Rappelons que, ces méthodes d'optimisation n'assurent qu'une convergence locale. C'est pourquoi le choix initial des paramètres de commande  $c_1$ ,  $c_2$  et  $c_3$  est décisif pour les amorcer convenablement.

En exploitant la structure linéaire du système en boucle fermée, on propose d'initialiser l'optimisation à partir de la première paramétrisation de la loi de commande par Backstepping (cf. équation (6-53)). On fixe alors le paramètre de commande  $c_3$  à 1 et on résout le problème algébrique (6-62) en fonction du couple  $(\xi, \omega_n)$  afin d'approximer au mieux la dynamique en boucle fermée prescrite par le cahier des charges. Cette dynamique de sortie peut bien correspondre à un système de second ordre (premier ordre double) avec  $\xi = 1$  et  $\omega_n = 48$  (rad/s), ou en d'autres termes à l'un des couples de paramètres suivants :  $(c_1, c_2) = (47, 49)$  ou  $(c_1, c_2) = (49, 47)$ . L'existence de plus d'une solution est une preuve de la non identifiabilité des deux structures (6-53) et (6-59), et par suite, de la non convexité stricte des problèmes d'optimisation associés.

Après avoir initialisé convenablement les paramètres de commande pour se rapprocher des spécifications de la sortie  $y(t)$ , nous testons la faisabilité des contraintes de commande en résolvant le problème d'optimisation sans contraintes (6-71).

Comme il n'est pas possible d'atteindre toutes les dynamiques d'un système de second ordre linéaire par la commande non linéaire (6-59), nous proposons, dans le cas où le cahier des charges est prouvé faisable, d'évaluer au mieux la performance de la loi de commande (6-59) en minimisant le temps de

réponse du système sous les mêmes contraintes du cahier des charges. Pour ce faire, le critère suivant est utilisé :

$$J_c = T_r(y(t, c)) + \int_{t_0}^{t_f} \underbrace{[\eta_{11} |y(t, c) - \bar{y}(t)|_+ + \eta_{12} |y(t) - y(t, c)|_+ + \eta_{21} |u(t, c) - \bar{u}(t)|_+ + \eta_{22} |u(t) - u(t, c)|_+]}_{\text{pénalisation exacte}} dt \quad (6-72)$$

Structurellement, ce critère est mixte. Outre, la spécification directe du temps de réponse, il comporte un terme intégral de pénalisation exacte qui traduit, sous forme de gabarits, le reste des spécifications du cahier des charges. L'évaluation de ce critère peut se faire en le décomposant en deux termes. Le terme de pénalisation peut être calculé, comme auparavant, en introduisant sa variation avec le système d'état de la boucle fermée. Quant au temps de réponses, il ne peut qu'être estimé à base de la solution approchée de la sortie  $y(t)$ . Il est donc très important de bien choisir le pas d'intégration pour garantir un calcul du critère et de son gradient correct. La solution retenue, dans ce cas, consiste à choisir un faible pas d'intégration par rapport à la dynamique du système,  $10^{-4}$  (s) par exemple, et d'utiliser le calcul du gradient par les fonctions de sensibilité pour la partie intégrale du critère et par la méthode de dérivation complexe pour le terme temps de réponse.

Le système étant non linéaire, nous avons effectué une campagne de tests pour différentes entrées de référence  $r$ . Le tableau (6.23) synthétise les résultats obtenus par les deux algorithmes d'optimisation. L'étude comparative des résultats est conforme aux analyses avancées pour les problèmes numériques traités au chapitre 4. Elle affirme que les deux algorithmes présentent un même degré de performances pour ce problème. Nous nous contentons ici de présenter seulement les solutions optimales<sup>1</sup> trouvées : le meilleur temps de réponse  $T_r$ , la plage de variation de la commande  $u$  et les paramètres de commande associés.

Afin de mesurer le rôle du paramètre supplémentaire  $c_3$ , les deux configurations de la loi de commande  $c_3 = 1$  et  $c_3 > 0$  sont considérées. Elles correspondent respectivement aux lois de commande (6-53) et (6-59).

$r$ (mm)	$c_3 = 1$			$c_3 > 0$		
	$T_r$ (s)	$u$ (V)	$[c_1, c_2]_{opt}$	$T_r$ (s)	$u$ (V)	$[c_1, c_2, c_3]_{opt}$
-4	0.0958	[2.1e-3, 1.9354]	[49.5092, 49.5160]	0.0958	[2.1e-3, 1.9354]	[49.5092, 49.5160, 1.0001]
-3	0.0829	[3.1e-4, 1.8144]	[57.1750, 57.1753]	0.0829	[3.1e-4, 1.8144]	[57.1750, 57.1753, 1.0000]
-2	0.0677	[2.1e-4, 1.7005]	[70.0332, 70.0240]	0.0677	[2.1e-4, 1.7005]	[70.0331, 70.0239, 1.0000]
-1	0.0479	[3.5e-4, 1.5973]	[99.0388, 99.0419]	0.0479	[3.5e-4, 1.5973]	[99.0389, 99.0420, 0.9998]
-0.1	0.0151	[5.1e-4, 1.5165]	[313.2216, 313.1936]	0.0151	[5.1e-4, 1.5165]	[313.221, 313.193, 1.0001]
0.1	0.0056	[3.9e-3, 4.1001]	[851.2680, 851.6287]	0.0042	[1.2e-3, 3.3704]	[0.9133, 2.04e-3, 2.18e-6]
1	0.0176	[2.3e-3, 4.0999]	[269.0268, 269.4341]	0.0135	[7.7e-3, 3.4067]	[0.2703, 6.51e-3, 2.13e-5]
2	0.0249	[2.3e-3, 4.0998]	[190.1163, 190.6286]	0.0196	[3.3e-4, 3.5045]	[0.1848, 9.56e-3, 3.83e-5]
3	0.0305	[4.7e-3, 4.0998]	[155.6219, 155.2520]	0.0246	[4.0e-4, 3.5750]	[0.1588, 0.0117, 5.69e-5]
4	0.0352	[3.9e-3, 4.0998]	[134.4515, 134.7703]	0.0285	[3.4e-3, 3.5831]	[0.1342, 0.0136, 7.55e-5]

Tab. 6.23: Performances optimales de la boucle fermée en fonction de l'amplitude de la référence

<sup>1</sup> Ici, l'optimalité est à prendre toujours au sens local car nous n'avons aucune garantie de l'optimalité globale des solutions. L'optimalité locale est vérifiée via une série de tests aléatoires autour de la solution obtenue.

De ce tableau, nous observons que le cahier des charges est dûment respecté pour les deux configurations de la loi de commande et pour toutes la plage de signaux de référence : le régime stationnaire est établi en moins de 0,1 (s) avec un dépassement maximal de l'ordre de 5% et une commande positive inférieure 5 volts.

Dans le cas de la première paramétrisation de la loi de commande ( $c_3 = 1$ ), les dynamiques optimales de la sortie  $y(t)$  sont obtenues avec des paramètres de commande  $[c_1, c_2]_{opt}$  qui correspondent à une pulsation propres  $\omega_{n,opt}$  et à un coefficient d'amortissement  $\xi_{opt}$  sur la frontière de l'espace des dynamiques atteignables définie par  $\omega_{n,opt} = 1/\sqrt{1-\xi_{opt}^2}$  (cf. figure 6.28). Par conséquent, plus le système est rapide, plus le coefficient d'amortissement optimal  $\xi_{opt}$  est proche de 1. Ce qui implique, temporellement, un dépassement extrêmement faible, et fréquemment, une très faible partie imaginaire des deux pôles complexes conjugués tendant à se confondre en un seul pôle réel double.

La figure 6.31 décrit l'évolution du temps de réponse minimal en fonction de la taille des l'échelons d'entrée pour les deux paramétrisations de la loi de commande.

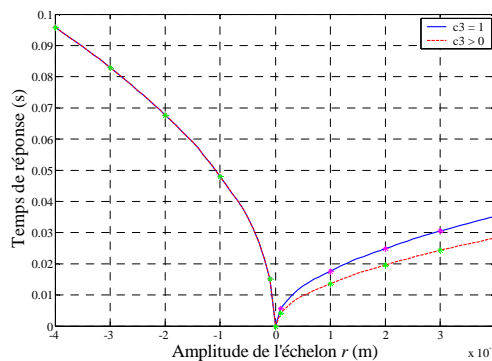


Fig. 6.31 – Variation du temps de réponse minimal en fonction de l'amplitude de l'échelon

L'allure des courbes vient étayer l'avantage de la paramétrisation de la deuxième loi de commande qui permet, comme nous le remarquons sur la figure 6.31, d'améliorer le temps de réponse du système bouclé d'environ 20% pour les entrées de référence positives. Pour les entrées négatives, le temps de réponses reste pratiquement le même pour les deux configurations de commande.

En effet, comme nous l'avons déjà constaté, la variation du paramètre  $c_3$  permet d'élargir le domaine des dynamiques et d'atteindre, comme dans le cas des entrées positives, des régimes rapides avec des efforts moins importants (cf. figures 6.32) : les dépassements de la réponse  $y(t)$  réduisent l'entrefer entre le pendule et l'électroaimant, augmentant ainsi la rapidité du système toute en réduisant la force électromagnétique exercée sur le pendule et donc la commande  $u$ .

Néanmoins, le comportement du système de suspension magnétique vis-à-vis des entrées de référence est loin d'être symétrique. Dans le cas de références négatives (des déplacements vers le bas (cf. figure 6.32)), le régime optimal du système, au sens du critère (6-72), reste apériodique (cf. figure 6.32) car les oscillations vont à l'encontre du critère à minimiser : les dépassements dus à ces derniers ne font qu'augmenter l'entrefer impliquant ainsi une augmentation de la force électromagnétique et donc une commande plus importante pour conserver la stabilité du pendule. Comme cette commande est limitée par la contrainte  $u(t) \geq 0$ , il n'est pas possible d'atteindre des régimes plus rapides. Dans ce cas, la

sensibilité de la fonction critère par rapport au paramètre  $c_3$  n'est pas significative et elle n'apporte aucune amélioration notable vis-à-vis à la minimisation du temps de réponse.

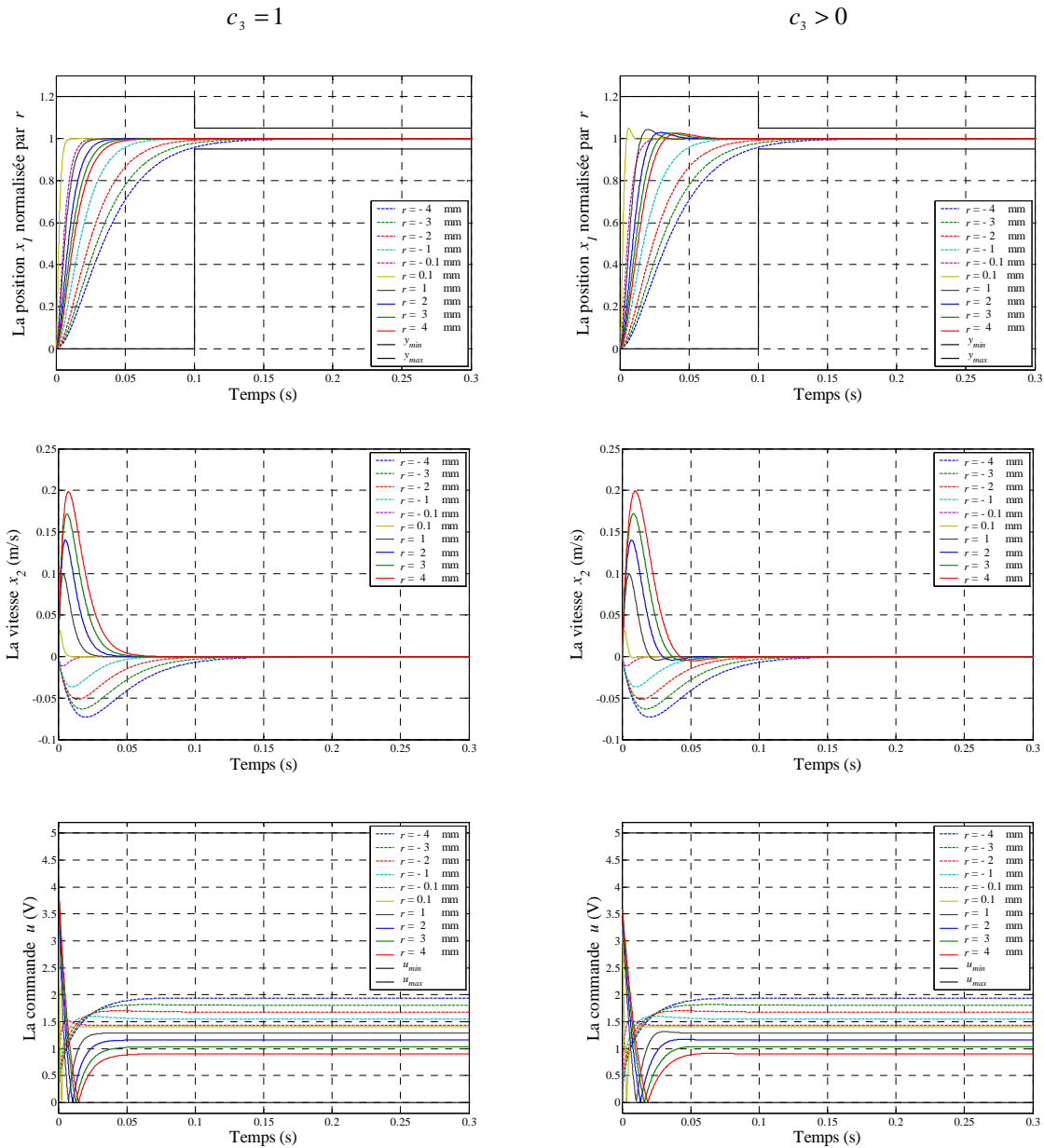


Fig. 6.32– Réponses temporelles optimales du système en boucle fermée

Par la suite, nous proposons de choisir l'entrée de consigne qui correspond aux pires performances comme étalon afin de l'adapter lors les synthèses de lois de commande à venir. Dans notre étude, cette entrée correspond à  $r = -4$  (mm) où le régime associé optimal, au sens du critère (6-72), est complètement apériodique :  $(\xi, \omega_n) = (49.5227, 0.9998)$ .

#### 6.5.4.4. Remarques concernant la résolution numérique des problèmes d'optimisation

◆ Une deuxième alternative à la formulation par gabarits serait de choisir, pour la sortie  $y(t)$ , un modèle de référence compatible avec les exigences du cahier des charges. Dans ce cas, nous pouvons exploiter la structure linéaire du système bouclé de nouveau pour formuler un nouveau problème d'optimisation où le but est de minimiser un critère sur l'erreur entre la sortie du système en boucle

fermée et la sortie d'un modèle de référence de second ordre. Ce critère peut prendre plusieurs formes telles ISE, IAE, ITSE ou ITAE. Quelque soit le choix de la forme de la fonction coût, cette dernière est nécessairement augmentée par un terme de pénalisation exacte traduisant les contraintes sur la commande non linéaire.

◆ Même si les méthodes d'optimisation utilisées présentent une robustesse vis-à-vis aux problèmes mal conditionnés, le choix des pondérations reste important car il peut influencer la vitesse de convergence des algorithmes. Il nous appartient donc de les ajuster afin d'atteindre les meilleurs résultats en un minimum d'itérations. Un choix judicieux consiste à normaliser chaque terme. Pour le critère (6-72), par exemple, on pose :  $\eta_{11} = 1/|\bar{y}(t)|$ ,  $\eta_{12} = 1/|\underline{y}(t)|$ ,  $\eta_{21} = 1/|\bar{u}(t)|$  et  $\eta_{22} = 1/|\underline{u}(t)|$ .

◆ Outre les pondérations, le choix des bornes d'intégration est primordial afin de traduire correctement les objectifs de commande. Par exemple, en plaçant  $t_0$  proche du temps du premier maximum de la réponse indicielle du système en boucle fermée, la fonction coût place une pondération fictive nulle sur la partie transitoire initiale de la réponse. Par conséquent, le contrôleur est ajusté pour minimiser l'erreur au delà du temps du premier maximum sans aucunes contraintes sur la phase transitoire initiale du signal. De plus, les résultats de simulation montrent que le choix de l'instant  $t_f$  est numériquement critique, en particulier, lorsqu'il s'agit de minimiser un temps de réponse d'un système non linéaire.

◆ L'évaluation de la fonction critère se fait numériquement en se basant sur le vecteur temps de la résolution numérique des équations différentielles du système en boucle fermée. La précision de ce calcul dépend de la densité de l'échantillonnage et de la méthode d'intégration choisie. Dans notre cas, le calcul a été fait en utilisant les méthodes de résolution à pas adaptatifs implémentés sous Matlab. Dans ces fonctions, même si l'intégration se fait à base d'un pas variable, les résultats peuvent être recouverts, à la demande de l'utilisateur, pour un pas fixé ce qui procure à l'utilisateur une plus grande facilité d'utilisation pour des vérifications ultérieures.

◆ Dans un cas général, la fonction "ode45" basée sur les formules explicites de Runge-Kutta d'ordre 4 et les coefficients de Dormand-Prince est utilisée. Néanmoins, cette fonction présente parfois des limites quand il s'agit d'un système d'équations différentielles à plusieurs échelles de temps (phénomènes rapide et lent). Ces dynamiques singulières peuvent bien surgir pendant un processus d'optimisation, même si cela n'est pas présent dans le résultat final. Dans ce cas, les méthodes dites rigides "stiff" ne sont plus efficaces et nous avons recours à des solveurs plus adaptés. Sur le logiciel Matlab, on peut citer dans cette catégorie les fonctions "ode15s", "ode23s", "ode23t" et "ode23tb".

◆ Il arrive parfois (rarement) que toutes les méthodes disponibles sur Matlab deviennent infructueuses ce qui rend le processus d'optimisation caduc ou le rend extrêmement lent. Une alternative à ces solveurs de Matlab est d'utiliser de nouvelles versions de solveurs implémentés sous le logiciel SUNDIALS (SUite of Nonlinear and DIFFerential/ALgebraic equation Solvers) [Sun07, Hin05, Ser05]. Ce dernier offre une grande variété de solveurs algébro-différentiels. Les plus appropriées à notre étude sont les fonctions "cvoid" et "cvoids". Les algorithmes employés dans ces fonctions sont à plusieurs étapes "multi stage" et à ordre et à pas variables. Pour des problèmes non rigides "nonstiff", elles utilisent le schéma prédicteur/correcteur d'Adams-Bashforth-Moulton, avec un ordre variant entre 1 et 12, tandis que pour des problèmes rigides (stiff), elles utilisent les formules BDFs (Backward Differentiation Formulas), avec un ordre variant entre 1 et 5.

A la base, l'implémentation des fonctions SUNDIALS est faite en langages C et Fortran mais la nouvelle boîte à outils SundialsTB (SUNDIALS ToolBox) [Ser06] permet d'interfacer la plupart de ses fonctions avec Matlab. Les tests effectués sur plusieurs problèmes montrent que la fonction "cvoid" est plus robuste numériquement et elle permet de fournir des résultats de très bonne qualité et plus rapidement en exploitant les techniques de programmation parallèle.

◆ En plus de la robustesse numérique, la fonction "cvodes" permet d'évaluer, parallèlement à l'intégration numérique des équations du système en boucle fermée, la sensibilité paramétrique de chaque variable d'état par rapport à tout les paramètres du modèle ce qui permet de calculer les différents gradients et d'estimer, par la suite, les  $\varepsilon$ -sous-différentiels. Cette option proposée à l'utilisateur est implémentée, en amont, dans le noyau de la fonction "cvodes". Elle permet de réduire la complexité de la tâche de calcul des sensibilités paramétriques ce qui représente un gain de temps considérable pour un processus d'optimisation.

### 6.5.5. Synthèse de la commande Backstepping par retour de sortie

La loi de commande développée jusqu'à ici dépend de la vitesse du pendule  $x_2 = \dot{z}$  qui n'est pas mesurable sur la maquette que nous disposons. Cette dernière doit alors être reconstruite à partir des signaux entrée/sortie du système (l'entrée  $u$  et de la seule mesure  $z$ ).

Une première solution rudimentaire consiste à estimer la vitesse  $\dot{z}$  à partir d'une différentiation numérique de la position  $z$ . On écrit :

$$\hat{\dot{z}}_i = \hat{x}_{2,i} \approx \frac{\Delta x_{1,i}}{\Delta t} = \frac{z_{i+1} - z_i}{t_{i+1} - t_i} \quad \text{pour } i = 1, \dots, n_f - 1 \quad (6-73)$$

où  $t_i$  et  $t_{i+1}$  représentent deux instants de mesure consécutifs et  $z_i$  et  $z_{i+1}$  leurs positions correspondantes.

Il est clair que la qualité de cette estimation dépend fortement de la résolution de la mesure de position. Idéalement, plus le pas d'échantillonnage est petit plus l'estimation est meilleure. En revanche, en présence d'un bruit de mesure (dû au capteur entre autre), la différence  $z_{i+1} - z_i$  devient rapidement petite et se noie dans le bruit de mesure ce qui dégrade considérablement l'estimée  $\hat{x}_2$ . Même si le capteur de position utilisé, sur notre maquette, délivre une mesure de bonne qualité, sa sortie est entachée d'un bruit de mesure qui peut fausser complètement l'estimation  $\hat{x}_2$  et réduire les performances de la commande voir même déstabiliser le système bouclé. Ce bruit est estimé statistiquement sur notre banc d'essai par une loi normale de moyenne nulle et de variance  $\sigma^2$  proche de  $10^{-9}$ .

En utilisant la dérivation numérique et le modèle de bruit décrit ci-dessus, les résultats de simulation correspondants aux précédents paramètres de contrôle optimaux sont donnés par la figure ci-dessous :



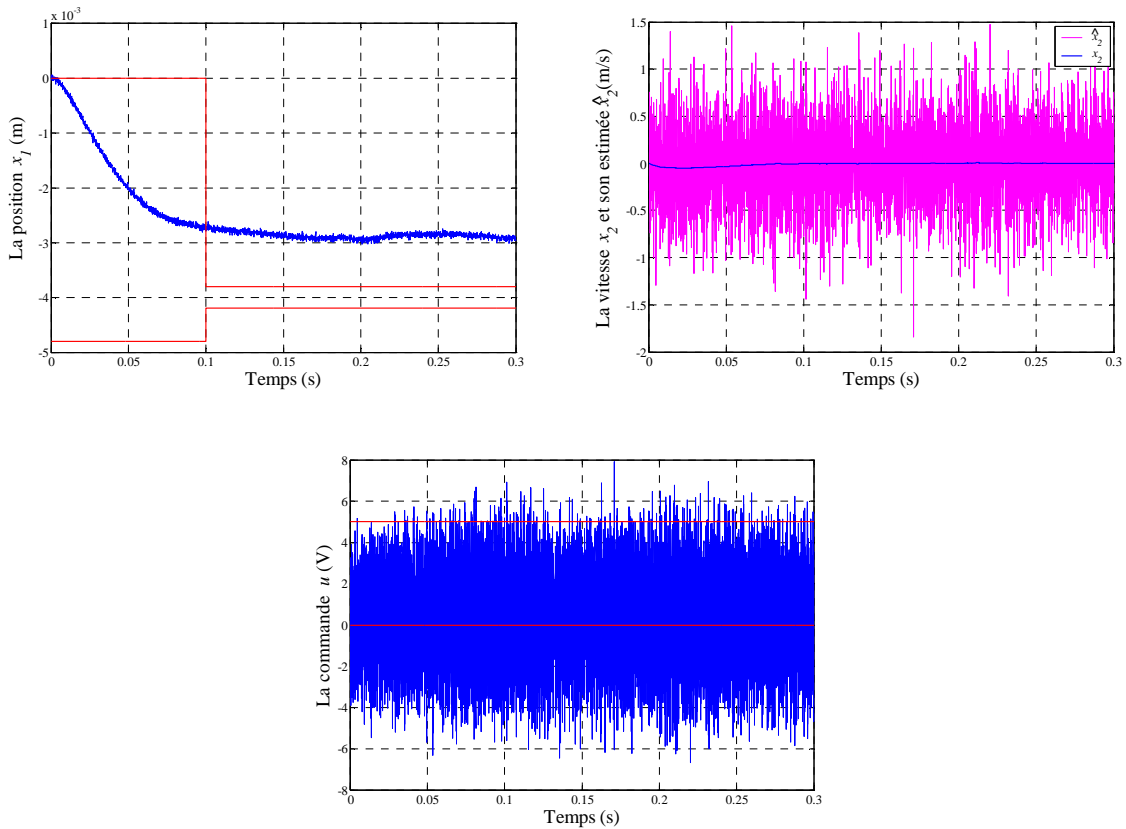


Fig. 6.33 – Réponses temporelles du système en boucle fermée (modèle+Backstepping+dérivation numérique)

Ces courbes de simulation témoignent de l’inadéquation de cette approche pour l’estimation d’un état d’un système dynamique soumis à des bruits de mesure. Nous observons que l’estimée de la vitesse est trop grossière et ne reflète pas du tout son évolution (courbe bleue sur la figure au milieu). La commande est inévitablement affectée et ne respecte plus ses contraintes de gabarit. En revanche, les simulations montrent que le système reste stable (a priori, il n’existe aucune garantie de stabilité) et conserve sa dynamique rapide au dépend d’une importante erreur statique.

Ces résultats pouvaient bien être pires car la commande par Backstepping est essentiellement une technique modale totalement basée sur la parfaite connaissance du système. La non robustesse de cette méthode vis-à-vis aux erreurs de modélisation paramétrique et structurelle est son principal défaut. Afin de pallier à ce problème, nous proposons dans un premier temps de synthétiser un observateur non linéaire de la vitesse qui assure la stabilité asymptotique globale du système.

### 6.5.5.1. Synthèse d’un observateur d’ordre réduit de la vitesse

En s’inspirant du principe de construction des observateurs linéaires [Rot03], nous proposons de synthétiser un observateur d’ordre réduit non linéaire afin d’estimer la vitesse  $x_2$ .

En effet, la première équation du modèle (6-46) peut être considérée comme une mesure  $\xi$  dépendant, ici, uniquement de la variable d’état à reconstruire  $x_2$ .

$$\xi = \dot{x}_1 = x_2 \tag{6-74}$$

Suivant le principe de construction des observateurs, on peut proposer comme reconstruteur de  $x_2$ , le vecteur  $\hat{x}_2$  défini par :

$$\begin{cases} \dot{\hat{x}}_2 = -g + \theta \lambda(x_1)u^2 + c_4(\xi - \hat{\xi}) \\ y = \beta x_1 \quad \text{et} \quad \hat{\xi} = \hat{x}_2 \end{cases} \quad (6-75)$$

ou alors :

$$\begin{cases} \dot{\hat{x}}_2 = -g + \theta \lambda(x_1)u^2 + c_4(\dot{y}/\beta - \hat{\xi}) \\ \hat{\xi} = \hat{x}_2 \end{cases} \quad (6-76)$$

L'inconvénient de cette structure d'observateur est de nécessiter, pour élaborer la mesure  $\xi$ , la dérivation de la sortie réelle  $y$  et donc revenir au même problème qu'auparavant.

Une façon de contourner cette difficulté consiste à définir la variable

$$\hat{v} = \hat{x}_2 - c_4 x_1 \quad (6-77)$$

ce qui permet d'obtenir :

$$\begin{aligned} \dot{\hat{v}} &= \dot{\hat{x}}_2 - c_4 \dot{x}_1 \\ &= -g + \theta \lambda(x_1)u^2 + c_4(\dot{y}/\beta - \hat{\xi}) - c_4 \dot{x}_1 \\ &= -g + \theta \lambda(x_1)u^2 - c_4 \hat{\xi} \end{aligned} \quad (6-78)$$

où n'apparaît aucune dérivation de la sortie.

En tenant compte du fait que  $\hat{\xi} = \hat{x}_2 = \hat{v} + c_4 x_1$ , (6-78) se réécrit sous la forme :

$$\dot{\hat{v}} = -g + \theta \lambda(x_1)u^2 - c_4(\hat{v} + c_4 x_1) \quad (6-79)$$

Ainsi, l'erreur d'observation  $\varepsilon = x_2 - \hat{x}_2$  suit une dynamique définie par :

$$\begin{aligned} \dot{\varepsilon} &= \dot{x}_2 - \dot{\hat{x}}_2 \\ &= -g + \theta \lambda(x_1)u^2 - \dot{\hat{v}} - c_4 \dot{x}_1 \\ &= -g + \theta \lambda(x_1)u^2 - (-g + \theta \lambda(x_1)u^2) + c_4(\hat{v} + c_4 x_1) - c_4 x_2 \\ &= c_4(\hat{x}_2 - x_2) = -c_4 \varepsilon \end{aligned} \quad (6-80)$$

Par conséquent, il suffit d'imposer  $c_4 > 0$  pour que l'erreur soit exponentiellement décroissante selon l'équation  $\varepsilon(t) = \varepsilon(0)e^{-c_4 t}$ , et le système se transforme en :

$$\begin{cases} \dot{x}_1 = c_4 x_1 + \hat{v} + \varepsilon \\ \dot{\hat{v}} = -g + \theta \lambda(x_1)u^2 - c_4(\hat{v} + c_4 x_1) \\ \dot{\varepsilon} = -c_4 \varepsilon \\ \hat{x}_2 = \hat{v} + c_4 x_1 \\ y = \beta x_1 \end{cases} \quad (6-81)$$

### 6.5.5.2. Analyse et simulation du système en boucle fermée

L'analyse de la stabilité du système bouclé par la structure Backstepping-observateur est faite dans l'annexe 2 (cf. section 8.2.1). Elle montre que la dynamique de l'observateur synthétisé n'influence pas la stabilité du système bouclé initial (système+Backstepping) qui reste conservée.

Le système en boucle fermée résultant de la commande (6-59) et de l'observateur (6-79) est linéaire et s'écrit :

$$\begin{cases} \dot{e}_1 = -c_1 e_1 + e_2 + \mathcal{E} \\ \dot{e}_2 = -e_1 / c_3 - c_2 / c_3 e_2 + (c_4 + c_1) \mathcal{E} \\ \dot{\mathcal{E}} = -c_4 \mathcal{E} \end{cases} \quad (6-82)$$

Sous les conditions  $c_{i=1,2,3,4} > 0$ , les valeurs propres de ce système sont à partie réelle négative et sont données par :

$$\begin{cases} \lambda_{1,2} = -\frac{1}{2c_3} (c_1 c_3 + c_2 \pm \sqrt{(c_1 c_3 - c_2)^2 - 4c_3}) \\ \lambda_3 = -c_4 \end{cases} \quad (6-83)$$

Nous remarquons que les modes du retour d'état et de l'observateur d'ordre réduit sont complètement découplés et par conséquent, le principe de séparation est singulièrement vérifié pour cette commande non linéaire. Cette propriété facilite le choix et l'interprétation de la dynamique de sortie de la boucle fermée.

La structure d'optimisation de la boucle fermée (système+Backstepping+observateur) est donnée par la figure ci-dessous :

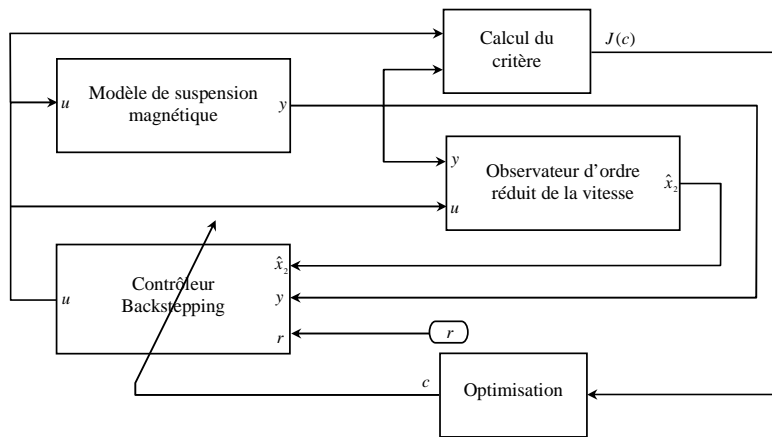


Fig. 6.34 – Structure d'optimisation de la commande Backstepping par retour de sortie

La question pratique du choix des paramètres initiaux fait appelle aux mêmes remarques que pour le choix des valeurs propres pour le calcul d'un retour d'état et d'un observateur. En général, on choisit pour l'observateur une dynamique légèrement plus rapide (typiquement deux fois) que la dynamique choisie lors du calcul du retour d'état, afin que cette dernière reste la dynamique dominante (bien que certains auteurs remettent en question cette règle [Lar93]). Contrairement au cas linéaire où les choix

effectués sont validés par une analyse fréquentielle, ils ne seront approuvés, dans le cas non linéaire, que par une limitation des signaux de commande.

Nous conservons ainsi les paramètres  $c_1$ ,  $c_2$  et  $c_3$  optimaux et nous initialisons  $c_4$  tel que  $\text{Re}(\lambda_1) = \text{Re}(\lambda_2) = \lambda_3 / 2$ . Ceci revient à poser  $c_4 \approx 100$ .

Pour montrer la convergence de l’observateur de vitesse, l’état  $\hat{v}$  est initialisé à -0.2 (m/s). Les résultats de l’optimisation sont donnés par

$T_r$ (s)	$u$ (V)	$[c_1, c_2, c_3, c_4]_{opt}$
0.0957	[0.1609, 1.9355]	[49.512, 50.021, 1.011, 102.838]

Les résultats de simulations sont satisfaisants. L’erreur d’observation due à la condition initiale est très vite annulée et les simulations sont très proches de ceux du correcteur Backstepping seul. Ces résultats étaient prévisibles vu la séparation des dynamiques entre l’observateur et le correcteur.

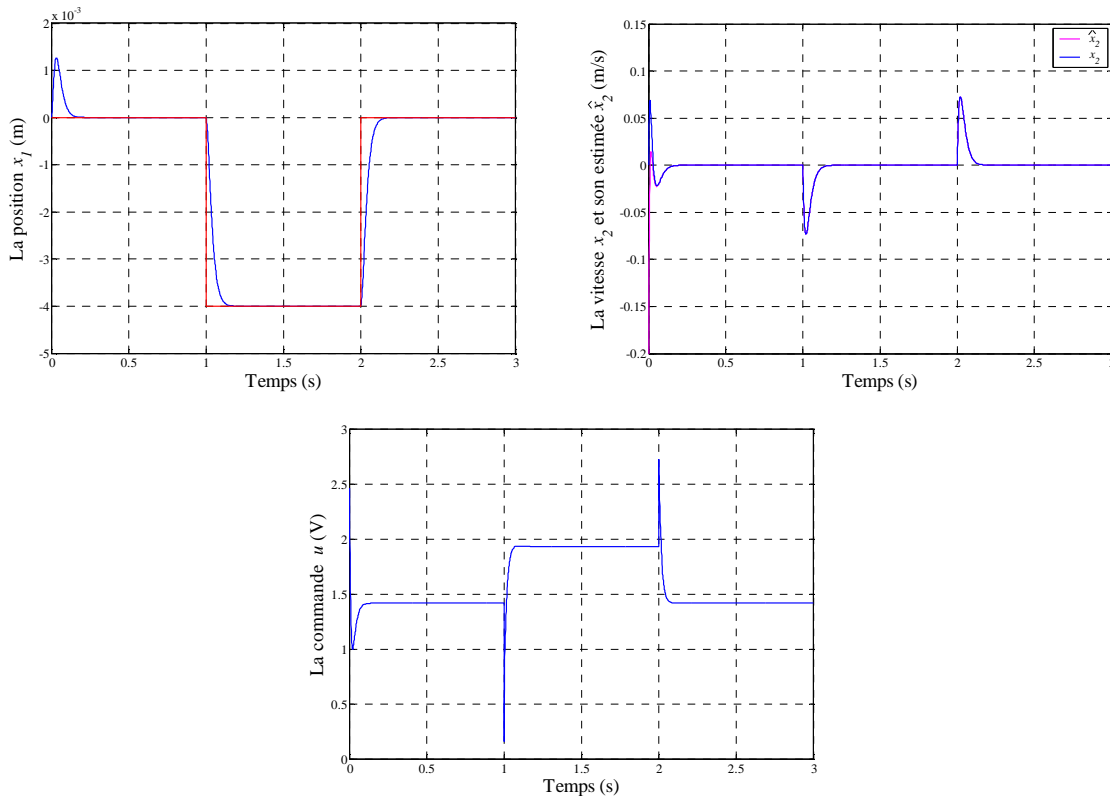


Fig. 6.35 – Résultats de simulation de la commande Backstepping par retour de sortie

Notons toutefois que cette bonne performance du correcteur dépend de la parfaite connaissance des paramètres du modèle. Ce point sera plus détaillé dans la phase de validation expérimentale.

### 6.5.5.3. Implémentation et validation expérimentale

Afin de valider ces résultats de simulation, la commande Backstepping avec retour de sortie est implémenté sur le système réel. Le banc expérimental utilisé est celui de la figure 6.36. Il se compose du dispositif de suspension magnétique et d’un ordinateur muni d’une carte entrée-sortie PCI-6024E.

interfacée par la boîte à outils Real “Time Windows Target” de Matlab [Tar99]. La période d’échantillonnage est égale à  $3.10^{-3}$  (s), et le déplacement peut être effectué entre -4 mm et 4 mm.



Fig. 6.36 – Banc d’essai de la suspension magnétique

La figure 6.37 présente les résultats expérimentaux obtenus le banc d’essai. Ces derniers ne sont pas complètement satisfaisants. En effet, même si l’allure et l’ordre des grandeurs sont respectés, la poursuite du signal référence n’est pas assurée et une importante erreur statique est présente. Le temps de réponse prélevé sur le système est supérieur à celui trouvé par la méthode d’optimisation. Il est vaut 0.12 (s).

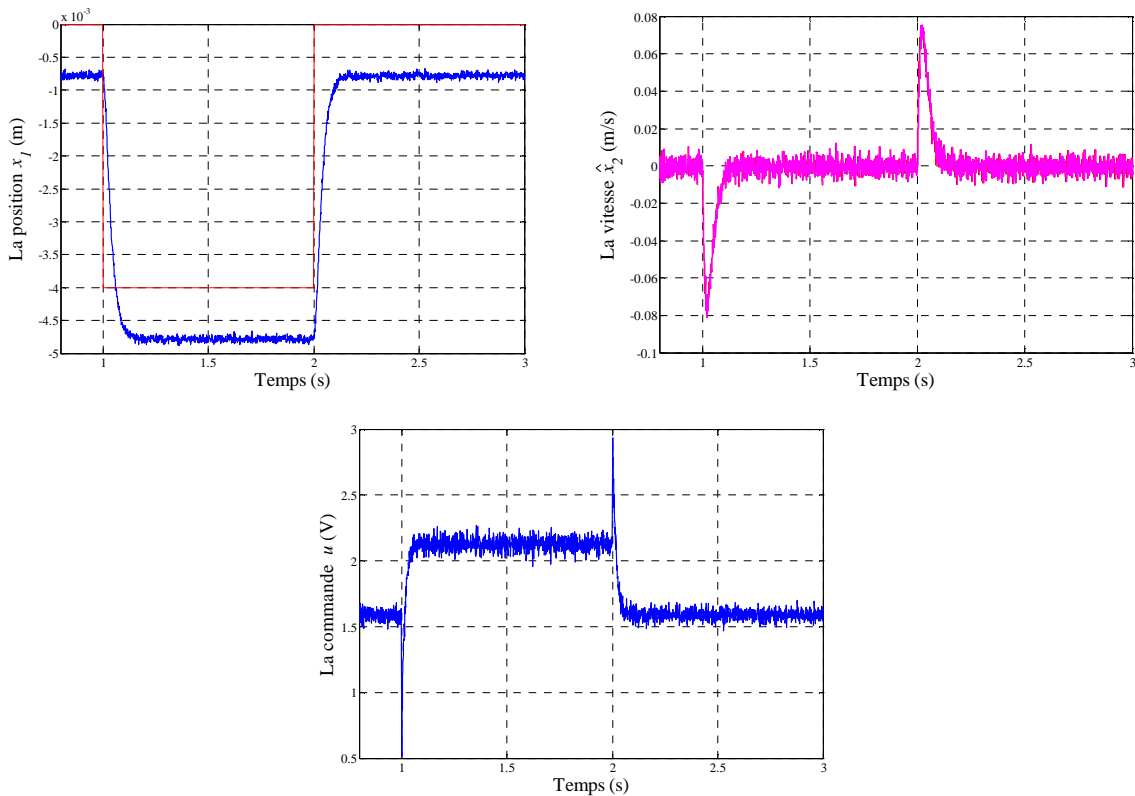


Fig. 6.37 – Réponses temporelles expérimentales de la commande Backstepping par retour de sortie

Très souvent la commande par Backstepping n’est pas robuste vis-à-vis des erreurs de modélisation. Ici, l’erreur statique constatée provient de l’imperfection du modèle utilisé et particulièrement de la mauvaise connaissance de ses paramètres. La constante de gravitation  $g$ , le gain  $k_v$  et la distance  $x_0$

étant bien identifiés sur le banc d’essai, l’erreur de modélisation découle de l’incertitude sur le paramètre  $\theta = kk_v^2 / m$ . Une analyse des réponses trouvées montre que ce paramètre a été surestimé. Cela peut correspondre soit à une sous estimation de la masse du pendule  $m$  soit à une surestimation du coefficient  $k$ .

Le matériel ainsi que le banc d’essai utilisé ne permettent pas d’identifier ce paramètre  $\theta$  avec une très grande précision. La solution serait donc de concevoir une loi de commande plus robuste qui permet de pallier à ce problème d’incertitude.

Dans ce qui suit, deux solutions seront proposées. La première consiste à ajouter une action intégrale dans le loi de commande et la deuxième se base sur le principe d’une commande adaptative.

### 6.5.6. Synthèse d’une commande Backstepping à action intégrale par retour de sortie

Une première solution au problème d’annulation de l’erreur statique de la commande Backstepping par retour de sortie consiste à introduire une action intégrale lors de la synthèse de cette loi de commande. Pour ce faire, les fonctions de Lyapunov seront modifiées.

Les nouvelles étapes de synthèse de la loi de commande sont données par :

#### Étape 1 :

Considérons de nouveau le sous-système

$$\dot{x}_1 = x_2 \quad (6-84)$$

Soient la première variable d’erreur  $e_1 = x_1 - r$  et la fonction de Lyapunov assignable :

$$V(e_1) = \frac{1}{2}(e_1^2 + \kappa q^2) \quad \text{avec} \quad \kappa > 0 \quad (6-85)$$

où  $q = \int_0^t e_1(\tau) d\tau$  est un nouvel état introduit.

La dérivée temporelle de la fonction de Lyapunov assignable (6-85) est donnée par

$$\dot{V} = e_1 \dot{e}_1 = e_1(x_2 + \kappa q) \quad (6-86)$$

Si on choisit la première commande virtuelle stabilisante

$$x_2^{des} = -c_1 e_1 - \kappa q = \alpha(x_1) \quad \text{avec} \quad c_1 > 0 \quad (6-87)$$

Nous obtenons ainsi :  $\dot{V} = -c_1 e_1^2 \leq 0$ .

#### Étape 2 :

La deuxième variable d’erreur est alors :  $e_2 = x_2 - \alpha$ . Le système se réécrit alors sous la nouvelle forme :

$$\begin{cases} \dot{e}_1 = e_2 + \alpha \\ \dot{e}_2 = -g + \theta \lambda(e_1) u^2 + c_1(e_2 + \alpha) + \kappa e_1 \end{cases} \quad (6-88)$$

où  $\lambda(e_1) = (x_0 - r - e_1)^{-2}$ .

Nous augmentons la fonction de Lyapunov assignable de la première étape par un terme pénalisant l'erreur entre la vitesse  $x_2$  et sa valeur désirée  $x_2^{des} = \alpha$  :

$$V_1(e_1, e_2) = \frac{1}{2}(e_1^2 + c_3 e_2^2) \quad (6-89)$$

Cette fonction est définie positive et sa dérivée temporelle est donnée par :

$$\begin{aligned} \dot{V}_1 &= e_1 \dot{e}_1 + c_3 e_2 \dot{e}_2 \\ &= e_1(e_2 + \alpha) + c_3 e_2(-g + \theta \lambda(e_1) u^2 + c_1(e_2 + \alpha) + \kappa e_1) \\ &= e_1 \alpha + c_3 e_2(e_1/c_3 - g + \theta \lambda(e_1) u^2 + c_1(e_2 + \alpha) + \kappa e_1) \end{aligned} \quad (6-90)$$

Si on choisit

$$u^2 = \frac{1}{\theta \lambda(e_1)} (-c_2/c_3 e_2 - e_1/c_3 + g - c_1(e_2 + \alpha) - \kappa e_1) \quad \text{avec } c_2 > 0 \quad (6-91)$$

on obtient  $\dot{V}_1 = -c_1 e_1^2 - c_2 e_2^2 \leq 0$ .

Ainsi, le nouveau système en boucle fermée est équivalent au système linéaire :

$$\begin{cases} \dot{q} = e_1 \\ \dot{e}_1 = -\kappa q - c_1 e_1 + e_2 \\ \dot{e}_2 = -(e_1 + c_2 e_2)/c_3 \end{cases} \quad (6-92)$$

où on voit bien l'apparition du terme intégral. Les nouveaux paramètres de commande deviennent alors  $c_1, c_2, c_3$  et  $\kappa$ .

En couplant le même observateur de la vitesse (cf. section 6.5.5.1) avec la nouvelle loi de commande (6-91), nous obtenons le système en boucle fermée équivalent suivant :

$$\begin{cases} \dot{q} = e_1 \\ \dot{e}_1 = -\kappa q - c_1 e_1 + e_2 + \varepsilon \\ \dot{e}_2 = -e_1/c_3 - c_2/c_3 e_2 + (c_4 + c_1)\varepsilon \\ \dot{\varepsilon} = -c_4 \varepsilon \end{cases} \quad (6-93)$$

La stabilité de ce nouveau système est prouvée dans l'annexe 2 (cf. section 8.2.2).

Nous remarquons que la dynamique de l'observateur est toujours découplée de celles du correcteur. Ceci permet d'appliquer le principe de séparation.

### 6.5.6.1. Simulation et validation expérimentale

En adoptant les mêmes problèmes d’optimisation qu’auparavant avec les paramètres de commande  $c_{i=1,\dots,4}$  et  $\kappa$ , l’algorithme d’optimisation AGU aboutit aux résultats suivants :

$T_r$ (s)	$u$ (V)	$[c_1, c_2, c_3, c_4, \kappa]_{opt}$
0.1977	[1.4e-5, 2.1997]	[56.351, 22.111, 1.010, 1218.064, 147.699]

Nous remarquons que la spécification concernant le temps de réponse n’est pas vérifiée. En effet, nous avons pas pu prouvé la faisabilité du cahier des charges pour la nouvelle structure de la loi de commande avec notre approche par optimisation non différentiable.

Les réponses temporelles du système en boucle fermée sont données par la figure ci-dessous :

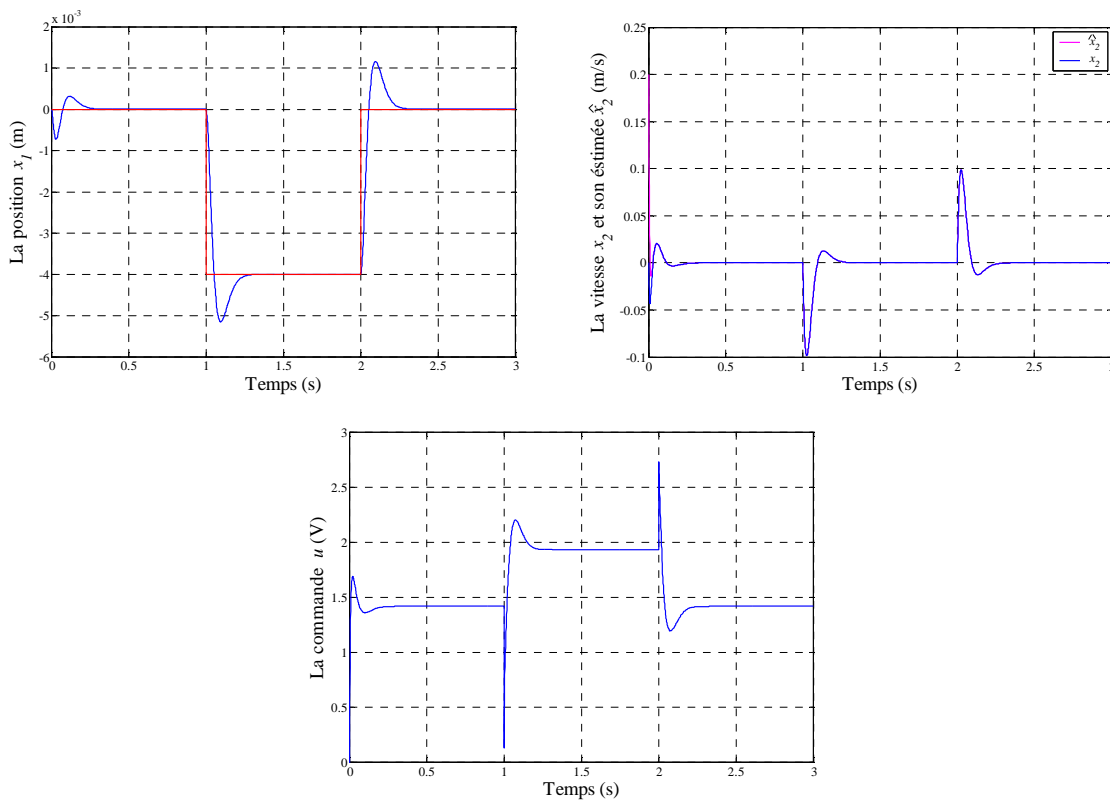


Fig. 6.38 – Résultats de simulation de la commande Backstepping à action intégrale par retour de sortie

Ils montrent que le cahier des charges n’est pas rempli : le dépassement est supérieur à 20% et le temps de réponse est de l’ordre de 200 ms. Ceci peut être expliqué par le fait que l’ajout d’une action intégrale réduit la bande passante du système en boucle fermée l’empêchant ainsi à atteindre des régimes rapides. Parallèlement, l’optimisation du critère (6-72) ne favorise pas l’amélioration du dépassement qui se trouve en compromis directe avec l’amélioration du temps de réponse.

Toutefois, l’ajout de l’action intégrale a permis de répondre à une spécification supplémentaire du cahier des charges qui est le rejet de perturbations. En effet, en appliquant deux perturbations constante  $i_0$  à l’entrée du système aux instants 1.5 et 2.5 secondes nous obtenons les résultats suivants :



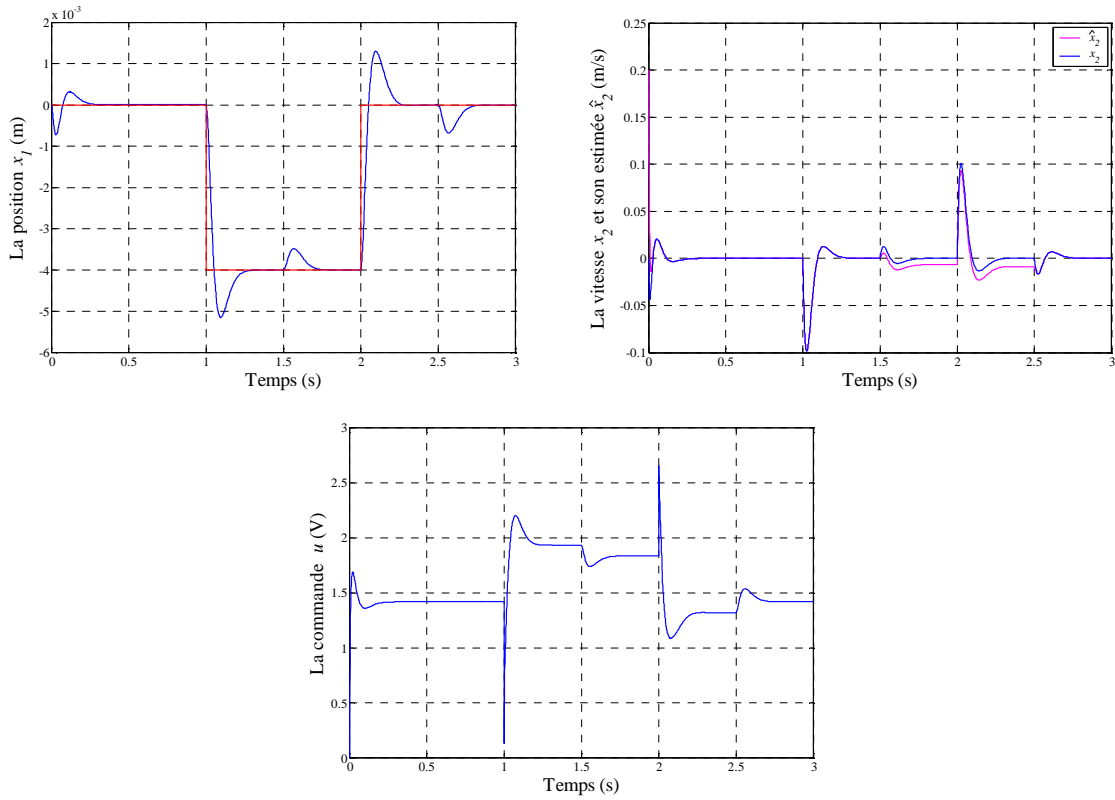


Fig. 6.39 – Rejet de perturbations de la commande Backstepping à action intégrale par retour de sortie

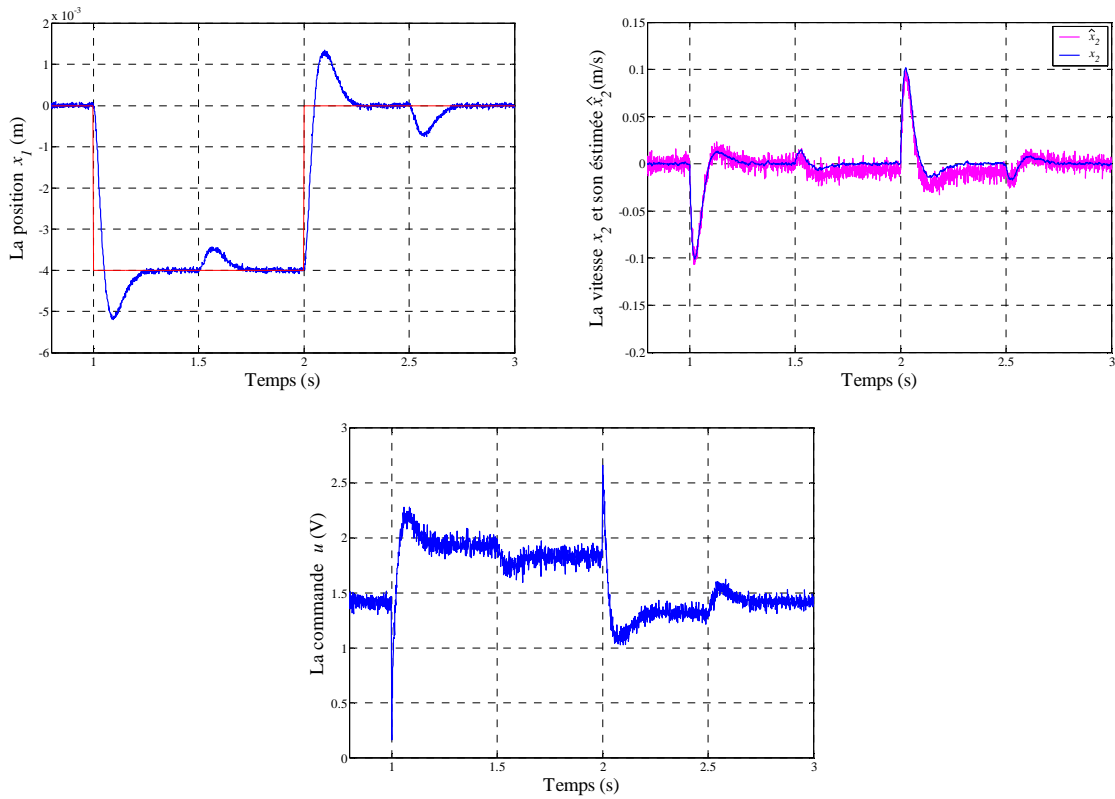


Fig. 6.40 – Résultats expérimentaux de la commande Backstepping à action intégrale par retour de sortie

Dans un premier temps, nous remarquons que les résultats expérimentaux sont en parfaite concordance avec les résultats de simulation. Nous observons aussi que même si l'observateur de la vitesse ne permet pas de l'estimer correctement (apparition d'une erreur statique), le rejet (annulation) de perturbations est parfaitement assuré. En effet, l'utilisation de l'action intégrale a permis de compenser l'effet de cette erreur que l'observateur ne pouvait pas annuler car il a été synthétisé à base d'un modèle ne tenant pas compte de la nouvelle entrée  $i_0$ .

### 6.5.7. Commande Backstepping par retour de sortie adaptative

Dans cette partie, nous allons, par une approche Backstepping, synthétiser une commande par retour de sortie adaptative à base de l'observateur d'ordre réduit de la vitesse déjà trouvé. L'approche proposée est celle de la commande à gain inconnu (cf. section 8.1.4.4).

Il s'agit d'adapter le principal gain de la loi de commande précédemment synthétisée afin qu'elle puisse compenser l'incertitude paramétrique sur le principal paramètre  $\theta = kk_v^2 / m$  supposé parfaitement connu dans le modèle initial.

Ainsi, la loi de commande (6-91) devient :

$$\begin{aligned}
 u^2 &= \frac{\hat{\zeta}}{\lambda(e_1)} (-c_2 / c_3 e_2 + g - e_1 / c_3 - c_1 (c_4 (e_1 + r) + (e_2 + \alpha))) \\
 &= \frac{\hat{\zeta}}{\lambda(e_1)} (-c_2 / c_3 e_2 + g - e_1 / c_3 - c_1 (e_2 - c_1 e_1))
 \end{aligned}
 \tag{6-94}$$

où  $\hat{\zeta} = \hat{\theta}^{-1}$ .

L'emploi de cette valeur estimée  $\hat{\zeta}$  au lieu du paramètre inconnu nous mènera encore à l'idée de pénaliser l'écart entre l'estimée et sa valeur réelle. Par conséquent, nous construisons la fonction de Lyapunov assignable suivante :

$$V_4 = V_3 + \frac{\theta}{2\gamma} \tilde{\zeta}^2 \quad \text{avec} \quad \gamma > 0
 \tag{6-95}$$

où  $\tilde{\zeta} = \theta^{-1} - \hat{\zeta}$ .

Sa dérivée est donnée par :

$$\begin{aligned}
 \dot{V}_4 &= e_1 (c_4 (e_1 + r) + \alpha + \varepsilon) + c_3 e_2 (e_1 / c_3 - g + \theta \lambda(e_1) u^2 + c_1 (c_4 (e_1 + r) + (e_2 + \alpha))) + (c_4 + c_1) \varepsilon \\
 &\quad - (1/d_1 + 1/d_2) \varepsilon^2 - \frac{\theta}{\gamma} \tilde{\zeta} \dot{\hat{\zeta}}
 \end{aligned}
 \tag{6-96}$$

En remplaçant la commande définie par (6-94), nous obtenons :

$$\begin{aligned} \dot{V}_4 &= e_1(-c_1 e_1 + \varepsilon) + c_3 e_2(e_1 / c_3 - g + \theta \hat{\zeta})(-c_2 / c_3 e_2 + g - e_1 / c_3 - c_1(c_4(e_1 + r) + (e_2 + \alpha))) \\ &\quad + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon - (1/d_1 + 1/d_2)\varepsilon^2 - \frac{\theta}{\gamma} \tilde{\zeta} \dot{\hat{\zeta}} \\ &= e_1(-c_1 e_1 + \varepsilon) + c_3 e_2(e_1 / c_3 - g + (1 - \theta \tilde{\zeta})(-c_2 / c_3 e_2 + g - e_1 / c_3 - c_1(c_4(e_1 + r) + (e_2 + \alpha)))) \\ &\quad + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon - (1/d_1 + 1/d_2)\varepsilon^2 - \frac{\theta}{\gamma} \tilde{\zeta} \dot{\hat{\zeta}} \end{aligned} \quad (6-97)$$

Après simplification, nous obtenons :

$$\begin{aligned} \dot{V}_4 &= e_1(-c_1 e_1 + \varepsilon) + c_3 e_2(-c_2 / c_3 e_2 + (c_4 + c_1)\varepsilon) - (1/d_1 + 1/d_2)\varepsilon^2 \\ &\quad - \theta \tilde{\zeta} \left( c_3 e_2(-c_2 / c_3 e_2 + g - e_1 / c_3 - c_1(c_4(e_1 + r) + (e_2 + \alpha))) + \dot{\hat{\zeta}} / \gamma \right) \end{aligned} \quad (6-98)$$

Le lecteur peut reconnaître la première ligne de l'expression  $\dot{V}_4$  qui est égale à  $-q(e_1, e_2, \varepsilon)$  (cf. section 8.2.1). Il nous reste alors de choisir l'estimateur  $\hat{\zeta}$  afin que la dérivée de la fonction de Lyapunov totale  $V_4$  soit négative. Ici, nous choisissons d'annuler le deuxième terme,

$$c_3 e_2(c_2 / c_3 e_2 - g + e_1 / c_3 + c_1(c_4(e_1 + r) + (e_2 + \alpha))) = \frac{\dot{\hat{\zeta}}}{\gamma} \quad (6-99)$$

et nous obtenons l'estimateur paramétrique suivant :

$$\dot{\hat{\zeta}} = \gamma c_3 e_2(e_1 / c_3 - g + c_2 / c_3 e_2 + c_1(c_4(e_1 + r) + (e_2 + \alpha))) \quad (6-100)$$

### 6.5.7.1. Analyse et simulation du système en boucle fermée

Le schéma bloc de la nouvelle boucle de commande est donné par la figure ci-dessous :

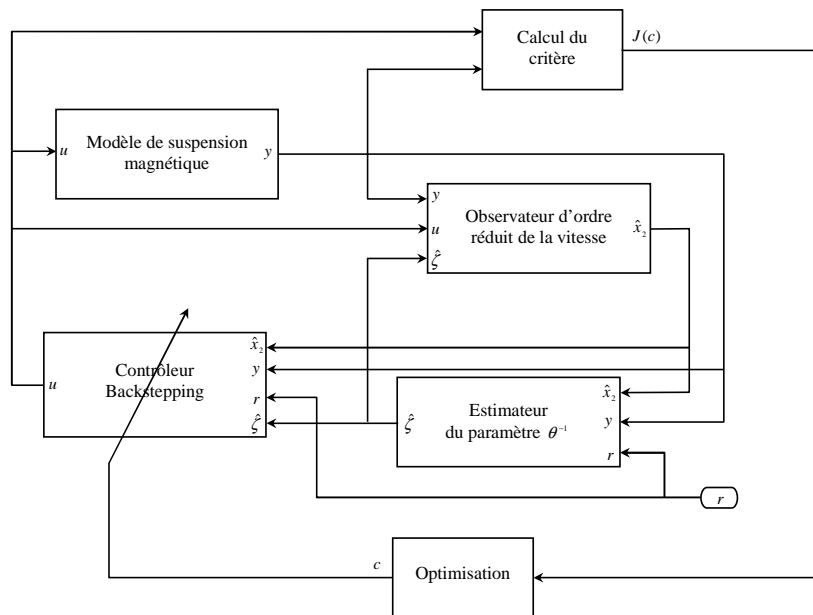


Fig. 6.41 – Schéma bloc de la boucle fermée (système+observateur+contrôleur Backstepping adaptative)

Les performances de la nouvelle loi de commande dépendent donc des paramètres  $c_{i=1,\dots,4}$  et de nouveau paramètre  $\gamma$ .

La nouvelle boucle fermée est non linéaire, elle nécessite beaucoup plus de précaution lors de la simulation et de l'évaluation de la fonction coût. La méthode "cvsode" de la boîte à outils SundialsTB a été adoptée pour résoudre le système différentiel de la boucle fermée.

En solutionnant les mêmes problèmes d'optimisation (6-68 et 6-72) pour la nouvelle loi de commande (6-94), les résultats de simulation obtenus sont donnés par la figure ci-dessous :

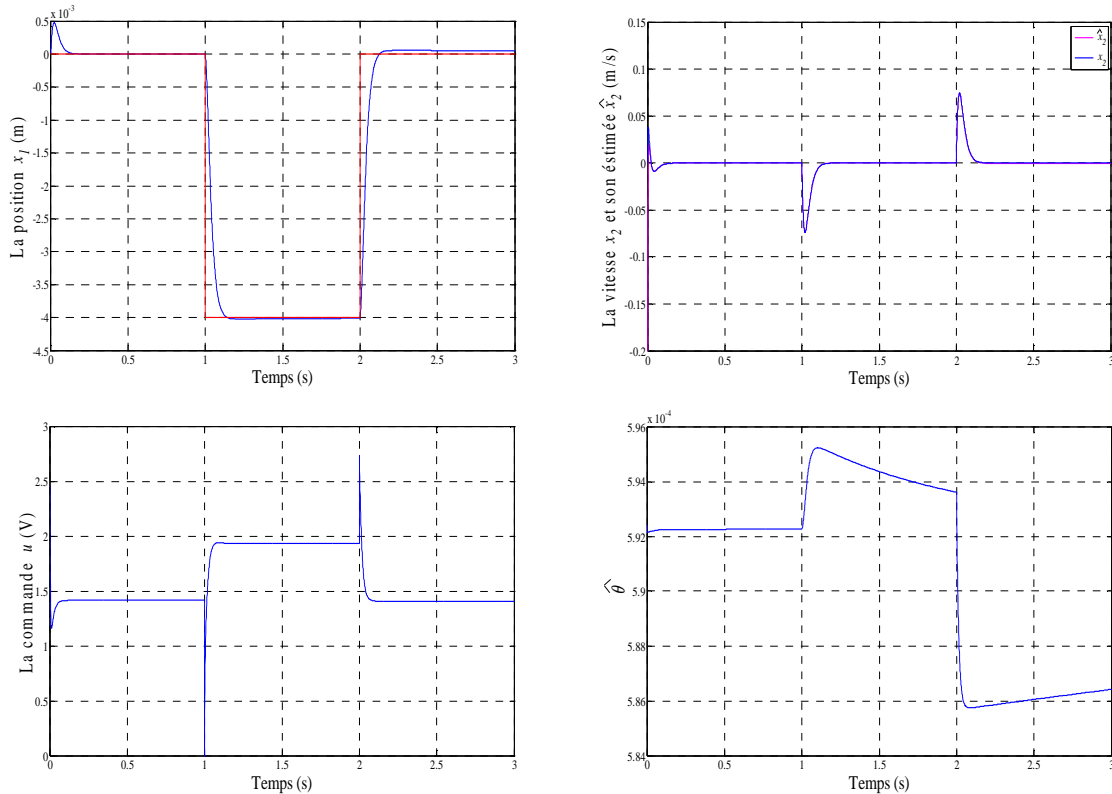


Fig. 6.42 – Résultats de simulation de la commande Backstepping par retour de sortie adaptative

Les réponses sont asymétriques : même si les dépassements sont inférieurs à 5%, la commande est plus lente lorsqu'il s'agit d'un échelon positif. Le nouveau temps de réponse optimal obtenu est de 85.1 millisecondes. Il est 12% meilleur que celui de la commande Backstepping non adaptative. Cette amélioration n'est possible qu'au dépend d'une commande à la limite de la saturation : l'échelon négatif provoque un effet de phase non minimale sur le signal de commande qui frôle la limite 0 V. Les nouveau paramètres de commande sont donnés par :

$T_r$ (s)	$u$ (V)	$[c_1, c_2, c_3, c_4, \gamma]_{opt}$
0.0851	[1e-5, 2.0197]	[50.703, 33.141, 0.707, 314.091, 102.334]

Nous remarquons, ici, que l'effet du paramètre  $c_3$  sur les résultats d'optimisation est plus important. Il permet d'atteindre des régimes très peu oscillatoires et très rapides ce qui améliore les performances de la commande synthétisée.

### 6.5.7.2. Implémentation et validation expérimentale

Les relevés du banc d'essai correspondants à la commande Backstepping par retour de sortie adaptative sont donnés par la figure 6.43.

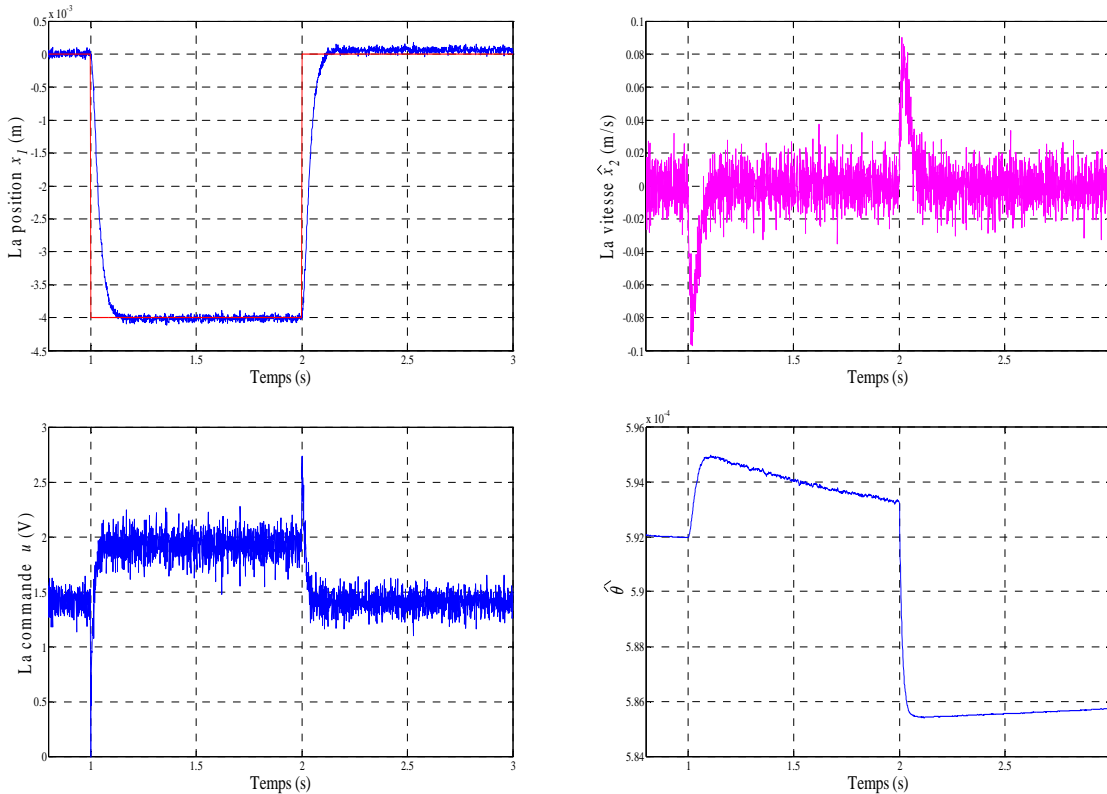


Fig. 6.43 – Réponses expérimentales de la commande Backstepping par retour de sortie adaptative

Les relevés temporels concordent fortement avec les résultats de simulation. Toutefois, nous remarquons deux petites différences : un léger retard dans le temps de réponse (95 millisecondes au lieu de 85.1) et un petit offset entre les courbes d'estimation du paramètre  $\theta$ .

Même si l'estimée  $\hat{\theta} = \hat{\gamma}^{-1}$  n'atteint pas son régime stationnaire rapidement, cet offset reste visible. Toutefois, il n'y'a aucune garantie qui permet de conclure à la convergence de l'estimée  $\hat{\theta}$  vers sa valeur réelle  $\theta$ . En effet, la condition (6-99) n'assure qu'une limitation de l'erreur  $\tilde{\theta} = \hat{\theta} - \theta$ .

Par la suite, nous testons la commande développée pour un objectif de rejet d'une perturbation constante. L'implémentation de ce test montre les résultats de la figure 6.44.

La commande synthétisée reste performante et rejette la perturbation. Comme pour l'asservissement de la position, le rejet de la perturbation se fait plus rapidement pour des déplacements négatifs. Cette différence de dynamiques entre les déplacements positifs et négatifs est surtout liée à l'asymétrie du système et au pari que nous avons pris en choisissant les échelons négatifs comme le pire cas des performances atteignables. En optimisant le correcteur pour cette catégorie de références, la commande est devenue moins performante (ici moins rapide) pour le cas des déplacements positifs.

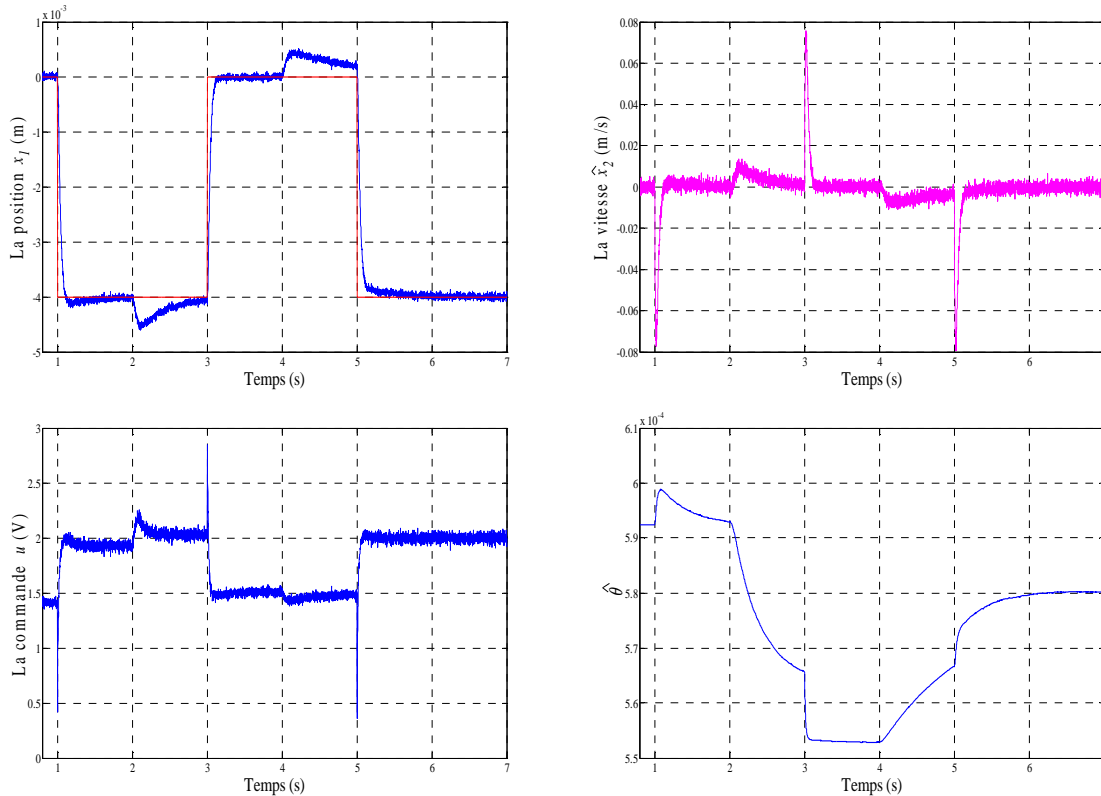


Fig. 6.44 – Rejet de perturbations de la commande Backstepping par retour de sortie adaptative

Une solution à ce problème serait d’effectuer la synthèse (donc l’optimisation) de la loi de commande sur un profil type de références qui excite le maximum de dynamiques possibles. Cette solution est souvent adoptée pour les systèmes industriels non linéaires peu variants ou invariants dans le temps.

Les résultats de cette loi de commande restent néanmoins prometteuses car nous avons pu prouver la faisabilité du cahier des charges avec une loi de commande adaptative réaliste.

### 6.5.8. Prise en compte de la dynamique de l’actionneur

Dans cette partie, le modèle du système de suspension magnétique, utilisée jusqu’à ici, est remis en cause. On s’intéresse en particulier à la partie actionneur qui présente une dynamique non négligeable et qui risque de réduire les performances de loi de commande synthétisée.

En réalité, l’actionneur peut être modélisé par un circuit RL régi par l’équation différentielle de premier ordre suivante :

$$\frac{\partial i}{\partial t} = \frac{-R}{L} i + \frac{1}{L} u \quad (6-101)$$

La constante de temps  $\tau = L/R$  était initialement négligée sous l’hypothèse que  $R \gg L$ . Néanmoins, l’identification expérimentale entrée/sortie de l’organe de puissance montre que cette dynamique est loin d’être insignifiante :  $\tau = 10^{-3}$  (s).

Par conséquent, en prenant le vecteur d’état comme étant  $x = [z, \dot{z}, i]^T$ , le système peut être décrit par le système d’états suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -g + \frac{k}{m} \frac{x_3^2}{(x_0 - x_1)^2} \\ \dot{x}_3 = (-x_3 + k_v u) / \tau \\ y = \beta x_1 \end{cases} \quad (6-102)$$

De ce système (6-102), résulte le point d'équilibre  $x_* = [x_{1*}, 0, \sqrt{mg/k(c - x_{1*})}]^T$  dépendant d'une position donnée  $x_{1*}$  et d'une commande à l'équilibre  $u_* = \tau/k_v x_{3*}$ .

### 6.5.8.1. Retouche de la commande Backstepping par retour de sortie adaptative

Afin d'éviter le parcours de toutes les précédentes étapes de synthèse, nous proposons de retoucher la loi de commande précédemment synthétisée. Il s'agit d'appliquer directement la loi de commande (6-94) au système (6-102) et d'ajuster les paramètres de commande afin que le cahier des charges reste toujours vérifié et que les résultats expérimentaux et de simulation soient plus concordants.

En optimisant ses paramètres, la loi de commande (6-94) produit les résultats de simulation suivants :

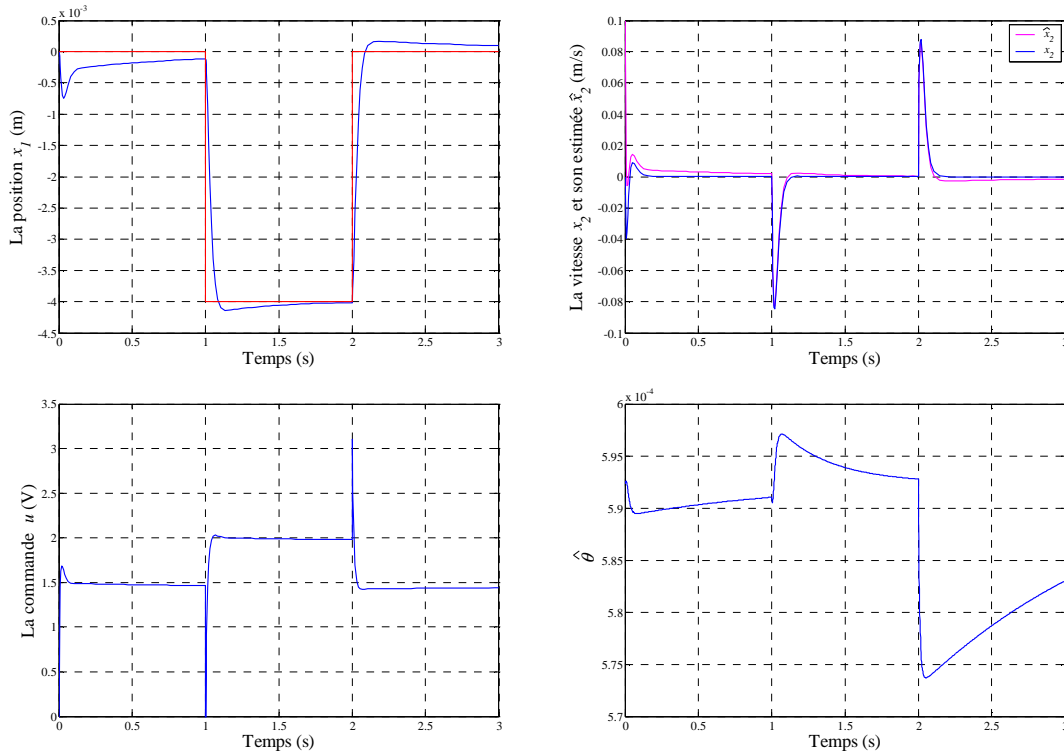


Fig. 6.45 – Résultats de simulation de la commande Backstepping par retour de sortie adaptative

Nous observons que les réponses du système en boucle fermée sont légèrement plus rapides en phase de montée mais elles mettent beaucoup plus de temps pour atteindre la stationnarité. L'observateur de la vitesse est biaisé car il a été synthétisé à base d'un modèle simplifié où la dynamique de l'actionneur était négligée. Cependant l'aspect adaptatif du gain de la commande (6-94) permet de réajuster automatiquement les réponses du système pour que la position  $x_1$  gagne bien sa référence. Les nouveaux paramètres et performances de la commande (6-94) sont donnés par :

$T_r$ (s)	$u$ (V)	$[c_1, c_2, c_3, c_4, \gamma]_{opt}$
0.0859	[2.1e-5, 2.0187]	[49.665, 46.208, 0.849, 70.478, 601.207]

Les figures 6.46 et 6.47 représentent les résultats d'implémentation sur le banc d'essai. Cette fois, les réponses temporelles sont en parfaite concordance avec les courbes de simulation.

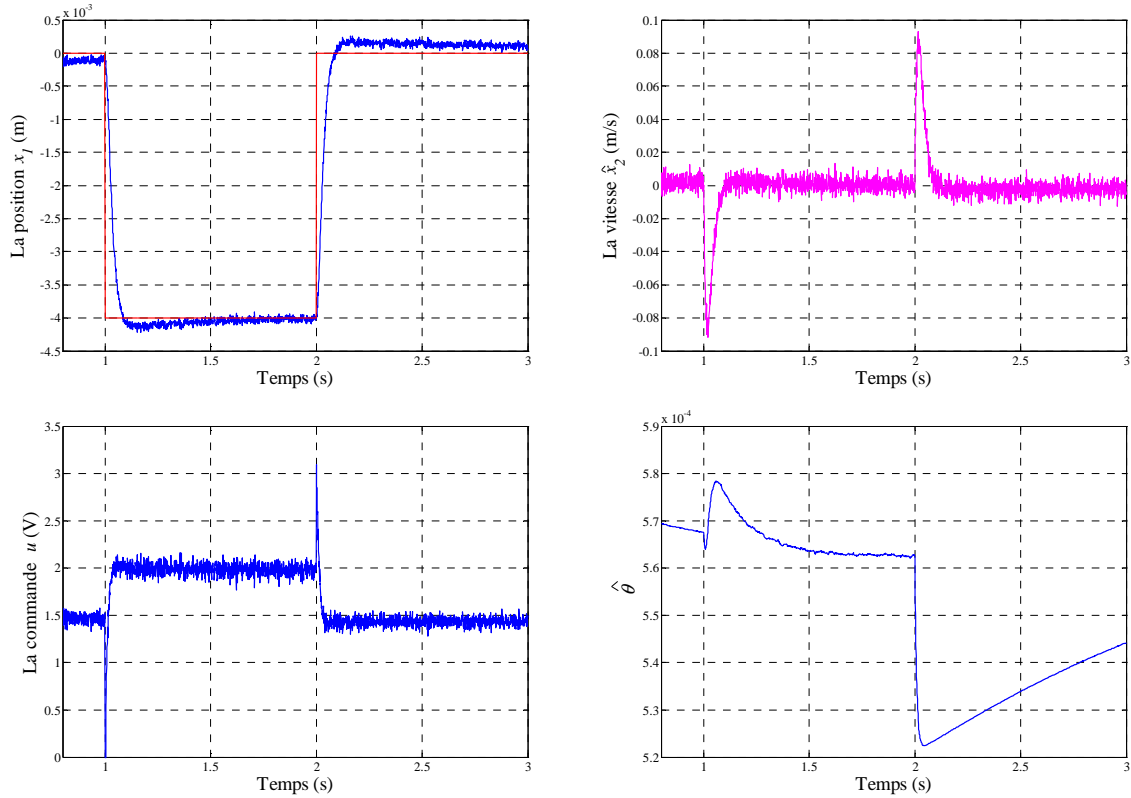


Fig. 6.46 – Réponses expérimentales de la commande Backstepping par retour de sortie adaptative

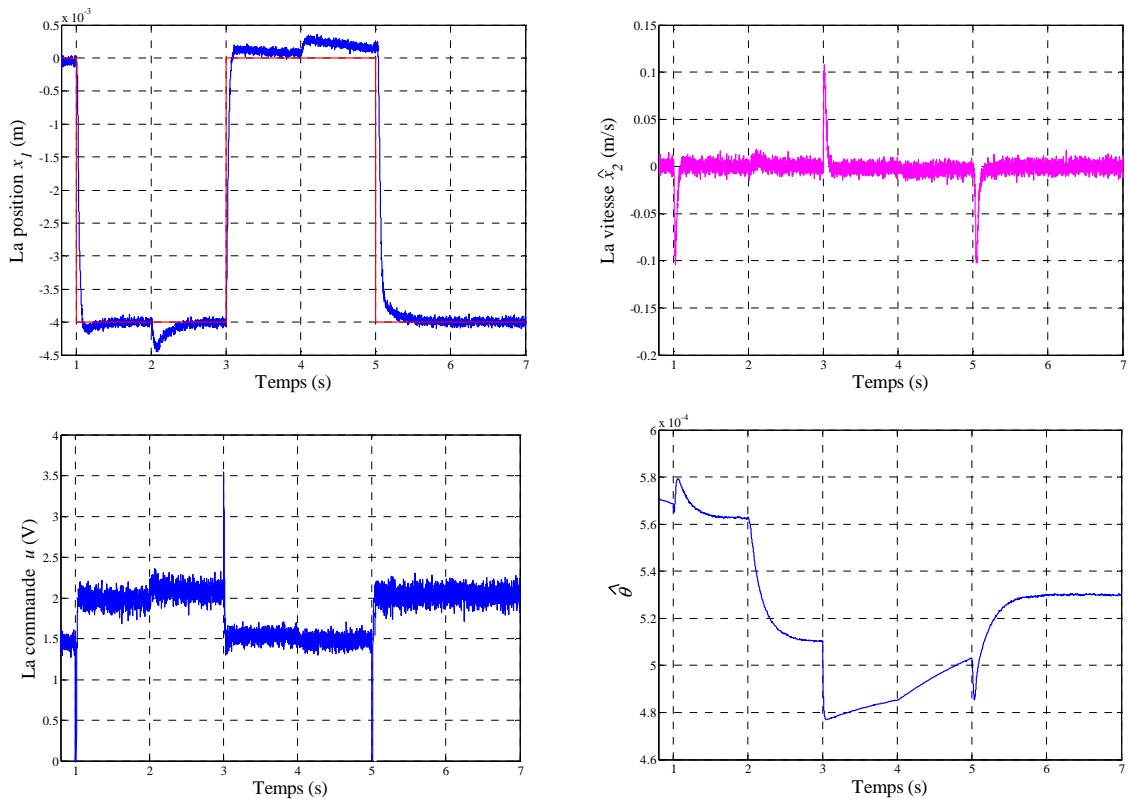


Fig. 6.47 – Rejet de perturbations de la commande Backstepping par retour de sortie adaptative



Sur la figure 6.46 et pour la réponse en position, nous prélevons un dépassement de 5% et un temps de réponse de 90 millisecondes environ. Ces résultats sont très proches à ceux trouvés par simulation (4.4% de dépassement et 87 millisecondes de temps de réponse).

Les tests de rejet de perturbations confirment les résultats déjà obtenus sans considération de la dynamique de l'actionneur : le rejet est plus lent lorsque il s'agit d'une perturbation positive. Les variations de l'estimée  $\hat{\theta}$  à chaque modification de la consigne ou de la perturbation prouvent le bon caractère adaptatif de la loi de commande synthétisée.

In fine, les résultats obtenus pour cet exemple montrent que même pour un problème de commande simple, le problème de validation d'un cahier des charges nécessite plusieurs étapes et qu'il faut disposer d'outils de synthèse spécifiques capables de gérer l'aspect multicritère des spécifications.

Les résultats obtenus par l'approche de synthèse par optimisation non différentiable montrent que l'algorithme AGU et les outils de calcul numérique développés sont efficaces et peuvent présenter un bon outil d'aide à la conception de correcteurs et à la validation de cahiers des charges. Leur combinaison avec des méthodes algébriques de type Backstepping fournit un outil global puissant permettant de prendre en compte les aspects non linéaires d'un modèle.

## 6.6. Commande d'un système de forage pétrolier

Un système de forage est un ensemble structurel dont l'objectif premier est de détruire de la matière, en général de la roche, afin de forer un puits. La structure de forage peut être perçue comme une poutre qui tourne en surface, à vitesse constante, et dont l'extrémité, fore par l'intermédiaire d'un outil. Sous certaines conditions de fonctionnement, la vitesse de rotation de cet outil présente un comportement irrégulier généré par la présence des frottements secs, phénomène que l'on appelle plus communément le "stick-slip" ou encore le "collé-glissé".

Ce phénomène se traduit par une oscillation de la vitesse de rotation, l'outil s'arrête et repart soudainement en mouvement pour atteindre des vitesses supérieures au double de la vitesse moyenne imposée en surface. L'effet du "stick-slip" sur les équipements du forage est l'un des problèmes les plus importants à résoudre. En effet, les oscillations engendrées peuvent souvent provoquer des niveaux de couples capables d'endommager le système de forage, rendant ainsi l'opération relativement coûteuse. Elles sont par ailleurs réputées pénalisantes vis-à-vis de l'avancement axial qui représente la performance instantanée du système. Différentes études ont été réalisées dans la littérature pour réduire ce phénomène à l'aide de plusieurs formes de lois de commande [Ser02, Tuc99, Smi95].

Dans la plus part de ces travaux, déjà existants, l'approche linéaire est prépondérante. On y trouve principalement des lois de commande de type Proportionnel Intégral (PI) [Tuc99], de type  $H_{\infty}$  basées sur des modèles linéaires tangents [Ser02], ou des régulateurs issus des techniques de commande optimale [Smi95]. En revanche, très peu de travaux de recherche ont été effectués pour traiter ce problème épineux par des approches non linéaires, car l'étude dans le domaine linéaire n'a pas pu conduire à des résultats très satisfaisants.

A l'heure actuelle, l'évolution matérielle des techniques d'implantation des stratégies de contrôle numérique permet de faire évoluer les lois de commande basées sur des correcteurs classiques PI, PID pour s'intéresser aux lois de commande plus avancées et pouvant potentiellement offrir un peu plus de performances.

Dans ce contexte, cette application traite du problème d'asservissement non linéaire du système de forage rotary afin d'évaluer le niveau d'amélioration des performances.

Le travail réalisé dans cette étude entre dans le cadre d'une collaboration interne avec Dr. Farag Abdulgalil et Pr. Houria Siguerdidjane dont l'objectif consiste à apporter une réponse au problème du "stick-slip". Pour cela, une synthèse puis une optimisation d'une loi de commande de type Backstepping atténuant cet effet sont proposées. La contribution de ce travail comporte alors deux idées principales :

- La synthèse d'une loi de commande Backstepping appropriée, permettant d'atteindre les performances demandées (en choisissant bien la structure).
- Mesure de la limite des performances atteignables du système en boucle fermée par optimisation non différentiable.

Ce travail de collaboration s'est consolidé par la rédaction d'une communication [Las06] (cf. annexe 3). Cette communication a été présentée à la conférence IFAC Workshop on Control Applications of Optimisation, Paris-Cachan, France, 2006 et qui est aussi apparue dans la revue International Journal of Tomography and Statistics, vol. 6, No. S07, pp. 134-139, 2006.

Le papier de cette étude peut être consulté dans l'annexe 3 de ce mémoire. Dans ce qui suit, on se contentera juste de comparer les performances du système bouclé obtenues par l'algorithme AGU avec les performances réalisées par les stratégies de contrôle de la littérature [Abd06, Ser02, Tuc99, Smi95]. Pour plus de détails sur cette application, le lecteur peut se reporter aux travaux de thèse de F. Abdulgalil [Abd06] et à l'annexe 3 du présent document.

Stratégie de commande	Temps de réponse	Dépassement	Commande maximale
PI	32 (s)	18 %	$7 \cdot 10^6$ (Nm)
$H_{\infty}$	12 (s)	22 %	$15 \cdot 10^5$ (Nm)
Linéarisation entrée/état	19 (s)	nul	$12 \cdot 10^5$ (Nm)
Linéarisation entrée/état + PID	15 (s)	nul	$14 \cdot 10^5$ (Nm)
Backstepping	2 (s)	nul	$12 \cdot 10^5$ (Nm)

Tab. 6.24 – Comparaison des performances de plusieurs lois de commande

Le tableau 6.25 reporte les valeurs des différents paramètres de comparaison obtenues par les méthodes de commande exposées dans [Abd06] avec celles existantes dans la littérature. Ce tableau permet de constater que les performances temporelles du système bouclé ont été améliorées par rapport aux réponses présentées dans la littérature.

On observe, à travers ces résultats, que le contrôleur par Backstepping conduit à une réponse sans oscillations avec le meilleur temps de réponse (de l'ordre de 2 secondes) et des amplitudes faibles. Les résultats de simulation de contrôleur sont donnés par les figures de l'annexe 3. Ils montrent l'efficacité de l'approche par optimisation utilisée, en particulier vis-à-vis à des critères de type gabarit temporel.

## 6.7. Commande d'un missile fortement manœuvrant

Cette section présente également la mise en œuvre des outils de synthèse proposés dans cette thèse sur un problème benchmark typique de l'aéronautique. Il s'agit d'un problème de commande de missile. Il peut être considéré comme un benchmark car, après sa publication en 1992 par Reichert [Rei92], il a fait l'objet de nombreux travaux (voir par exemple [Nic93, Sha93, Hua95, Bal96, Dev02]).

Après avoir présenté le problème de commande associé et analysé le comportement du système en boucle ouverte, nous synthétisons des lois de commande non linéaire par poursuite asymptotique. Nous appliquons par la suite l'approche de retouche de correcteur proposée dans ce mémoire pour améliorer la performance de la loi de commande synthétisée.

Le caractère fortement non linéaire du missile en boucle ouverte justifie l'utilisation de la méthode de commande non linéaire développée dans ce rapport. Elle est effectivement mise en œuvre et les performances de la loi de commande obtenue sont comparées avec ceux des lois de commande présentées par Devault et al. [Dev02].

### 6.7.1. Objectifs de la commande

Nous considérons comme application la commande de la voie de tangage d'un missile fortement manoeuvrant mono axe, c'est-à-dire la commande d'un missile suivant le plan vertical. Celle-ci est constituée de deux boucles imbriquées :

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

C'est la seconde boucle qui nous intéresse ici, en ne considérant que les mouvements du missile dans le plan vertical. Le système est représenté sur la figure 6.48.

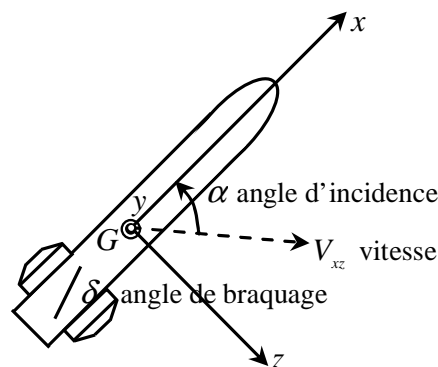


Fig. 6.48– Missile dans le plan vertical

$G$  désigne le centre de gravité. Le trièdre  $(G_x, G_y, G_z)$  est lié au missile. Le plan vertical est celui défini par  $(G_x, G_z)$ .  $V_{xz}$  est la projection de la vitesse du missile par rapport au vent sur le plan vertical. L'angle  $\alpha$  entre  $V_{xz}$  et  $G_x$  est appelé l'angle d'incidence. On introduit la vitesse de rotation  $q$  du missile autour du point  $G$  avec les conventions habituelles d'orientation.

Le but du correcteur de pilotage est de suivre des demandes d'accélération  $\eta_c$  suivant  $G_z$  en échelon en agissant sur la déflection (angle de braquage)  $\delta$  de la gouverne correspondante. Le modèle missile que nous considérons possède des caractéristiques intéressantes par rapport à ce qui a été vu précédemment : il est (fortement) non linéaire et ses performances sont difficilement interprétables via les paramètres de commande. De plus, il a été étudié de façon intensive dans la littérature, ce qui nous permettra d'évaluer nos lois de commande en les comparant aux lois de commande précédemment proposées. Pour plus de détails sur la description d'un missile, se rapporter aux thèses [Fer95, Fro95a]. Dans ce qui suit, nous allons considérer le modèle proposé par Reichert dans l'article [Rei92] pour un missile volant à 20000 pieds d'altitude pour un nombre de Mach égal à 3.

### 6.7.2. Modèle du missile

Le modèle considéré suppose que le missile est rigide et que les actionneurs sont linéaires. D'autre part, le nombre de Mach apparaît comme un paramètre extérieur variant dans le temps. En prenant comme variables d'état  $\alpha$  et  $q$  pour le missile, nous avons la représentation d'état :

$$\begin{cases} \dot{\alpha} = K_\alpha M C_n(\alpha, \delta, M) \cos(\alpha) + q \\ \dot{q} = K_q M^2 C_m(\alpha, \delta, M) \end{cases} \quad (6-103)$$

avec les coefficients aérodynamiques de portance qui sont donnés en fonction de  $\alpha$ ,  $\delta$  et le nombre de Mach  $M$  par :

$$\begin{cases} C_n(\alpha, \delta, M) = a_n \alpha^3 + b_n |\alpha| \alpha + c_n (2 - M/3) \alpha + d_n \delta \\ C_m(\alpha, \delta, M) = a_m \alpha^3 + b_m |\alpha| \alpha + c_m (-7 + 8M/3) \alpha + d_m \delta \end{cases} \quad (6-104)$$

Nous pouvons considérer qu'étant déterminés expérimentalement, les coefficients aérodynamiques sont entachés d'erreurs. De plus, on peut estimer que leur domaine de validité est défini par des angles d'incidence appartenant à une plage de  $\pm 0,35$  radians.

La sortie à commander (mesurée) est l'accélération normale. Elle est donnée par :

$$\eta = K_\eta M^2 C_n(\alpha, \delta, M) \quad (6-105)$$

La vitesse de rotation  $q$  est aussi mesurée. La dynamique de l'actionneur est modélisée par un système de second ordre :

$$\ddot{\delta} = -\omega_a^2 \delta - 2\xi_a \omega_a \dot{\delta} + \omega_a^2 \delta_c \quad (6-106)$$

La signification des différentes grandeurs est donnée dans le tableau 6.25. Le reste des paramètres sont des constantes. Leurs valeurs sont données par le tableau 6.26.

$\alpha$	Angle d'incidence
$q$	Vitesse de rotation dans le plan ( $G_x, G_z$ )
$M$	Nombre de Mach
$\delta$	Angle de gouverne
$\delta_c$	Commande de l'angle de gouverne
$\eta_c$	Consigne d'accélération normale en g
$\eta$	Accélération normale en g

Tab. 6.25 – Différentes grandeurs du modèle

$a_n$	19,3470 (rad <sup>-3</sup> )	$P_0$	4,6618.10 <sup>4</sup> (kg.m <sup>-2</sup> )
$b_n$	-31,0084 (rad <sup>-2</sup> )	$S$	0,0409 (m <sup>2</sup> )
$c_n$	-9,7174 (rad <sup>-1</sup> )	$m$	204.0241 (kg)
$d_n$	-1,9481 (rad <sup>-1</sup> )	$V$	315.8947 (rad.s <sup>-1</sup> )
$a_m$	40,5193 (rad <sup>-3</sup> )	$d$	0.2286 (m)
$b_m$	-64,1657 (rad <sup>-2</sup> )	$I_y$	247.4384 (kg.m <sup>2</sup> )
$c_m$	2,9221 (rad <sup>-1</sup> )	$K_\alpha$	0,7 $P_0 S / m / V$
$d_m$	-11,8029 (rad <sup>-1</sup> )	$K_q$	0,7 $P_0 S d / I_y$
$\omega_a - \xi_a$	150 (rad.s <sup>-1</sup> ) - 0,7	$K_\eta$	0,7 $P_0 S / mg$

Tab. 6.26 – Valeurs numériques des paramètres du modèle

### 6.7.3. Cahier des charges

Le cahier des charges est celui proposé par Reichert [Rei92] ;

– pour une consigne  $\eta_c$  en échelon, le temps de montée (à 63% de la valeur finale) doit être de l'ordre de 0,35 secondes, le dépassement inférieur à 10% et l'erreur statique inférieure à 1% ;

– éviter la saturation des actionneurs à la fois en position et en vitesse : cette dernière saturation, normalisée par la consigne en accélération, doit être de l'ordre de 0,44 (rad/s/g) ;

– limitation de la bande passante du correcteur à cause de la présence de modes souples de structure non modélisés (sur la boucle ouverte linéarisée atténuation de 30 db à 300 rad/s).

– robustesse à des erreurs sur les coefficients aérodynamiques des fonctions de portance :  $\pm 25\%$  sur  $C_m(\alpha, \delta, M)$ .

À ces spécifications s'ajoute donc la limitation sur l'angle d'incidence :  $-0,35 < \alpha < 0,35$ .

### 6.7.4. Commande du missile par un correcteur PI

Dans un premier temps, nous proposons de retoucher un correcteur PI tiré de la littérature [Gas88, Fro97]. Ce dernier possède une structure de deux boucles en cascade. Il a été établi en se basant sur le linéarisé tangent du système non linéaire autour d'un angle d'incidence  $\alpha_0 = 0,15$  (rad) et un nombre de Mach de 3. Dans la suite de cette étude, les résultats de cette loi de commande seront utilisés comme références.

La structure de ce correcteur PI est représentée par la figure ci-dessous :

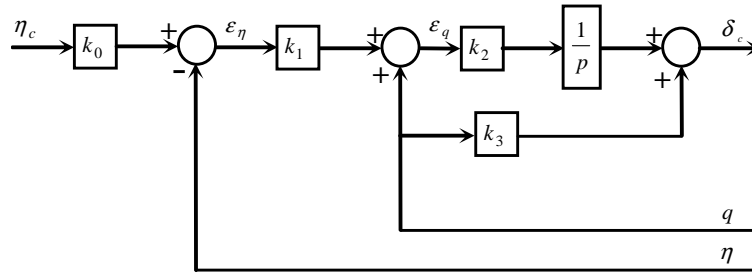


Fig. 6.50 –Structure du contrôleur PI

L'expression du correcteur correspondant est donnée par :

$$u(p) = \frac{k_3 p + k_2}{p} q(p) + \frac{k_1 k_2}{p} (k_0 \eta_c(p) - \eta(p)) \quad (5-110)$$

Un premier bon réglage a été obtenu dans [Fro97]. Les paramètres de commande donnés sont :

$$k_0 = 1.12, \quad k_1 = 0.0867, \quad k_2 = 2.5 \quad \text{et} \quad k_3 = 0.5 \quad (6-111)$$

La valeur de  $k_0$  a été obtenue en remarquant qu'à l'équilibre, le signal entrant dans l'intégrateur devient nul. En effet, nous avons :

$$\varepsilon_q = k_1 (k_0 \eta_c - \eta_e) + q_e = 0 \quad (6-112)$$

D'autre part, des équations (6-103) du missile, nous avons :

$$q_e \approx -\frac{K_\alpha}{K_\eta M} \eta_e \quad (6-113)$$

De ces deux équations, nous en déduisons que  $\eta_e = \eta_c$  si

$$k_0 \approx \frac{K_\alpha}{K_\eta M k_1} + 1 = 1.12 \quad (6-114)$$

Ceci réduit le nombre de paramètres de loi de commande à 3 paramètres seulement :  $k = [k_1, k_2, k_3]$ .

Les simulations du système en boucle fermée ont été réalisées pour une séquence d'échelons tirée de [Dev02]. Cette séquence est plus riche que celle utilisée dans l'article de base de Reichert [Rei92] et permet d'atteindre un domaine de vol plus large, ce qui permet d'évaluer la performance et la robustesse de la loi de commande synthétisée.

Les réponses temporelles issues de la simulation du correcteur PI sur le système non linéaire, (cf. figure 6.51), montrent que le temps de montée à 63% (la constante de temps du système en boucle fermée par abus de langage) est inférieur à 0,3 secondes, l'erreur statique nulle et le dépassement de l'ordre de 16%. De plus, la contrainte de saturation sur la vitesse de déflexion est largement respectée alors que la limite inférieure sur l'angle d'incidence  $\alpha$  est quasiment atteinte pour le premier échelon.

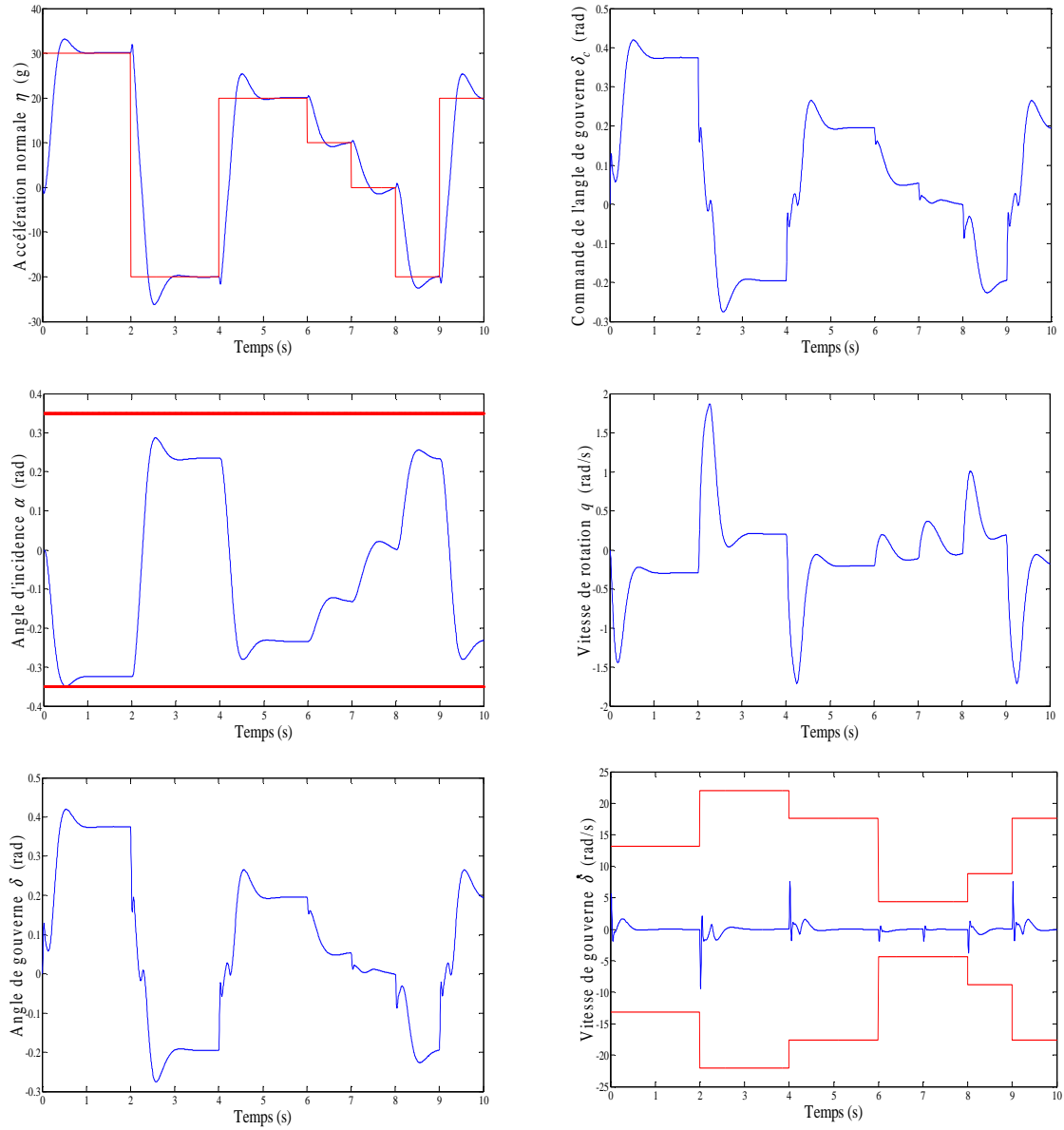
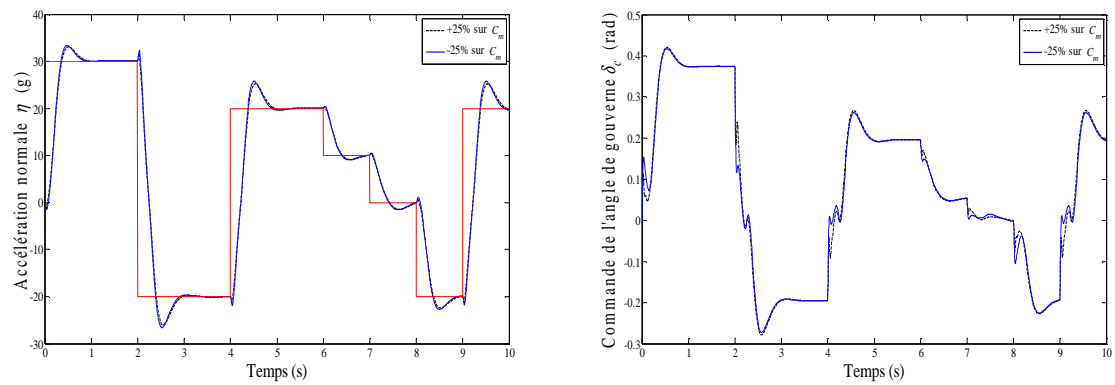


Fig. 6.51 – Correcteur PI : réponses temporelles du système nominal



(...)

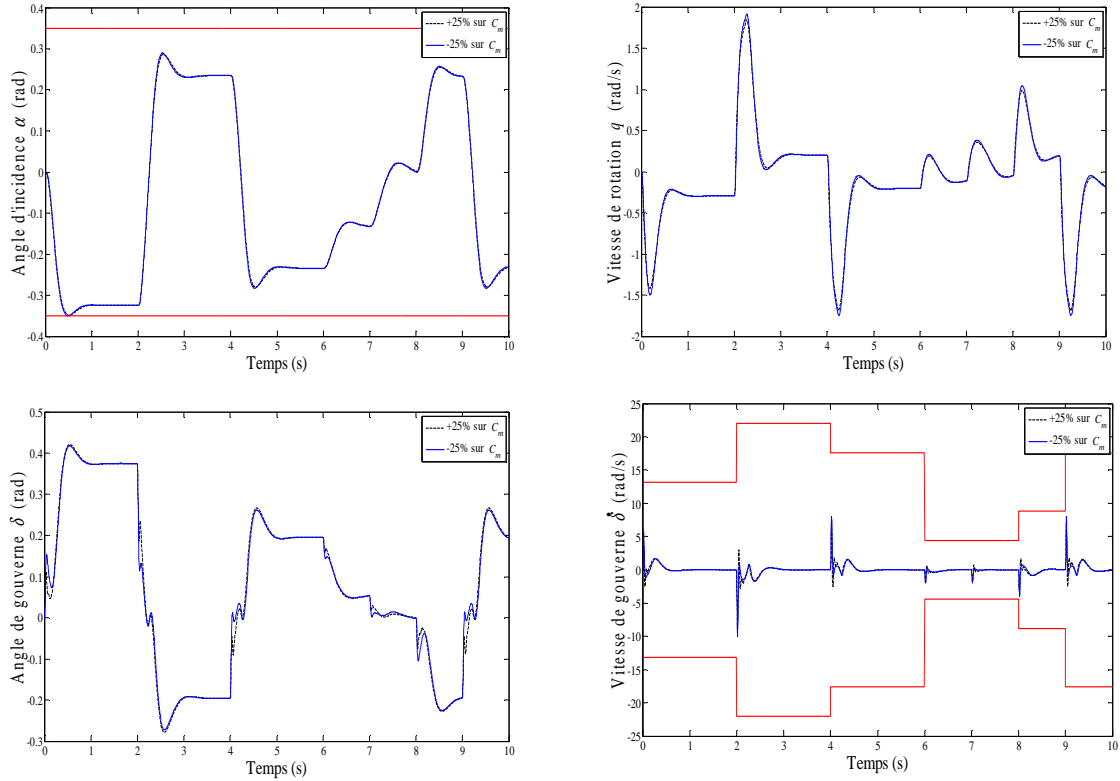


Fig. 6.52 – Correcteur PI : réponses temporelles du système incertain ( $\pm 25\%$  sur  $C_m(\alpha, \delta, M)$ )

En outre, en faisant une simulation avec une perturbation  $C_m(\alpha, \delta, M)$  de plus ou moins 25%, le système bouclé garde un comportement correct très proche du cas nominal. Ainsi, le correcteur PI peut être qualifié de robuste vis-à-vis des erreurs sur le coefficient aérodynamique  $C_m(\alpha, \delta, M)$ .

Dans [Fro97, Sco97], les auteurs proposent une démonstration des performances de ce correcteur classique via une approche d'analyse basée sur la norme incrémentale. Ces performances qui restent d'ailleurs maintenues même pour des variations paramétriques du modèle non linéaire. En effet, il a été observé que lorsque le pire cas des linéarisations de ce système est correctement commandé, le système non linéaire l'est également. Il n'est alors pas nécessaire de faire appel à des méthodes de séquençage de gains sophistiquées.

Dans ce qui suit, nous proposons de retoucher le correcteur PI pour que la sortie  $\eta$  atteigne des régimes plus rapides avec les mêmes spécifications du cahier des charges. Pour ce faire, nous traduisons les contraintes en la minimisation du critère de pénalisations suivant :

$$J_k = T_r(\eta(t, k)) + |M_{dB}(300) + 30|_+ + \int_{t_0}^{t_r} \left( \eta_{11} |\eta(t, k) - \bar{\eta}(t)|_+ + \eta_{12} |\underline{\eta}(t) - \eta(t, k)|_+ + \eta_{21} |\alpha(t, k) - \bar{\alpha}(t)|_+ \dots \right. \\ \left. + \eta_{22} |\underline{\alpha}(t) - \alpha(t, k)|_+ + \eta_{31} |\dot{\delta}(t, k) - \bar{\delta}(t)|_+ + \eta_{32} |\underline{\dot{\delta}}(t) - \dot{\delta}(t, k)|_+ \right) dt \quad (6-115)$$

où  $M_{dB}(300)$  est le module en décibel du système en boucle ouverte.

La spécification de robustesse vis-à-vis du coefficient aérodynamique  $C_m(\alpha, \delta, M)$  est prise en compte en minimisant la somme de trois critères de type (6-115) et ceci pour les valeurs



$0,75 \cdot C_m(\alpha, \delta, M)$ ,  $C_m(\alpha, \delta, M)$  et  $1,25 \cdot C_m(\alpha, \delta, M)$ . La spécification concernant la limitation de la bande passante est prise en compte directement via un calcul sur la boucle ouverte du modèle linéarisé.

L'évaluation du critère et ses gradients est assurée via les résultats de simulation effectués sous Matlab. Nous utiliserons la fonction "cvide" de boîte à outils SundialsTB pour les réponses temporelles et le calcul formel pour évaluer le module de la boucle ouverte à la pulsation 300 rad/s.

Les résultats d'optimisation de l'algorithme AGU sont donnés par la figure (6.53). Les paramètres de commande optimaux sont donnés par :

$$k_0 = 1.1194, k_1 = 0.0867, k_2 = 3.1250 \text{ et } k_3 = 0.3000 \tag{6-116}$$

Comparant aux résultats du correcteur PI initial, le temps de réponse du système est diminué presque 3 fois : le meilleur temps de réponse réalisé est de 0.35 (s) contre 0.88 (s) pour le correcteur initial. Ce temps de réponse est surprenant, il est même égal au temps de montée fixé par le cahier des charges.

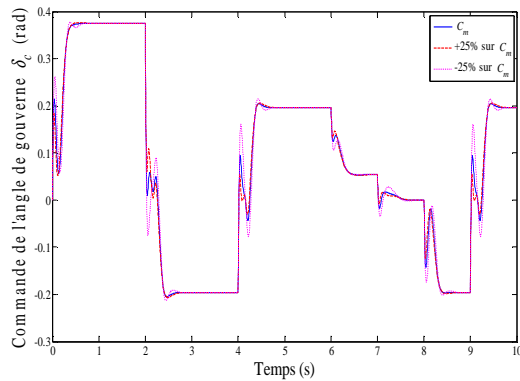
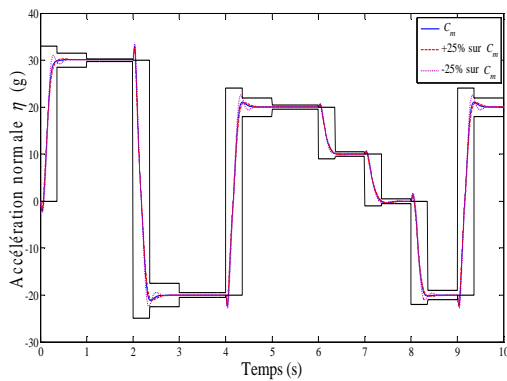
Pour le signal de référence choisi, le tableau ci-dessous compare les différentes amplitudes des signaux de la boucle de commande.

	Correcteur PI initial	Correcteur PI retouché
$\alpha(t)$	[-0.349, 0.295]	[-0.317, 0.251]
$q(t)$	[-1.702, 1.910]	[-2.431 2.810]
$\delta_c(t)$	[-0.285, 0.412]	[-0.211, 0.373]
$\max_t(\dot{\delta}(t))$	0.202	0.268
$D_{\%}(\eta)$	16%	7%

Tab. 6.27 – Comparaison des amplitudes des signaux de la boucle fermée

Nous observons que même si le système est devenu plus rapide, l'amplitude du signal de commande  $\delta_c$  est réduite tout comme l'angle d'incidence  $\alpha$ . Néanmoins, nous remarquons que l'actionneur est plus sollicité en vitesse. En outre, nous signalons l'augmentation de la vitesse de rotation du missile de plus de 42%.

Les résultats du système incertain remplissent toutes les spécifications du cahier des charges. Le système en boucle fermée est presque insensible aux incertitudes sur  $C_m$ .



(...)

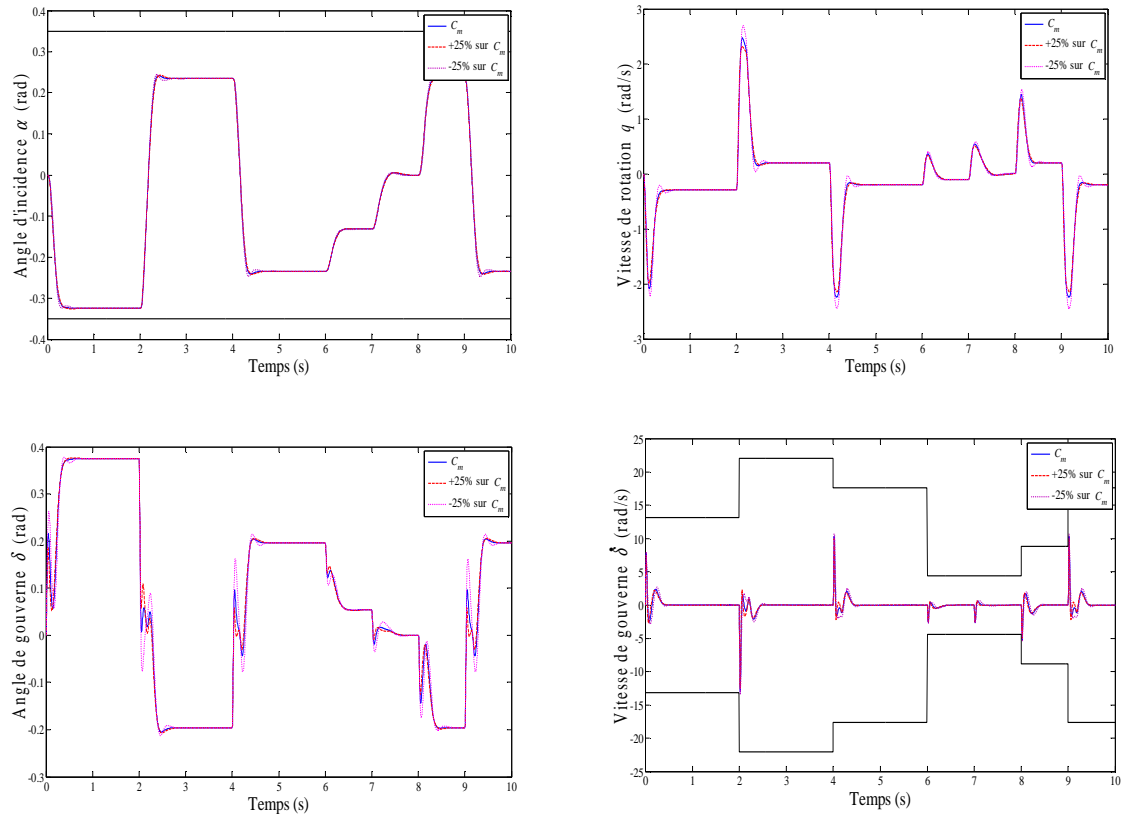


Fig. 6.53 – Correcteur PI retouché : réponses temporelles du système avec et sans incertitudes

### 6.7.5. Commande linéarisante par retour d'état dynamique

Dans cette section, nous évaluons les performances d'une commande par retour d'état dynamique. La théorie de cette commande est abondamment traitée dans les ouvrages de la commande non linéaire [Kha02, Iso95a, Nij90]. Nous nous contentons ici à rappeler les principales étapes de calcul de cette loi de commande pour notre système.

Cette technique a été initialement appliquée sur ce système par [Dev02]. Notre étude vient retoucher la commande synthétisée dans ce travail en ajustant ses paramètres et en évaluant ses performances par rapport aux résultats du correcteur PI.

#### 6.7.5.1. Calcul de la loi de commande

Avant d'entamer la synthèse de la loi de commande, nous rappelons les propriétés clef du système non linéaire traité.

Le système (6-103) est à déphasage non minimal. En effet, en examinant la dynamique des zéros, si  $\eta = 0$ , ceci implique  $C_m = 0$ , la dynamique restante sera donc :

$$\ddot{\alpha} = K_q M^2 C_m (\alpha, \delta, M) \quad (6-107)$$

En conséquence, l'angle de déflexion  $\delta$  qui maintient la sortie  $\eta$  à zéro est :

$$\delta = -\frac{1}{d_n} [a_n \alpha^3 + b_n \alpha |\alpha| + c_n (2 - M/3) \alpha] \quad (6-108)$$

En utilisant l'expression de  $C_m$ , il vient pour  $M = 3$

$$\ddot{\alpha} = K_q M^2 ((a_m - d_m a_n / d_n) \alpha^3 + (b_m - d_m b_n / d_n) \alpha) + (c_m - d_m c_n / d_n) \alpha \quad (6-109)$$

On montre ainsi que cette équation différentielle, satisfaite pour  $\alpha$ , présente un point de selle autour de l'équilibre  $(0,0)$  et deux autres points d'équilibre en dehors du domaine requis défini par :  $-0,35 < \alpha < 0,35$  (cf. figure 6.49).

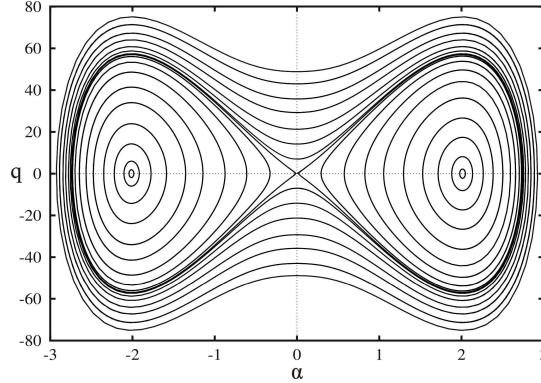


Fig. 6.49 – Plan de phase du système (6-109)

Par conséquent, la méthode de linéarisation entrée-sortie n'étant pas applicable directement sur le système étudié, nous procédons alors de la manière qui suit.

En posant  $z_1 = \alpha$ ,  $z_2 = \dot{\alpha}$  et  $z_3 = \ddot{\alpha}$ , on peut réécrire les équations d'état du système sous la forme canonique suivante<sup>2</sup> :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = F(z_1, z_2, z_3, \delta, \delta_c, \dot{\delta}_c) \end{cases} \quad (6-117)$$

où la fonction  $F$  est donnée sous la forme

$$F = a_1(z_1)z_2 + a_2(z_1)z_2^2 + a_3(z_1)z_3 + a_4(z_1)\delta + a_5(z_1)\delta_c + a_6(z_1)\dot{\delta}_c \quad (6-118)$$

avec

$$\begin{cases} a_1(z_1) = K_q M^2 C_{m1} \\ a_2(z_1) = K_\alpha M [\cos(z_1)C_{n2} - 2\sin(z_1)C_{n1} - \cos(z_1)C'_n] \\ a_3(z_1) = K_\alpha M [\cos(z_1)C_{n1} - \sin(z_1)C'_n] \\ a_4(z_1) = K_\alpha M d_n [\cos(z_1)\omega_a^2 + 2\sin(z_1)\omega_a z_2 - \cos(z_1)z_2^2 - \sin(z_1)z_3] - K_q M^2 d_m \omega_a \\ a_5(z_1) = \omega_a [K_\alpha M (-\cos(z_1)d_n \omega_a - 2\sin(z_1)d_n z_2) + K_q M^2 d_m] \\ a_6(z_1) = K_\alpha M \cos(z_1)d_n \omega_a \end{cases} \quad (6-119)$$

et pour  $j = m, n$  et  $\lambda = \text{sign}(z_1)$

<sup>2</sup> Ici, l'actionneur est considéré comme un modèle de premier ordre avec une constante de temps  $\tau_a = \omega_a^{-1}$ . Ce changement n'affecte pas trop le comportement du système en boucle fermée.

$$\begin{cases} C'_j = a_j z_1^3 + \lambda b_j z_1^2 + c_j z_1 \\ C_{j1} = 3a_j z_1^2 + 2\lambda b_j z_1 + c_j \\ C_{j2} = 6a_j z_1 + 2\lambda b_j \end{cases} \quad (6-120)$$

L'expression de l'angle de déflexion  $\delta$  en fonction du vecteur  $z = (z_1, z_2, z_3)$  est obtenue, en considérant l'équation pour  $z_3$ , comme étant :

$$\delta(z) = h_0(z_1)[h_1(z_1) + h_2(z_1)z_2 - z_3 + h_3(z_1)\delta_c] \quad (6-121)$$

où

$$\begin{cases} h_0(z_1) = 1/[K_\alpha M d_n (\cos(z_1)\omega_a + \sin(z_1)z_2) - K_q M^2 d_m] \\ h_1(z_1) = K_q M^2 C'_m \\ h_2(z_1) = K_\alpha M (\cos(z_1)C_{n1} - \sin(z_1)C'_n) \\ h_3(z_1) = K_\alpha M \cos(z_1)d_n \omega_a \end{cases} \quad (6-122)$$

Définissons, maintenant, le signal d'erreur  $\varepsilon = z_{1R} - z_1$  de sorte à obtenir une convergence asymptotique. On procède par dérivation jusqu'à l'apparition d'une loi de commande dynamique. Pour cela, on dérive jusqu'à l'ordre 3 :

$$\ddot{\varepsilon} + \beta_1 \dot{\varepsilon} + \beta_2 \varepsilon + \beta_3 \varepsilon = 0 \quad (6-123)$$

On obtient, après réarrangement, la loi de commande suivante<sup>3</sup> :

$$\dot{\delta}_c = k(z_1)[- \beta_3 z_1 - (\beta_2 + b_2(z_1))z_2 - (\beta_1 + b_1(z_1))z_3 - b_3(z_1)\delta_c - b_0(z_1) + v] \quad (6-124)$$

où

$$\begin{cases} b_0(z_1) = [a_4(z_1)h_0(z_1)h_1(z_1) + a_2(z_1)z_2^2] \\ b_1(z_1) = [a_3(z_1) - a_4(z_1)h_0(z_1)] \\ b_2(z_1) = [a_1(z_1) + a_4(z_1)h_0(z_1)h_2(z_1)] \\ b_3(z_1) = [a_5(z_1) + a_4(z_1)h_0(z_1)h_3(z_1)] \\ v = \ddot{z}_{1R} + \beta_1 \dot{z}_{1R} + \beta_2 z_{1R} + \beta_3 z_{1R} \\ k(z_1) = 1/a_6(z_1) \end{cases} \quad (6-125)$$

Sachant qu'en utilisant l'équation (6-121), l'équation (6-118) prend la forme

$$F = b_0(z_1) + b_2(z_1)z_2 + b_1(z_1)z_3 + b_3(z_1)\delta_c + a_6(z_1)\dot{\delta}_c \quad (6-126)$$

En résumé, le système en boucle fermée est défini par :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = b_0(z_1) + b_2(z_1)z_2 + b_1(z_1)z_3 + b_3(z_1)\delta_c + a_6(z_1)\dot{\delta}_c \\ \dot{\delta}_c = k(z_1)[- \beta_3 z_1 - (\beta_2 + b_2(z_1))z_2 - (\beta_1 + b_1(z_1))z_3 - b_3(z_1)\delta_c - b_0(z_1) + v] \end{cases} \quad (6-127)$$

La boucle sur  $\alpha$  se simplifie alors en :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = -\beta_3 z_1 - \beta_2 z_2 - \beta_1 z_3 + v \end{cases} \quad (6-128)$$

<sup>3</sup> L'implémentation de cette loi de commande nécessite la mesure de l'angle d'attaque  $\alpha$ , de sa vitesse et son accélération. Dans cette étude, ces grandeurs sont supposées mesurables.

Ce système est stable tant que le polynôme (6-123) est Hurwitz.

Après application de l'équation (6-124), le système (6-127) se réécrit sous la forme de boucle suivante :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = -\beta_3 z_1 - \beta_2 z_2 - \beta_1 z_3 + v \\ \dot{\delta}_c = k(z_1)[- \beta_3 z_1 - (\beta_2 + b_2(z_1)) z_2 - (\beta_1 + b_1(z_1)) z_3 - b_3(z_1) \delta_c - b_0(z_1) + v] \end{cases} \quad (6-129)$$

Les simulations montrent que les coefficients  $a_6(z_1)$  et  $b_3(z_1)$  sont approximativement constants. Ainsi, le système (6-129) peut être exprimé sous la forme :

$$\dot{z} = Az + Bv + D(z) \quad (6-130)$$

où le polynôme caractéristique de la matrice  $A$  est donné par :

$$P = (p + b_3 / a_6)(p^3 + \beta_1 p^2 + \beta_2 p + \beta_3) \quad (6-131)$$

Ce polynôme est Hurwitz pour les conditions suivantes :

$$\begin{cases} b_3 / a_6 > 0 \\ \beta_i > 0 \text{ pour } i = 1, 2, 3 \\ \beta_2 > \beta_3 / \beta_1 \end{cases} \quad (6-132)$$

La dynamique des zéros du système (6-117) avec la fonction  $F$  définie par (6-126), est obtenue en annulant  $z_1$  et sa dérivée  $\dot{z}_1$ . Après simplification, cette dynamique est définie par :

$$K_\alpha d_n \dot{\delta}_c + K_q M d_m \delta_c = 0 \quad (6-133)$$

Cette équation différentielle est stable car la constante  $K_q M d_m / K_\alpha d_n$  est positive. Ainsi, la boucle de linéarisation reste stable tant qu'aucun zéro instable n'est compensé.

Le schéma bloc de la boucle de commande synthétisée est donné par la figure suivante :

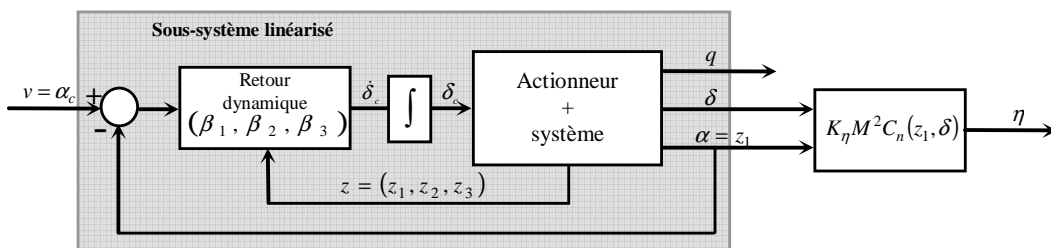


Fig. 6.54 – Schéma bloc de la commande par retour d'état dynamique

À ce stade, nous avons synthétisé un correcteur efficace pour asservir l'angle d'attaque  $\alpha$ . Néanmoins l'objectif d'autopilotage du missile reste l'asservissement de sa position via l'accélération normale de son centre de gravité. Une première solution consiste alors à inverser l'équation de la sortie pour asservir l'accélération normale  $\eta$  (cf. figure 6-54).

Toutefois, cette approche n'est pas toujours commode car elle se base sur une connaissance parfaite du système. Même si l'effet de la non linéarité issue du coefficient  $C_m$  est compensé par la boucle de linéarisation, il reste à surpasser la non linéarité due au coefficient  $C_n$ . L'éventuelle existence d'incertitudes sur ce coefficient nous impose alors d'augmenter le système linéarisé par une boucle externe en cascade avec la première comme le montre la figure 6.55. Nous avons choisi un correcteur PI classique pour assurer cette boucle externe qui vient robustifier la commande par retour d'état dynamique pour assurer un asservissement efficace de l'accélération normale du missile à sa référence souhaitée.

La commande externe du système est définie par :

$$v(p) = k_1(1 + k_2/p)(\eta_c(p) - \eta(p)) \quad (6-134)$$

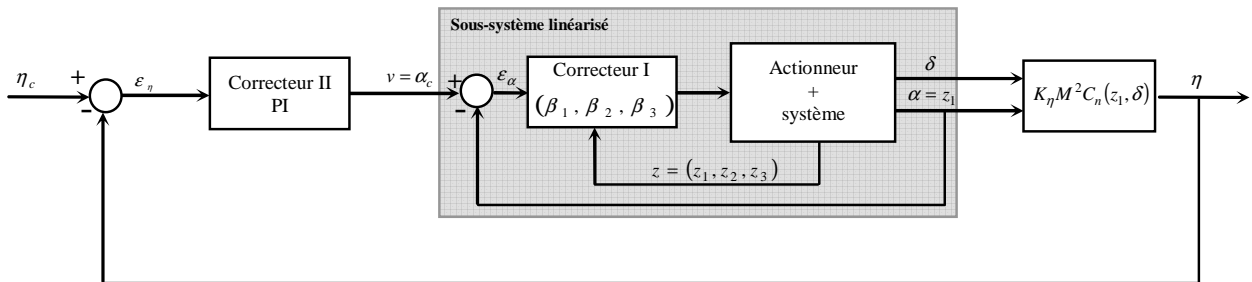


Fig. 6.55 – Schéma bloc de l'autopilotage du missile par une commande par retour d'état dynamique et poursuite asymptotique

### 6.7.5.2. Résultats de simulation

Le but est d'ajuster les paramètres de commande  $\beta_1, \beta_2, \beta_3, k_1$  et  $k_2$  afin de remplir toutes les spécifications du cahier de charges.

Nous adoptons la même approche que celle utilisée pour la retouche du correcteur PI. Nous conservons ainsi le même critère de base (6-115), et nous en construisons un global à minimiser qui est la somme de trois critères de type (6-115) pour des valeurs du coefficient  $C_m$  égales à  $0,75 \cdot C_m$ ,  $C_m$  et  $1,25 \cdot C_m$ . Ceci nous permet d'assurer une robustesse de la loi de commande.

Au critère d'optimisation s'ajoute les contraintes sur les paramètres de la boucle interne :  $\beta_1 > 0, \beta_2 > 0, \beta_3 > 0$  et  $\beta_1\beta_2 > \beta_3$ . Les trois premières inégalités sont prises en compte par reparamétrisation quadratique alors que la quatrième est considérée par l'ajout d'un terme de pénalisation exacte dans la fonction critère.

Afin de bien amorcer l'algorithme AGU, nous initialisons les paramètres de la boucle interne de façon qu'elle soit 5 fois plus rapide que le temps de réponse désirée. Cela se fait par un placement de pôles du polynôme caractéristique (6-131). Les paramètres de la boucle externe sont initialement choisis nuls.

En utilisant le même créneau de référence, les résultats d'optimisation de l'algorithme AGU sont donnés par la figure 6.56.

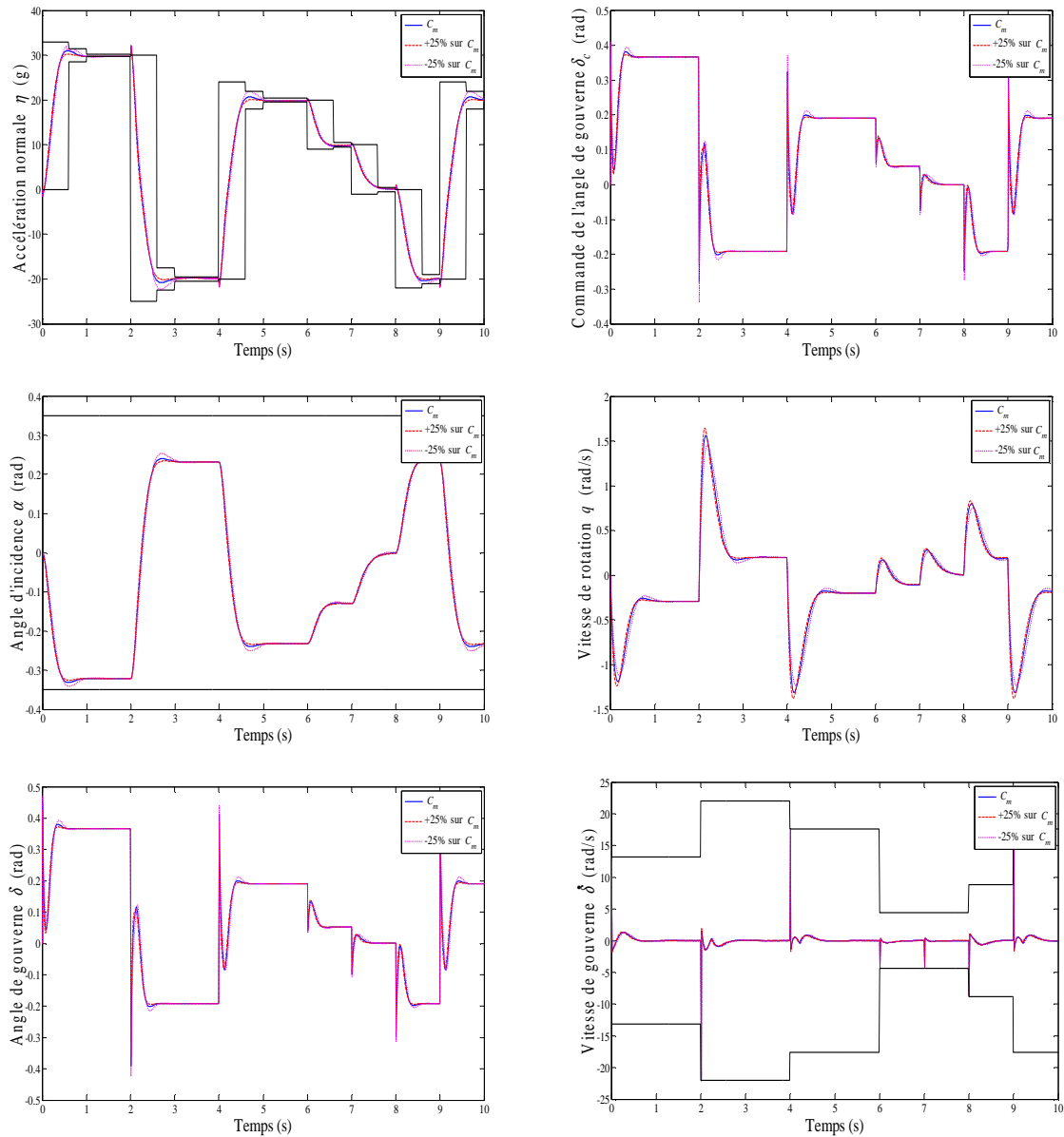


Fig. 6.56 – Résultats de la commande par retour d'état dynamique et poursuite asymptotique

Cette figure montre l'ensemble des signaux de la boucle fermée obtenus, pour d'une part le système nominale et de d'autre part le système avec des perturbations de  $\pm 25\%$  sur le coefficient aérodynamique  $C_m$ .

On observe que la commande est maintenue à l'intérieur des contraintes imposées, et cela en dépit des perturbations qui agissent sur le système. De plus, les performances requises, en termes de précision statique ( $< 1\%$ ), de temps de montée ( $< 0,35s$ ) et de dépassement ( $< 10\%$ ), sont atteintes. Bref, le cahier des charges demandé est rempli.

Le meilleur temps de réponse atteint est de 0.58 (s). Ce temps est 1.65 fois supérieur à celui du correcteur PI classique. Ceci est surtout dû aux fortes amplitudes du signal de commande  $\delta_c$  qui se trouve saturé après chaque variation du signal de consigne  $\eta_c$ . Les paramètres du correcteur optimal sont donnés par :  $[\beta_1, \beta_2, \beta_3, k_1, k_2] = [7769.941, 152759.771, 0.07106.894, -2.154, 0.041]$ .

Nous remarquons que le gain proportionnelle  $k_1$  de la boucle externe est négatif, ceci est tout à fait normal car le gain statique de la  $\alpha$ -boucle l'est aussi.

En regard des résultats obtenus, nous pouvons conclure que les performances du correcteur PI sont supérieures surtout que la mise en oeuvre de ce dernier est beaucoup plus simple que celle de la commande par retour d'état dynamique.

## 6.8. Conclusion

L'ensemble des applications traitées confirme que l'étude de la faisabilité d'un cahier des charges et l'évaluation des limites de performance atteignables par une loi de commande donnée sont des tâches difficiles. Les résultats obtenus pour une série d'exemples numériques, académiques et industriels montrent que même pour des structures de commande classiques, ces problèmes ne sont pas « immédiats » et nécessitent des méthodes d'optimisation spécifiques capables de gérer l'aspect multicritère des cahiers des charges pour pouvoir être menés à bien

Les résultats de l'approche de synthèse par optimisation non différentiable montrent que les algorithmes ASM et AGU ainsi que les outils de calcul numérique développés sont efficaces. Sur plusieurs benchmark de référence, ils donnent les meilleurs résultats actuellement connus. Dans certains cas, ce sont même les seuls algorithmes à avoir pu mener à un résultat.

Pour les applications à l'Automatique, ces algorithmes d'optimisation présentent un grand potentiel puisqu'ils permettent de prendre en compte de façon directe des critères complexes, mêlant des contraintes de types différents (contraintes temporelles, fréquentielles, gabarits, indicateurs scalaires de type 'marge'...). En combinaison avec des approches algébriques (type Backstepping), le fait de pouvoir facilement déterminer les valeurs des paramètres invite même à ajouter des degrés de liberté supplémentaires et à « oser » des designs plus variés et offrant un plus grand potentiel de réglage. Plus généralement, la combinaison de ces outils avec d'autres méthodes est naturelle puisqu'elle permet de retoucher efficacement des correcteurs pour en augmenter les performances.





# Chapitre 7

## Conclusions et perspectives

### Apports scientifiques et originalité du travail

Cette étude se situe dans le cadre d'une problématique multicritère, où le correcteur synthétisé doit répondre à plusieurs exigences issues des diverses spécifications d'un cahier des charges générique. Historiquement, divers outils mathématiques ont été développés pour la synthèse convexe. La plus répandue est la transposition des objectifs en contraintes matricielles linéaires (formulation LMI ou SDP). Le développement de ces méthodes de synthèse convexe a été surtout motivé par les avancées dans le domaine de l'optimisation convexe ce qui a fait de la formulation convexe du cahier des charges une démarche incontournable. Cependant, la difficulté majeure de cette approche réside dans la traduction appropriée des objectifs génériques du cahier des charges sous la forme mathématique ad'hoc. Bien que très puissante, la formulation LMI n'est pas en mesure de traduire tous les critères, même des critères simples et courants, simplement et directement.

Dans une démarche différente et complémentaire, notre travail de thèse se positionne dans le cadre générique où le cahier des charges est finement et fidèlement traduit. Il est transcrit, avec le minimum d'approximations, en un ensemble de critères et/ou contraintes portant sur les signaux E/S du système. Dans le cas des systèmes linéaires, cette démarche peut être facilement étendue aux critères et contraintes fréquentiels. Lors de la résolution de ce type de problème, les difficultés se reportent sur les méthodes d'optimisation non linéaire. Seules ces méthodes permettent de traiter, dans un même formalisme, des problèmes de commande et de retouche de correcteurs sous des contraintes de cahiers des charges industriels génériques de plus en plus exigeants.

L'ensemble des contraintes ainsi obtenu est très complexe et comporte des contraintes structurelles, temporelles, fréquentielles, implicites et explicites. Le problème de la faisabilité simultanée de ces contraintes n'est pas convexe et surtout pas différentiable. Il ne peut être traité qu'en explorant efficacement l'espace paramétrique de la loi de commande afin de chercher la meilleure combinaison des paramètres permettant de répondre simultanément à la totalité des contraintes du cahier des charges.

Dans cette optique, nous avons analysé un large panel des critères et des contraintes possibles en Automatique en cherchant quelles étaient les régularités des problèmes génériques qui puissent être utilisées pour obtenir une résolution efficace.

Le constat est que les problèmes d'optimisation formulés sont non différentiables par construction (vérification simultanée de plusieurs contraintes ou contraintes de type gabarit ou norme infini) et peuvent contenir des grandeurs à variation non différentiable voir même non Lipschitz (temps de

réponse, valeurs propres). Ces dernières doivent donc être évitées quand cela est possible. Vu la mesure nulle de l'espace paramétrique non différentiable, ces problèmes peuvent être qualifiés de problèmes différentiables presque partout. Nous constatons même que les problèmes sont plus réguliers que les fonctions génériques de ces espaces et que dans les cas habituels, les zones non différentiables sont des nappes (explicites ou implicites) dans l'espace des paramètres.

Bien que de mesure nulle, la construction même des problèmes fait que ces zones sont presque toujours atteintes lors des optimisations ; cela est à l'origine du mauvais fonctionnement de tous les algorithmes qui ne sont pas adaptés pour gérer cette spécificité.

Dans notre travail, et afin de pallier à ces difficultés de la formulation générique, l'effort a été surtout mis sur les méthodes de résolution de ces problèmes d'optimisation qui nécessitent des algorithmes bien spécifiques.

L'aptitude à pouvoir prendre en compte des problèmes différentiables presque partout permet d'adopter le principe de pénalisation exacte qui ne fonctionne pas bien avec les algorithmes classiques. La résolution des problèmes d'optimisation sous contraintes est alors ramenée à des problèmes d'optimisation sans contraintes sans changer de catégorie de problèmes. Ainsi, nous pouvons avoir une approche unifiée des cas avec et sans contraintes.

Pour les contraintes de limites sur les paramètres, nous avons proposé une approche par reparamétrisation très efficace basée sur un nouveau changement de variables très lisse qui simplifie la résolution des problèmes. Le cas d'une seule contrainte de borne est traité via un changement de variables quadratique.

En termes de stratégies d'optimisation, nous avons opté pour deux classes d'algorithmes d'optimisation qui se basent principalement sur la notion de descente :

Le premier algorithme développé est un algorithme de recherche directe. Il est inspiré de l'algorithme de simplexe non linéaire de Nelder-Mead. Cette technique a été choisie pour sa robustesse, éprouvée et prouvée dans la littérature, vis-à-vis des morphologies particulières des critères non lisses. L'algorithme développé est nommé ASM (Algorithme de Simplexe Modifié), il permet de surmonter le problème de dégénérescence de l'algorithme de base et de traiter également des problèmes sous contraintes de bornes.

Le deuxième algorithme est un algorithme de plus profonde descente basé sur la notion du  $\varepsilon$ -sous-différentiel au sens de Clarke. Cette notion est à la fois une généralisation de la notion du gradient aux cas non différentiables (comme un sous-différentiel) mais aussi un outil mathématique puissant qui présente des propriétés très intéressantes qui permettent de généraliser des définitions mathématiques d'objets infinitésimaux à un sens particulier non infiniment petit ("à un  $\varepsilon$  près"). Nous citons par exemple la notions de  $\varepsilon$ -stationnarité qui est la mesure de la stationnarité dans le cas de critères non différentiables pour un grain d'observation qui serait une boule de rayon  $\varepsilon$ . Cette notion présente aussi une propriété de calcul qui permet d'obtenir des approximations simplement par un nombre fini de gradients échantillonnés autour d'un point donné. Cette propriété particulière a été exploitée afin d'estimer une bonne direction de descente autour des zones où l'optimisation via une méthode de type gradient n'est plus possible (non différentiables, mal conditionnées). Plusieurs versions de cet algorithme ont été développées et présentées. La plus avancée est nommée AGU (Algorithme du

Gradient Universel). Elle se base sur une stratégie à deux phases de descente : une première phase de type quasi-Newton et une deuxième phase de recherche plus fine de type  $\varepsilon$ -sous-différentiel.

Comme les estimations de  $\varepsilon$ -sous-différentiel nécessitent elles-mêmes de nombreux calculs de gradients classiques, la charge de calcul peut être rédhibitoire sans précaution particulière. Pour être efficace, les algorithmes de type AGU doivent disposer d'une estimation précise et rapide du gradient. Nous avons pour cela étudié et développé des méthodes adaptées qui exploitent au mieux la structure des problèmes pour gagner en temps et en précision.

En particulier, nous avons mis l'accent sur les méthodes de calcul exact à base de fonctions de sensibilité paramétrique qui permettent à la fois d'évaluer le critère et son gradient en même temps que la simulation du système différentielle de la boucle fermée. Cette technique n'implique pas des calculs plus compliqués que l'approche par différences finies, bien au contraire. Elle est particulièrement efficace dans le cas des systèmes linéaires où, suite à des réductions de système, elle permet un gain drastique du temps de calcul.

Une deuxième méthode à base de calcul complexe et des propriétés des fonctions analytiques est proposée. Elle est étroitement liée aux propriétés de calcul numérique et donc au noyau de calcul utilisé. Cette technique est praticable avec tout logiciel numérique donnant directement des estimés complexes des fonctions analytiques. Elle permet d'améliorer les performances de l'approche classique par différences finies et de réduire considérablement le nombre d'évaluations de la fonction coût et donc du temps de calcul nécessaire à l'estimation des  $\varepsilon$ -sous-différentiels. Son plus grand avantage réside dans le fait qu'elle n'occasionne aucun problème au niveau du choix de la taille des perturbations scalaires qui sont utilisés pour approximer les dérivées.

L'efficacité de tous les algorithmes développés est initialement mise à l'épreuve sur une série de problèmes tests de références. Les résultats d'application sur des problèmes de commande et de retouche de correcteurs variés (linéaire et non linéaire) sont très concluants et confirment l'avantage de ces algorithmes, en particulier, par rapport aux algorithmes classiquement implémentés en optimisation non linéaire.

Parmi les différents problèmes que nous avons traités, nous avons choisi quelques exemples représentatifs et motivants l'application de l'optimisation non linéaire en Automatique.

A ce titre, la synthèse d'une loi de commande non linéaire par Backstepping, sur un système de suspension magnétique, est particulièrement représentative : dans le cas où le système doit vérifier des critères temporels précis, le résultat n'est tout simplement pas accessible sans optimisation. L'approche développée est testée dans un cadre d'une synthèse de loi de commande optimale et ensuite dans le cadre de sa retouche. Les résultats d'implémentation sont en parfaite concordance avec les simulations et prouvent l'efficacité de l'approche par optimisation non linéaire adoptée.

Globalement, notre recherche nous permet de proposer une approche qui s'avère efficace pour la retouche et même la synthèse de correcteurs pour des critères génériques. En outre, le temps raisonnable obtenu pour les synthèses permet de considérer des lois de commande plus complexes. L'approche générique permet d'offrir à l'expert un outil souple et rapide et rend possible une approche réellement générique par complexification progressive des critères.

## Perspectives

Les perspectives dans la continuation des actions menées au cours de cette thèse s'orientent dans trois principales directions :

- Le fait de travailler dans le cadre d'une approche générique sur les systèmes, avec des critères unifiés qui peuvent être traités par un code d'optimisation performant peut constituer la base d'un logiciel général de Conception Assistée par Ordinateurs (CAO) pour l'ingénieur automatique. Pour les différentes applications numériques présentées dans cette thèse, nous avons codé un ensemble de programmes et nous avons abordé les problèmes d'optimisation jusque dans ces aspects numériques. De tels programmes sont de bons prototypes à des développements de logiciels CAO.
- L'exploitation de la structure du problème pour un calcul efficace des gradients peut et doit être encore améliorée. Des approches algébriques formelles peuvent être développées pour systématiser l'écriture des opérateurs intervenant dans le calcul global des  $\varepsilon$ -sous-différentiels. Cela peut également contribuer à faire de ces approches un outil facilement accessible au concepteur de lois de commande (donc faciliter une approche CAO).
- Le calcul des  $\varepsilon$ -sous-différentiels peut très certainement être beaucoup amélioré. Rappelons que pour les problèmes différentiables presque partout, il peut être estimé à partir de plusieurs calculs de gradient en plusieurs points de l'espace. Nous avons mis au point un échantillonnage isotrope en dimension quelconque, efficace mais nécessitant de nombreux calculs de gradient. Le choix de ces échantillons peut être encore amélioré en prenant en compte les structures des critères d'optimisation et en utilisant les propriétés géométriques locales des critères autour des points d'estimation. Il y a là une étude à mener sur la géométrie locale des zones non différentiables.

# **Chapitre 8**

## **Annexes**

## 8.1. Annexe 1 : Commande et observateur par Backstepping

### 8.1.1. Introduction

Les théorèmes liés aux conditions nécessaires et aux conditions suffisantes de la stabilité ne permettent pas de guider l'utilisateur dans le choix de la fonction de Lyapunov et ainsi de conclure à la solubilité du problème de stabilité du système si on ne trouve pas une telle fonction.

Dans le domaine des systèmes non linéaires, un axe de recherche très actif est celui qui porte sur les méthodes permettant la construction et la synthèse de fonctions de Lyapunov. Les formes quadratiques, par leur simplicité, sont les plus souvent utilisées. Pour une certaine classe de systèmes non linéaires dits triangulaires ou triangularisables (cf. section 8.1.4.1), l'approche la plus adéquate est le Backstepping que nous décrivons aux paragraphes qui suivent.

### 8.1.2. Historiques

Le Backstepping a été développé par Kanellakopoulos et al. [Kan91] et inspiré par les travaux de Feuerer et Morse [Feu78] d'une part et Tsinias [Tsi89] et Kokotović et Sussmann [Kok89] d'autre part. Elle offre une méthode systématique pour effectuer le design d'un contrôleur pour les systèmes non linéaires. L'idée consiste à calculer une loi de commande afin de garantir que la dérivée d'une certaine fonction (de Lyapunov) soit définie positive et que sa dérivée soit toujours négative. La méthode consiste à fragmenter le système en un ensemble de sous-systèmes imbriqués d'ordre décroissant. Le calcul de la fonction de Lyapunov s'effectue, ensuite, récursivement en partant du sous système d'ordre le plus faible. A chaque étape, l'ordre du système est augmenté et la partie non stabilisée lors de l'étape précédente est traitée. À la dernière étape, la loi de commande est trouvée. Celle-ci permet de garantir, en tout temps, la stabilité globale du système commandé tout en travaillant en poursuite et en régulation.

### 8.1.3. Fonctions de Lyapunov assignables

Lorsque l'on s'intéresse à la stabilisation des systèmes non linéaires par les fonctions de Lyapunov et à l'analyse de cette stabilité, on dit que l'on assigne la fonction de Lyapunov. Le concept des Fonctions de Lyapunov Assignables ou "Control Lyapunov Functions" (CLF) a été introduit à la fin des années 70 par Jacobson [Jac77] et Jurdjevic et Quinn [Jur78] et que l'on peut également trouver par exemple dans la référence [Aey99].

Soit le système non linéaire

$$\dot{x} = f(x, u) \tag{8-1}$$

où  $x \in \mathfrak{R}^n$  est le vecteur d'état et  $u \in \mathfrak{R}$  est la commande vérifiant  $f(0, 0) = 0$ .

On cherche à déterminer une commande  $\alpha(x)$  telle que  $x = 0$  soit l'équilibre globalement asymptotiquement stable du système bouclé  $\dot{x} = f(x, \alpha(x))$ .

Cela revient à déterminer une commande  $u = \alpha(x)$  telle que :

$$L_f V(x) = \frac{\partial V}{\partial x} f(x, \alpha(x)) < 0 \quad (8-2)$$

où  $L_f V(x)$  est la dérivée de Lie de la fonction de Lyapunov candidate  $V(x)$  dans la direction de  $f(x, u)$ .

**Définition 8.1 (Fonction de Lyapunov Assignable)** Une fonction lisse définie positive et radialement non bornée  $V : \mathfrak{R}^n \rightarrow \mathfrak{R}^+$  est dite fonction de Lyapunov assignable (CLF) pour le système (8-1) si pour tout  $x \neq 0$

$$\exists u \in \mathfrak{R} \text{ tel que } \dot{V}(x) = V_x f(x, u) < 0 \quad (8-3)$$

Cette définition signifie que l'existence d'une CLF est équivalente à l'existence d'une loi de commande globalement stabilisante. Si un système donné possède une CLF, on peut certainement trouver une loi de commande globalement stabilisante. L'inverse est également vrai. Ceci est connu sous le nom du théorème d'Artestin [Son89].

Cependant, l'inconvénient majeur du concept CLF, comme outil de synthèse, provient de la non constructivité d'une fonction CLF pour la plupart des systèmes non linéaires. La recherche d'une fonction CLF appropriée peut s'avérer alors aussi complexe que la conception d'une loi de commande stabilisante. L'approche Backstepping résout ces deux problèmes simultanément.

Dans ce qui suit, les résultats présentés sont standards et peuvent être trouvés dans [Krs95] et [Kha02].

### 8.1.4. Backstepping

Le Backstepping est une procédure récursive qui consiste à trouver une fonction CLF, définie positive et radialement non bornée qui garantit la stabilité asymptotique globale d'un système donné [Fre95, Pra96, Sep97].

La méthodologie de construction de la fonction de Lyapunov d'un système par Backstepping a été développée au début des années 90 par P. V. Kokotovic et al. [Fre94, Fre95, Krs95]. Elle consiste à commencer par le problème de stabilisation d'un sous-système du premier ordre puis d'augmenter étape par étape l'ordre des sous-systèmes considérés jusqu'au système complet.

Avant d'introduire la commande par backstepping, nous rappelons le résultat de base de cette technique :

**Hypothèses 8.1 :** Considérons le système

$$\dot{x} = f(x) + g(x)u \quad (8-4)$$

où  $x \in \mathfrak{R}^n$  est le vecteur d'état et  $u \in \mathfrak{R}$  est la commande avec  $f(0) = 0$ . Il existe une loi de commande continuellement différentiable

$$u = \alpha(x), \quad \alpha(0) = 0 \quad (8-5)$$

et une fonction  $V : \mathfrak{R}^n \rightarrow \mathfrak{R}$  différentiable, définie positive et radialement non bornée tel que :



$$\frac{\partial V}{\partial x} [f(x) + g(x)\alpha(x)] \leq -W(x) < 0, \quad \forall x \in \mathfrak{R}^n \quad (8-6)$$

où  $W : \mathfrak{R}^n \rightarrow \mathfrak{R}$  est une fonction définie positive.

Sous ces hypothèses, nous pouvons énoncer le lemme suivant. Sa démonstration peut être trouvée dans [Krs95].

**Lemme 8.1 :** Soit le système (8-4) augmenté par un intégrateur

$$\begin{cases} \dot{x} = f(x) + g(x)\xi \\ \dot{\xi} = u \end{cases} \quad (8-7)$$

et supposons que l'équation d'état de  $x$  vérifie les hypothèses 8.1 avec  $\xi \in \mathfrak{R}$  comme commande. Alors, si  $W(x)$  est définie positive,

$$V_a(x, \xi) = V(x) + \frac{1}{2}(\xi - \alpha(x))^2 \quad (8-8)$$

est CLF pour tout le système (8-7), c'est à dire, il existe une loi de commande  $u = \alpha_a(x, \xi)$  qui rend le point d'équilibre  $(x, \xi) = (0, 0)$  de (8-7) globalement asymptotiquement stable. Une telle commande est :

$$u = -c(\xi - \alpha(x)) + \frac{\partial \alpha}{\partial x} [f(x) + g(x)u] - \frac{\partial V}{\partial x} g(x) \quad \text{avec} \quad c > 0 \quad (8-9)$$

Dans le cas où  $\dot{V}$  est semi-définie positive seulement, on peut appliquer le principe d'invariance de LaSalle pour déduire la stabilité asymptotique globale du point d'équilibre.

**Théorème 8.1 (Principe d'invariance de LaSalle)** *Considérons le système (8-7). Soit  $V(x)$  une fonction de Lyapunov vérifiant  $\dot{V}(x) \leq 0, \forall x \in \mathfrak{R}^n$ , si l'ensemble  $E = \{x \in \mathfrak{R}^n, \dot{V}(x) = 0\}$  ne contient pas de trajectoires triviales autre que l'origine, alors l'origine est un point d'équilibre globalement asymptotiquement stable.*

La loi de commande globalement stabilisante (8-9) n'est ni unique ni nécessairement la meilleure. En effet, le résultat principal du Backstepping n'est pas la forme spécifique de la loi de commande, mais plutôt la construction d'une fonction de Lyapunov dont la dérivée peut être rendue négative par une grande variété de lois de commande [Krs95, Kha02].

#### 8.1.4.1. Contraintes structurelles

Contrairement à la plupart des autres méthodes, le Backstepping n'a aucune contrainte au niveau du type de non linéarité. Cependant, le système doit se présenter sous la forme d'une structure triangulaire ou triangularisable en cascade où la commande n'intervient que dans une partie de l'état. Dans [Krs95], ces systèmes sont classés en deux catégories :

- ◆ Les systèmes à rétroaction pure "pur-feedback systems" : c'est une classe de systèmes triangulaires inférieurs sous la forme,

$$\begin{cases} \dot{x} = f(x) + g(x)\xi & \text{où } \xi \in \mathfrak{R} \\ \dot{\xi}_1 = f_1(x, \xi_1, \xi_2) \\ \dot{\xi}_2 = f_2(x, \xi_1, \xi_2, \xi_3) \\ \vdots \\ \dot{\xi}_{k-1} = f_{k-1}(x, \xi_1, \dots, \xi_k) \\ \dot{\xi}_k = f_k(x, \xi_1, \dots, \xi_k, u) \end{cases} \quad (8-10)$$

Pour que la synthèse soit réussie, le sous-système en  $x$  doit satisfaire les hypothèses 8.1. En outre, les fonctions  $f_{i=1, \dots, k-1}$  et  $f_k$  doivent être inversibles par rapport aux variables  $\xi_{i=1, \dots, k}$  et  $u$  respectivement.

◆ Les systèmes à rétroaction stricte “strict-feedback systems” : c’est des systèmes où la nouvelle variable d’état n’agit qu’affinement,

$$\begin{cases} \dot{x} = f(x) + g(x)\xi_1 \\ \dot{\xi}_1 = f_1(x, \xi_1) + g_1(x, \xi_1)\xi_2 \\ \dot{\xi}_2 = f_2(x, \xi_1, \xi_2) + g_2(x, \xi_1, \xi_2)\xi_3 \\ \vdots \\ \dot{\xi}_{k-1} = f_{k-1}(x, \xi_1, \dots, \xi_{k-1}) + g_{k-1}(x, \xi_1, \dots, \xi_{k-1})\xi_k \\ \dot{\xi}_k = f_k(x, \xi_1, \dots, \xi_k) + g_k(x, \xi_1, \dots, \xi_k)u \end{cases} \quad (8-11)$$

Les non linéarités  $f_i$  et  $g_i$  dans la  $i^{\text{ième}}$  équation d’état ( $i=1, \dots, k$ ) dépendent seulement des états  $x, \xi_1, \dots, \xi_i$ , c’est la raison pour laquelle on considère le sous-système en  $\xi$  comme une rétroaction stricte. Ce type de systèmes est plus simple pour la mise en œuvre de la synthèse de commande par Backstepping [Sep97]. C’est pourquoi il est souvent utilisé comme modèle pour le développement des résultats liés à cette technique.

#### 8.1.4.2. Commande par Backstepping

Considérons le système non linéaire à rétroaction stricte à deux variables d’état  $(x, \xi)$  :

$$\begin{cases} \dot{x} = f(x) + g(x)\xi & (s1) \\ \dot{\xi} = f_1(x, \xi) + g_1(x, \xi)u & (s2) \end{cases} \quad (8-12)$$

où  $u$  est la variable d’entrée de commande. Les fonctions  $f, g, f_1$  et  $g_1$  sont continues telles que  $g(x) \neq 0$  et  $g_1(x, \xi) \neq 0$  pour tout  $(x, \xi)$ . Quand  $u = 0$ , on suppose que le système (8-12) a pour point d’équilibre l’origine  $(0, 0)$ .

Le but de cette procédure est de commander tout d’abord le sous-système (s1) en  $x$  par l’intermédiaire de la variable  $\xi$ , appelée commande virtuelle, dont seule la dérivée est commandée par  $u$ , puis de commander le système global par  $u$ .

De ce fait, on peut trouver une commande  $u$  stabilisante à partir de la fonction de Lyapunov assignable strictement au système (8-12) et ceci étape par étape. Dans certains ouvrages, cette procédure est également appelée *Ajout de dérivateur* [Pra96].

Étape 1 : On considère le sous-système (s1), avec  $\xi$  comme étant la commande virtuelle. On suppose qu'il existe une fonction de Lyapunov  $V_x(x)$ <sup>1</sup>,  $V_x(0) = 0$  définie positive et radialement non bornée et une commande  $\alpha_x(x)$  telle que :

$$\dot{V}_x(x) = L_f V_x(x) + L_g V_x(x) \alpha_x(x) \leq -W(x) < 0 \quad (8-13)$$

Ainsi le sous-système (s1) est globalement asymptotiquement stable.

Étape 2 : On considère le système global (8-12). Pour trouver la commande stabilisante  $u$ , on introduit la fonction de Lyapunov suivante :

$$V_\xi(x, \xi) = V_x(x) + \frac{1}{2} (\xi - \alpha_x(x))^2 \quad (8-14)$$

Sa dérivée totale par rapport au temps est :

$$\dot{V}_\xi = -W(x) + (\xi - \alpha_x(x)) \left( f_1(x, \xi) + g_1(x, \xi) u - \frac{\partial \alpha_x(x)}{\partial x} (f(x) + g(x) \xi) + \frac{\partial V}{\partial x} g(x) \right) \quad (8-15)$$

Comme le but est de choisir une commande  $u$  qui rend la dérivée de la fonction de Lyapunov  $\dot{V}_\xi$  définie négative, on s'arrange pour obtenir des formes quadratiques et pour cela, on peut prendre

$$u(x, \xi) = \frac{1}{g_1(x, \xi)} \left( -c(\xi - \alpha_x(x)) - f_1(x, \xi) + \frac{\partial \alpha_x(x)}{\partial x} (f(x) + g(x) \xi) - \frac{\partial V}{\partial x} g(x) \right) \quad \text{avec } c > 0 \quad (8-16)$$

Ainsi, la fonction de Lyapunov  $V_\xi(x, \xi)$  est assignée pour assurer la stabilité asymptotique globale du système (8-12).

En procédant de manière récursive, on peut ainsi obtenir une commande par Backstepping non linéaire pour des systèmes d'ordre plus élevé.

### 8.1.4.3. *Backstepping adaptatif*

Pour les systèmes à paramètres incertains, il est possible de coupler la loi de commande synthétisée par Backstepping avec des estimations paramétriques qui permettent de l'adapter aux valeurs réelles des paramètres. Ceci peut se faire en étendant la fonction de Lyapunov par l'ajout d'un terme pénalisant l'erreur d'estimation. L'idée consiste donc à synthétiser une loi de commande par Backstepping comme si tous les paramètres du système sont connus et de les remplacer, par la suite, par leurs estimées respectives.

Afin d'illustrer cette technique, nous traitons l'exemple suivant [Krs95] :

Soit le modèle

$$\dot{x} = u + \theta x \quad (8-17)$$

où  $u$  est la commande et  $\theta$  un paramètre constant inconnu.

<sup>1</sup> L'indice  $x$  se rapporte au sous-système (s1) en  $x$ .

Le but est de d'assurer la régulation de l'état  $x(t) : x(t) \rightarrow 0, t \rightarrow \infty$ . Nous cherchons alors un estimateur  $\hat{\theta}(t)$  définie par :

$$\dot{\hat{\theta}} = \tau(x, \theta) \quad (8-18)$$

L'utilisation de cet estimateur dans la loi de commande  $u = \alpha(x, \hat{\theta})$  doit assurer la non positivité de la dérivée de la fonction de Lyapunov  $V(x, \hat{\theta})$ .

Afin de prendre en compte la dynamique de l'estimateur, nous introduisons un terme de pénalisation de l'erreur d'estimation  $\tilde{\theta}$  dans la fonction de Lyapunov. Un simple choix serait d'ajouter un terme quadratique  $\tilde{\theta}^2 / 2$  comme suit :

$$V(x, \hat{\theta}) = \frac{1}{2}(x^2 + \tilde{\theta}^2) = \frac{1}{2}x^2 + \frac{1}{2}(\hat{\theta} - \theta)^2 \quad (8-19)$$

Cette fonction est radialement non bornée et sa dérivée temporelle est donnée par :

$$\dot{V}(x, \hat{\theta}) = x(u + \theta x) + (\hat{\theta} - \theta)\dot{\hat{\theta}} = xu + \hat{\theta}\dot{\hat{\theta}} + \theta(x^2 - \dot{\hat{\theta}}) \quad (8-20)$$

Nous cherchons maintenant une loi de commande  $\alpha(x, \hat{\theta})$  et un estimateur  $\dot{\hat{\theta}} = \tau(x, \theta)$  pour garantir  $\dot{V} \leq -ax^2$  avec  $a > 0$ .

Comme  $\alpha$  et  $\hat{\theta}$  ne peuvent dépendre du paramètre incertain ou inconnu  $\theta$ , nous devons choisir :

$$\dot{\hat{\theta}} = x^2 \quad (8-21)$$

Donc, la condition

$$xu + \hat{\theta}x^2 \leq -ax^2 \quad (8-22)$$

permet de choisir la commande  $u$  telle que :

$$u = \alpha(x, \hat{\theta}) = -(a + \hat{\theta})x \quad (8-23)$$

La loi de commande (8-23) et l'estimateur (8-21) garantissent, ainsi, la stabilité de la boucle fermée quelque soit la valeur  $\theta$ .

#### 8.1.4.4. Commande à gain inconnu

Nous décrivons dans ce paragraphe une extension de l'approche Backstepping adaptatif pour loi de commande à un paramètre inconnu appelé la constante à grand gain "*high gain constant*". Dans [Krs95], ce problème est dressé sous le nom "*unknown virtual control coefficient*", où détermination de l'estimateur du paramètre inconnu nécessite seulement la connaissance de son signe.

Considérons de nouveau le système suivant :

$$\dot{x} = f(x) + g(x)u \quad (8-24)$$

La loi de commande  $u = \alpha(x)$  et la CLF  $V(x)$  assurent :

$$\dot{V} = V_x (f(x) + g(x) \alpha(x)) = -q(x) \quad (8-25)$$

où  $q(x)$  est une fonction définie positive.

Si nous considérons maintenant le système :

$$\dot{x} = f(x) + b g(x) u \quad (8-26)$$

où  $b$  est une constante inconnue mais de signe connu.

Il serait intéressant de chercher une nouvelle loi de commande en s'inspirant de la commande initiale  $u = \alpha(x)$ . C'est pour cela que nous allons chercher la dynamique de l'estimateur  $\hat{\zeta}$  telle que la nouvelle loi de commande  $u = \hat{\zeta} \alpha(x)$  soit stabilisante.

Ici, la variable  $\hat{\zeta}$  peut être interprétée comme une estimée de  $1/b$ . Nous augmentons alors la fonction CLF initiale par un terme quadratique qui pénalise l'erreur d'estimation  $\tilde{\zeta}$  :

$$V_1 = V(x) + \frac{b}{2\gamma} \tilde{\zeta}^2 = V(x) + \frac{b}{2\gamma} \left(\frac{1}{b} - \hat{\zeta}\right)^2 \quad \text{avec} \quad \gamma > 0 \quad (8-27)$$

La dérivée de la CLF donne :

$$\begin{aligned} \dot{V} &= V_x (f + g \alpha + b g \hat{\zeta} \alpha - g \alpha) - \frac{b}{\gamma} \tilde{\zeta} \dot{\hat{\zeta}} \\ &= -q(x) + V_x g (b \hat{\zeta} - 1) \alpha - \frac{b}{\gamma} \tilde{\zeta} \dot{\hat{\zeta}} \\ &= -q(x) - b \tilde{\zeta} \left( V_x g \alpha + \frac{1}{\gamma} \dot{\hat{\zeta}} \right) \end{aligned} \quad (8-28)$$

Si nous choisissons

$$\dot{\hat{\zeta}} = -\text{sgn}(b) V_x g \alpha \gamma, \quad (8-29)$$

la condition (8-25) est vérifiée et le système bouclé est globalement asymptotiquement stable.

Notons, toutefois, que l'estimateur développé ne présente aucune garantie de convergence asymptotique [Ksr95] ; l'estimateur  $\hat{\zeta}$  peut bien converger vers une valeur bornée différente de la vraie valeur  $1/b$ .

### 8.1.4.5. Observateur par Backstepping

La loi de commande synthétisée par Backstepping dépend des différents états du système étudié. Néanmoins, en général, les états d'un système donné ne sont pas tous accessibles à la mesure. C'est pourquoi, nous avons besoin de reconstruire ces états, à base des entrées et des sorties du système, afin qu'on puisse implémenter la commande Backstepping.

Pour les systèmes linéaires, cette problématique peut être décomposée en deux sous-problèmes qui peuvent être résolus séparément : la synthèse de la loi de commande par retour d'état et la synthèse de l'observateur des états non mesurables. Ce principe de séparation ne s'applique pas pour les systèmes

non linéaires. Dans [Ksr95], les auteurs proposent une approche de synthèse récursive où ils remplacent l'estimateur d'état dans le modèle et considèrent l'erreur d'estimation comme une perturbation. L'effet de la perturbation est ensuite contrebalancé par l'ajout d'un terme d'amortissement non linéaire. L'exemple suivant illustre les différentes étapes à suivre dans cette technique.

Considérons le système suivant :

$$\begin{cases} \dot{x} = -x + x^4 + x^2 \xi \\ \dot{\xi} = -k\xi + u \end{cases} \quad \text{avec } k > 0 \quad (8-30)$$

L'équilibre étudié est  $(x, \xi) = (0, 0)$ .

Quand  $x$  et  $\xi$  sont tout les deux mesurées, ce système peut être stabilisé par une commande Backstepping. En utilisant  $\xi$  comme une commande virtuelle dans la première équation d'état, un choix évident de la commande stabilisante est  $\alpha(x) = -x^2$ . Ce choix réduit la première équation d'état à  $\dot{x} = -x$ .

En introduisant la première variable d'erreur  $\varepsilon = \xi - \alpha(x)$  et en réécrivant le système (8-30), on obtient :

$$\begin{cases} \dot{x} = -x + x^2 \varepsilon \\ \dot{\varepsilon} = -k\xi + u + 2x(-x + x^2 \varepsilon) \end{cases} \quad (8-31)$$

Nous choisissons la fonction CLF  $V(x, \xi) = (x^2 + \varepsilon^2) / 2$  qui a pour dérivée :

$$\dot{V} = -x^2 + \varepsilon(x^3 - k\xi + u + 2x(-x + x^2 \varepsilon)) \quad (8-32)$$

Ainsi, le choix de la commande

$$u = -c\varepsilon - x^3 + k\xi - 2x(-x + x^2 \varepsilon) \quad \text{avec } c > 0 \quad (8-33)$$

garantit la stabilité globale asymptotique de l'équilibre  $(0, 0)$  du système bouclé.

Supposant maintenant que  $\xi$  n'est pas mesurable. La première loi de commande virtuelle ne peut être  $\xi$  et la variable d'erreur  $\varepsilon = \xi + x^2$  n'est pas réalisable en pratique.

En imitant le principe de construction d'observateurs des systèmes linéaires, nous choisissons :

$$\dot{\hat{\xi}} = -k\hat{\xi} + u \quad (8-34)$$

L'erreur d'estimation  $\tilde{\xi} = \xi - \hat{\xi}$  est régit par l'équation :

$$\dot{\tilde{\xi}} = -k\tilde{\xi} \rightarrow \tilde{\xi}(t) = \tilde{\xi}(0)e^{-kt} \quad (8-35)$$

En remplaçant  $\xi$  par  $\tilde{\xi} + \hat{\xi}$  dans la première équation de (8-31), nous trouvons :

$$\dot{x} = -x + x^4 + x^2 \tilde{\xi} + x^2 \hat{\xi} \quad (8-36)$$

L'introduction de la dynamique de l'observateur dans le système donne :

$$\begin{cases} \dot{x} = -x + x^4 + x^2 \hat{\xi} + x^2 \tilde{\xi} \\ \dot{\hat{\xi}} = -k\hat{\xi} + u \\ \dot{\tilde{\xi}} = -k\tilde{\xi} \end{cases} \quad (8-37)$$

L'étape suivante consiste à analyser l'effet de l'erreur d'observation sur la commande (8-33). Pour ce faire nous essayons de synthétiser une loi de commande qui stabilise le système augmenté (8-37). En remplaçant l'erreur  $\varepsilon = \xi - \alpha(x)$  par  $\varepsilon = \hat{\xi} - \alpha(x)$ , le système en boucle fermée s'écrit :

$$\begin{cases} \dot{x} = -x + x^2 \varepsilon + x^2 \tilde{\xi} \\ \dot{\varepsilon} = -c\varepsilon - x^3 + 2x^3 \tilde{\xi} \\ \dot{\tilde{\xi}} = -k\tilde{\xi} \end{cases} \quad (8-38)$$

Une analyse des équations de ce système en boucle fermée montre que pour  $\varepsilon \equiv 0$ , nous avons :  $\dot{x} = -x + x^2 \tilde{\xi}$  et  $\tilde{\xi}(t) = \tilde{\xi}(0)e^{-kt}$  avec comme solution :

$$x(t) = \frac{x(0)(1+k)}{[1+k - x(0)\tilde{\xi}(0)]e^t + x(0)\tilde{\xi}(0)e^{-kt}} \quad (8-39)$$

Cette solution diverge en un temps fini pour toutes conditions initiales vérifiant :  $x(0)\tilde{\xi}(0) > 1+k$ .

Afin de surmonter ce problème [Krs95] propose d'incorporer un terme d'amortissement non linéaire “*nonlinear damping term*”.

Le but de l'introduction du terme d'amortissement non linéaire est d'affecter l'entrée de perturbation dans le système bouclé. L'idée principale est d'avoir un terme dans la commande qui permet de supprimer les carrés qui sont multipliés par la perturbation. Si la perturbation  $\tilde{\xi}$  entre dans l'équation du modèle multipliée par une fonction bornée par une constante ou par une fonction linéaire. La loi de commande initialement choisie pourrait rester satisfaisante. Evidemment, ce n'est pas le cas ici. A cet effet, nous choisissons la première commande stabilisante égale à :

$$\alpha(x) = -x^2 - d_1 x^3 \quad \text{avec} \quad d_1 > 0 \quad (8-40)$$

En démarrant de la première fonction de Lyapunov  $V(x) = x^2/2$  et en ajoutant un terme de pénalisation de l'erreur d'estimation  $\tilde{\xi}$ , nous définissons la nouvelle CLF :

$$V_1(x, \tilde{\xi}) = V(x) + \frac{1}{2d_1 k} \tilde{\xi}^2 = \frac{1}{2} x^2 + \frac{1}{2d_1 k} \tilde{\xi}^2 \quad (8-41)$$

En utilisant la commande stabilisante (8-40), la dérivée de cette CLF est donnée par :

$$\dot{V}_1 = -x^2 + x^3 \varepsilon - d_1 \left( x^3 - \frac{\tilde{\xi}}{2d_1} \right)^2 - \frac{3}{4d_1} \tilde{\xi}^2 \leq -x^2 + x^3 \varepsilon - \frac{3}{4d_1} \tilde{\xi}^2 \quad (8-42)$$

D'où, si  $\varepsilon \equiv 0$  la loi de commande (8-40) rend  $(0,0)$  l'équilibre globalement asymptotiquement stable du système  $(x, \tilde{\xi})$ .

La dérivée de l'erreur  $\varepsilon$  est alors donnée par :

$$\dot{\varepsilon} = -k\hat{\xi} + u - \frac{\partial\alpha}{\partial x}(-x + x^4 + x^2\hat{\xi} + x^2\tilde{\xi}) \quad (8-43)$$

L'erreur d'estimation apparaît de nouveau dans la deuxième équation du système. Afin d'annuler son effet, nous ajoutons un autre terme d'amortissement non linéaire. La fonction de Lyapunov est alors augmentée comme suit :

$$V_2 = V_1 + \frac{1}{2}\varepsilon^2 + \frac{1}{2d_2k}\tilde{\xi}^2 \quad (8-44)$$

Sa dérivée temporelle est donnée par :

$$\begin{aligned} \dot{V}_2 &= -x^2 + x^3\varepsilon - d_1\left(x^3 - \frac{\tilde{\xi}}{2d_1}\right)^2 - \frac{3}{4d_1}\tilde{\xi}^2 + \varepsilon\left(-k\hat{\xi} + u - \frac{\partial\alpha}{\partial x}(-x + x^4 + x^2\hat{\xi})\right) - \varepsilon\frac{\partial\alpha}{\partial x}x^2\tilde{\xi} - \frac{1}{d_2}\tilde{\xi}^2 \\ &= -x^2 - d_1\left(x^3 - \frac{\tilde{\xi}}{2d_1}\right)^2 - \frac{3}{4d_1}\tilde{\xi}^2 - \frac{1}{d_2}\tilde{\xi}^2 + \varepsilon\left(x^3 - k\hat{\xi} + u - \frac{\partial\alpha}{\partial x}(-x + x^4 + x^2\hat{\xi})\right) - \varepsilon\frac{\partial\alpha}{\partial x}x^2\tilde{\xi} \end{aligned} \quad (8-45)$$

Si nous choisissons la commande  $u$  telle que :

$$u = -c\varepsilon - x^3 + k\hat{\xi} + \frac{\partial\alpha}{\partial x}(-x + x^4 + x^2\hat{\xi}) - d_2\varepsilon\left(\frac{\partial\alpha}{\partial x}x^2\right)^2 \quad (8-46)$$

La dérivée  $\dot{V}_2$  devient :

$$\begin{aligned} \dot{V}_2 &= -x^2 - d_1\left(x^3 - \frac{\tilde{\xi}}{2d_1}\right)^2 - \frac{3}{4d_1}\tilde{\xi}^2 - \frac{1}{d_2}\tilde{\xi}^2 - c\varepsilon^2 - \varepsilon\frac{\partial\alpha}{\partial x}x^2\tilde{\xi} - d_2\varepsilon^2\left(\frac{\partial\alpha}{\partial x}x^2\right)^2 \\ &= -x^2 - c\varepsilon^2 - d_1\left(x^3 - \frac{\tilde{\xi}}{2d_1}\right)^2 - \frac{3}{4d_1}\tilde{\xi}^2 - d_2\left(\varepsilon\frac{\partial\alpha}{\partial x}x^2 + \frac{1}{d_2}\tilde{\xi}\right)^2 - \frac{3}{4d_2}\tilde{\xi}^2 \\ &\leq -x^2 - c\varepsilon^2 - \frac{3}{4}\left(\frac{1}{d_1} + \frac{1}{d_2}\right)\tilde{\xi}^2 \end{aligned} \quad (8-47)$$

Finalement, il ressort donc que la condition (8-25) est remplie et l'origine est l'équilibre globalement asymptotiquement stable du système en boucle fermée.



## 8.2. Annexe 2 : Commande d'une suspension magnétique

### 8.2.1. Analyse de la stabilité du système commandé par Backstepping et observateur

Dans ce qui suit, nous démontrons la stabilité asymptotique globale du système bouclé par la loi de commande backstepping et l'observateur d'ordre réduit. L'approche proposée consiste à prouver l'existence d'une loi de commande par backstepping (et donc d'une fonction de Lyapunov assignable) assurant la stabilité asymptotique globale de la structure système-observateur-commande Backstepping.

#### Étape 1 :

En réécrivant le système d'état (6-81) en utilisant la variable erreur de poursuite  $e_1 = x_1 - r$ , on trouve :

$$\begin{cases} \dot{e}_1 = c_4(e_1 + r) + \hat{v} + \varepsilon \\ \dot{\hat{v}} = -g + \theta\lambda(e_1)u^2 - c_4(\hat{v} + c_4(e_1 + r)) \\ \dot{\varepsilon} = -c_4\varepsilon \end{cases} \quad (8-48)$$

Afin qu'on puisse appliquer l'approche par Backstepping, l'erreur d'observation  $\varepsilon$  est considérée, ici, comme une perturbation asymptotiquement nulle.

En choisissant la fonction de Lyapunov assignable  $V(e_1) = e_1^2 / 2$ , sa dérivée temporelle donne :

$$\dot{V} = e_1 \dot{e}_1 = e_1(c_4(e_1 + r) + \hat{v} + \varepsilon) \quad (8-49)$$

En sélectionnant  $\hat{v}$  comme commande virtuelle indépendante de l'erreur d'observation  $\varepsilon$  telle que,

$$\hat{v}^{des} = \alpha(e_1) = -c_1 e_1 - c_4(e_1 + r) \quad \text{avec} \quad c_1 > 0 \quad (8-50)$$

La dérivée temporelle de la fonction de Lyapunov devient :

$$\dot{V} = -c_1 e_1^2 + e_1 \varepsilon \quad (8-51)$$

Même si cette dérivée  $\dot{V}$  dépend de la perturbation  $\varepsilon$ , le choix de la commande virtuelle (8-50) peut être conservé si  $\varepsilon$  intervient dans la première équation du modèle multipliée par une fonction bornée par une fonction linéaire [Krs95]. Ceci est bien vérifié dans le cas du modèle (8-51), où la perturbation est multipliée par une constante égale à 1. L'exemple de la section 8.1.4.5 illustre un cas contraire où une nouvelle loi de commande doit être retrouvée.

Néanmoins, la condition de non positivité de  $\dot{V}$  reste non vérifiée ; le signe du terme  $e_1 \varepsilon$  est inconnu. Par conséquent, on propose de modifier la fonction de Lyapunov  $V$  en l'augmentant par un terme quadratique de l'erreur d'observation.

$$V_1 = V + \frac{1}{2c_4 d_1} \varepsilon^2 \quad (8-52)$$

La dérivée temporelle est donnée par :

$$\begin{aligned}
 \dot{V}_1 &= -c_1 e_1^2 + e_1 \varepsilon - \frac{1}{d_1} \varepsilon^2 \\
 &= -c_1 \left( e_1^2 - \frac{1}{c_1} e_1 \varepsilon - \frac{1}{c_1 d_1} \varepsilon \right)^2 \\
 &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1 d_1} \right) \varepsilon^2
 \end{aligned} \tag{8-53}$$

$\dot{V}_1$  est semi-définie négative si et seulement si  $d_1 < 4c_1$ .

Le choix de la première loi de commande virtuelle transforme la première équation d'état en :

$$\dot{e}_1 = -c_1 e_1 + \varepsilon \tag{8-54}$$

Ce qui correspond à la solution

$$e_1(t) = \begin{cases} e^{-c_1 t} (e_1(0) + t\varepsilon(0)) & \text{si } c_4 = c_1 \\ e_1(0)e^{-c_1 t} + \frac{\varepsilon(0)}{c_4 - c_1} (e^{-c_1 t} - e^{-c_4 t}) & \text{sinon} \end{cases} \tag{8-55}$$

### Étape 2 :

Nous définissons la nouvelle variable d'erreur comme étant l'écart entre la première commande virtuelle  $\hat{v}$  et sa valeur désirée  $\hat{v}^{des} = \alpha(e_1)$ ,

$$e_2 = \hat{v} - \alpha \tag{8-56}$$

La réécriture du modèle dans la nouvelle base  $(e_1, e_2, \varepsilon)$  donne :

$$\begin{cases} \dot{e}_1 = c_4(e_1 + r) + (e_2 + \alpha) + \varepsilon \\ \dot{e}_2 = -g + \theta\lambda(e_1)u^2 + c_1((e_2 + \alpha) + c_4(e_1 + r)) + (c_4 + c_1)\varepsilon \\ \dot{\varepsilon} = -c_4\varepsilon \end{cases} \tag{8-57}$$

La nouvelle fonction de Lyapunov assignable est alors donnée par :

$$V_2 = V_1 + \frac{1}{2}c_3 e_2^2 \quad \text{avec } c_3 > 0 \tag{8-58}$$

et sa dérivée  $\dot{V}_2$  est

$$\begin{aligned}
 \dot{V}_2 &= \dot{V}_1 + c_3 e_2 \dot{e}_2 \\
 &= e_1(c_4(e_1 + r) + (e_2 + \alpha) + \varepsilon) - \frac{1}{d_1} \varepsilon^2 + c_3 e_2 (-g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon) \\
 &= e_1(c_4(e_1 + r) + \alpha + \varepsilon) - \frac{1}{d_1} \varepsilon^2 + c_3 e_2 (e_1/c_3 - g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon)
 \end{aligned} \tag{8-59}$$

Le choix du terme  $u^2$  définissant la commande  $u$  se fait en imposant

$$e_1/c_3 - g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) = -c_2/c_3e_2 \quad \text{avec} \quad c_2 > 0$$

c'est-à-dire :

$$\begin{aligned} u^2 &= \frac{1}{\theta\lambda(e_1)}(-c_2/c_3e_2 + g - e_1/c_3 - c_1(c_4(e_1 + r) + (e_2 + \alpha))) \\ &= \frac{1}{\theta\lambda(e_1)}(-c_2/c_3e_2 + g - e_1/c_3 - c_1(e_2 - c_1e_1)) \end{aligned} \quad (8-60)$$

La dérivée de la fonction de Lyapunov  $V_2$  devient alors :

$$\dot{V}_2 = -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1d_1} \right) \varepsilon^2 - c_2e_2^2 + c_3(c_4 + c_1)e_2\varepsilon \quad (8-61)$$

De nouveau, nous voyons qu'il n'y a aucun besoin d'ajouter un autre terme sous la forme d'atténuation non-linéaire. Mais nous avons besoin d'un nouveau terme dans l'expression de la fonction de Lyapunov afin d'être sûr que le terme de signe indéfini, dans la dernière expression, n'occasionne pas l'instabilité du système bouclé. Par conséquent

$$V_3 = V_2 + \frac{1}{2d_2k} \varepsilon^2 = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 + \frac{1}{2c_4} \left( \frac{1}{d_1} + \frac{1}{d_2} \right) \varepsilon^2 \quad (8-62)$$

et

$$\begin{aligned} \dot{V}_3 &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1d_1} \right) \varepsilon^2 - c_2e_2^2 + c_3(c_4 + c_1)e_2\varepsilon - \frac{1}{d_2} \varepsilon^2 \\ &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1d_1} \right) \varepsilon^2 - c_2 \left( e_2^2 - \frac{c_3(c_4 + c_1)}{c_2} e_2\varepsilon + \frac{1}{c_2d_2} \varepsilon^2 \right) \\ &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1d_1} \right) \varepsilon^2 - c_2 \left( e_2 - \frac{c_3(c_4 + c_1)}{2c_2} \varepsilon \right)^2 - \left( \frac{4c_2 - d_2c_3^2(c_4 + c_1)^2}{4c_2d_2} \right) \varepsilon^2 \\ &= -q(e_1, e_2, \varepsilon) \end{aligned} \quad (8-63)$$

$\dot{V}_3$  est semi-définie négative si et seulement si  $4c_2 - d_2c_3^2(c_4 + c_1)^2 > 0$ .

In fine, le système est globalement asymptotiquement stable. ■

## 8.2.2. Analyse de la stabilité du système commandé par Backstepping à action intégrale et observateur

Nous analysons dans ce qui suit la loi de commande Backstepping à action intégrale du système augmenté (système+observateur).

En suivant la même procédure que la section précédente, nous prouvons l'existence d'une fonction de Lyapunov assignable assurant la stabilité asymptotique globale de la structure système-observateur-Backstepping à action intégral.

Étape 1 :

Le système est donné par :

$$\begin{cases} \dot{e}_1 = c_4(e_1 + r) + \hat{v} + \varepsilon \\ \dot{\hat{v}} = -g + \theta\lambda(e_1)u^2 - c_4(\hat{v} + c_4(e_1 + r)) \\ \dot{\varepsilon} = -c_4\varepsilon \end{cases} \quad (8-64)$$

Nous choisissons la fonction de Lyapunov assignable suivante :

$$V(e_1) = \frac{1}{2}(e_1^2 + \kappa q^2) \quad \text{avec} \quad \kappa > 0 \quad (8-65)$$

$$\text{avec } q = \int_0^t e_1(\tau) d\tau$$

sa dérivée temporelle est donnée par :

$$\dot{V} = e_1\dot{e}_1 + e_1\dot{q} = e_1(c_4(e_1 + r) + \hat{v} + \varepsilon + \kappa q) \quad (8-66)$$

En sélectionnant  $\hat{v}$  comme commande virtuelle indépendante de l'erreur d'observation  $\varepsilon$  telle que,

$$\hat{v}^{des} = \alpha(q, e_1) = -\kappa q - c_1 e_1 - c_4(e_1 + r) \quad \text{avec} \quad c_1 > 0 \quad (8-67)$$

La dérivée temporelle de la fonction de Lyapunov devient :

$$\dot{V} = -c_1 e_1^2 + e_1 \varepsilon \quad (8-68)$$

$$V_1 = V + \frac{1}{2c_4 d_1} \varepsilon^2 \quad (8-69)$$

La dérivée temporelle est donnée par :

$$\begin{aligned} \dot{V}_1 &= -c_1 e_1^2 + e_1 \varepsilon - \frac{1}{d_1} \varepsilon^2 \\ &= -c_1 \left( e_1^2 - \frac{1}{c_1} e_1 \varepsilon - \frac{1}{c_1 d_1} \varepsilon^2 \right) \\ &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1 d_1} \right) \varepsilon^2 \end{aligned} \quad (8-70)$$

$\dot{V}_1$  est semi-définie négative si et seulement si  $d_1 < 4c_1$ .

Le choix de la première loi de commande virtuelle transforme la première équation d'état en :

$$\dot{e}_1 = -c_1 e_1 - \kappa q + \varepsilon \quad (8-71)$$

Étape 2 :

Nous définissons la nouvelle variable d'erreur comme étant l'écart entre la première commande virtuelle  $\hat{v}$  et sa valeur désirée  $\hat{v}^{des} = \alpha(e_1)$ ,

$$e_2 = \hat{v} - \alpha \quad (8-72)$$

La réécriture du modèle dans la nouvelle base  $(e_1, e_2, \varepsilon)$  donne :

$$\begin{cases} \dot{q} = e_1 \\ \dot{e}_1 = c_4(e_1 + r) + (e_2 + \alpha) + \varepsilon \\ \dot{e}_2 = -g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon + \kappa e_1 \\ \dot{\varepsilon} = -c_4\varepsilon \end{cases} \quad (8-73)$$

La nouvelle fonction de Lyapunov assignable est alors donnée par :

$$V_2 = V_1 + \frac{1}{2}c_3e_2^2 \quad (8-74)$$

et sa dérivée  $\dot{V}_2$  est

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + c_3e_2\dot{e}_2 \\ &= e_1(c_4(e_1 + r) + (e_2 + \alpha) + \varepsilon) - \frac{1}{d_1}\varepsilon^2 + c_3e_2(-g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon + \kappa e_1) \\ &= e_1(c_4(e_1 + r) + \alpha + \varepsilon) - \frac{1}{d_1}\varepsilon^2 + c_3e_2(e_1/c_3 - g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + (c_4 + c_1)\varepsilon + \kappa e_1) \end{aligned} \quad (8-75)$$

Le choix du terme  $u^2$  définissant la commande  $u$  se fait en imposant

$$e_1/c_3 - g + \theta\lambda(e_1)u^2 + c_1(c_4(e_1 + r) + (e_2 + \alpha)) + \kappa e_1 = -c_2/c_3e_2$$

C'est-à-dire :

$$\begin{aligned} u^2 &= \frac{1}{\theta\lambda(e_1)}(-c_2/c_3e_2 + g - e_1/c_3 - c_1(c_4(e_1 + r) + (e_2 + \alpha)) - \kappa e_1) \\ &= \frac{1}{\theta\lambda(e_1)}(-c_2/c_3e_2 + g - e_1/c_3 - c_1(e_2 - c_1e_1 - \kappa q) - \kappa e_1) \end{aligned} \quad (8-76)$$

La dérivée de la fonction de Lyapunov  $V_2$  devient alors :

$$\dot{V}_2 = -c_1\left(e_1 - \frac{1}{2c_1}\varepsilon\right)^2 - \left(\frac{4c_1 - d_1}{4c_1d_1}\right)\varepsilon^2 - c_2e_2^2 + c_3(c_4 + c_1)e_2\varepsilon \quad (8-77)$$

De nouveau, nous voyons qu'il n'y a aucun besoin d'ajouter un autre terme sous la forme d'atténuation non-linéaire. Mais nous avons besoin d'un nouveau terme dans l'expression de la fonction de Lyapunov afin d'être sûr que le terme de signe indéfini, dans la dernière expression, n'occasionne pas l'instabilité du système bouclé. Par conséquent

$$V_3 = V_2 + \frac{1}{2d_2k}\varepsilon^2 = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2c_4}\left(\frac{1}{d_1} + \frac{1}{d_2}\right)\varepsilon^2 \quad (8-78)$$

et

$$\begin{aligned}
\dot{V}_3 &= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1 d_1} \right) \varepsilon^2 - c_2 e_2^2 + c_3 (c_4 + c_1) e_2 \varepsilon - \frac{1}{d_2} \varepsilon^2 \\
&= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1 d_1} \right) \varepsilon^2 - c_2 \left( e_2^2 - \frac{c_3 (c_4 + c_1)}{c_2} e_2 \varepsilon + \frac{1}{c_2 d_2} \varepsilon^2 \right) \\
&= -c_1 \left( e_1 - \frac{1}{2c_1} \varepsilon \right)^2 - \left( \frac{4c_1 - d_1}{4c_1 d_1} \right) \varepsilon^2 - c_2 \left( e_2 - \frac{c_3 (c_4 + c_1)}{2c_2} \varepsilon \right)^2 - \left( \frac{4c_2 - d_2 c_3^2 (c_4 + c_1)^2}{4c_2 d_2} \right) \varepsilon^2 \\
&= -q(e_1, e_2, \varepsilon)
\end{aligned} \tag{8-79}$$

$\dot{V}_1$  est semi-définie négative si et seulement si  $4c_2 - d_2 c_3^2 (c_4 + c_1)^2 > 0$ . Il est donc toujours possible de trouver  $d_2$  pour que cette inégalité soit vérifiée. Le système en boucle fermée est donc globalement asymptotiquement stable et il est équivalent au système linéaire suivant :

$$\begin{cases} \dot{q} = e_1 \\ \dot{e}_1 = -\kappa q - c_1 e_1 + e_2 + \varepsilon \\ \dot{e}_2 = -e_1 / c_3 - c_2 / c_3 e_2 + (c_4 + c_1) \varepsilon \\ \dot{\varepsilon} = -c_4 \varepsilon \end{cases} \tag{8-80} \blacksquare$$

## 8.3. Annexe 3 : Commande d'un système de forage pétrolier

### PARAMETRIC ADJUSTEMENT OF A BACKSTEPPING CONTROLLER BY NONSMOOTH OPTIMIZATION: APPLICATION TO A ROTARY DRILLING SYSTEM

B. Lassami, F. Abdulgalil, S. Font and H. Siguerdidjane

*Automatic Control Department  
Ecole Supérieure d'Electricité (Supélec)  
Gif-sur-Yvette, F-91192-cedex France.*

**Abstract:** This paper presents a general parametric optimization approach in control's field. It discusses the design of optimal controllers and the validation of the temporal specifications as well. The formulated problems are complex because they involve nonsmooth functions and criteria. Therefore, we propose an efficient descent algorithm based on the so called  $\varepsilon$  subdifferential notion. This last, mixing with an exact computation of gradients based on parametric sensitivity functions, appears to be well suited to problems with nonsmooth criteria. As illustration, this method is used to tune the parameters of a backstepping controller for a rotary drilling system. The purpose of this application especially concerns how to appropriately apply the  $\varepsilon$  subdifferential algorithm in order to improve the backstepping technique. We show that the choice of the controller structure and parameters has an important effect on the validation of the specifications. Simulation results are given to demonstrate the effectiveness of the proposed approach. *Copyright © 2006 IFAC*

**Keywords:** Optimization problem, adjustment, sensitivity functions, nonsmooth criteria, backstepping, rotary drilling system.

#### 1. INTRODUCTION

Modern control theory provides a collection of tools for the design of feedback controllers ensuring satisfactory closed loop performance of plants. Considerable attention has been devoted to the problem of designing optimal controllers taking into account complex specifications and maximizing the parametric stability margin. Moreover, general trend has been to check for high performance (rapidity, accuracy, rejection or attenuation of perturbation signals...), while ensuring moderate control signals and good robustness properties. This need of high system performance, together with the evolution of computation techniques and information processing, accentuates the necessity of optimization in automatic. Many studies related to the use of optimization for complex engineering requirements with low complexity algorithms have been developed. These theories mainly focus on convex optimization methods that concern synthesis techniques of controllers (Doyle, *et al.*, 1989). However, it is well known that the optimization of parametric controllers with fixed structure is in general a non convex optimization problem. We will focus on methods, which can be developed out of the convex context. In this case, there is no guarantee to obtain the absolute optimum and we resort to the global optimization methods which allow to retune controller parameters. Far from to being a direct operation, the control

design has to be made in several stages in order to obtain a satisfactory result. Once the design conditions are modified, the most effective way is not necessarily to start again the work from the beginning. Thus, the need for a retuning may arise all along the development of the system each time the design model or specifications evolve.

In this paper, the last situation is considered. Generic criteria properties will be shown as well as the way to consider the optimization problem in order to get an efficient resolution. For temporal specifications, the proposed approach allows an exact formulation of requirements to the detriment of an exact analysis of their feasibility. The flexibility of these generic optimization approaches is illustrated through a backstepping controller of a rotary drilling system. In this process, there are different types of vibrations, one of them and for which we are interested in this work is torsional oscillations by means of Stick-Slip phenomena. These ones are induced by nonlinear frictional torques between the drill bit and the rock surface. The focus of this work is to make an extension of the backstepping that has been synthesised in a previous work (Abdulgalil and Siguerdidjane, 2005), the aim is to show the potential level of our technique through an application for which retuning controller parameters by optimization is yet very difficult using a classical descent algorithm.

Motivated by the above comments, the first part of this study describes the formulation of generic specifications in control design problems under a global optimization problem form. The resolution of such a problem is difficult and requires specific algorithms. Subsequently, a descent algorithm based on the  $\varepsilon$  subdifferential notion is exposed, which is able to deal with nonsmooth criteria. The second and last part is dedicated to present the application and the obtained simulation results.

## 2. FORMULATION OF THE PROBLEM

Mathematical formulation of the different temporal requirements will permit to express the controller retuning problem as a global parametric optimization one. Let us consider the block diagram given by Fig.1. The tuned controller parameters are designed by the vector  $\theta$  which constitutes the decision variables. For particular input signals ( $r$ ,  $b$  and  $w$ ), any output signals will be of the form:

$$s(K(\theta), G, r(t), b(t), w(t), t) = s(\theta, t) \quad (1)$$

The classic criteria of the specifications (time response, maximum overshoot...) are also dependent on parameters vector  $\theta$ . These will be denoted by the expression:

$$\alpha_i(s(\theta, t), t) = \alpha_i(\theta) \quad (2)$$

Generally, the control problem can be translated using constraints on indicator  $\alpha_i$  or on Templates (Boyd and Barrat, 1991). In this case, these requirements are formulated using the following inequalities:

$$\begin{cases} \forall t > 0, s_{\min}(t) \leq s(\theta, t) \leq s_{\max}(t) \\ F_i(\alpha_i(\theta)) \leq 0 \end{cases} \quad (3)$$

It can be noticed that these various constraints can be equivalently formulated as criteria functions.

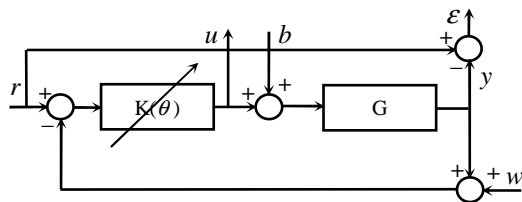


Fig. 1. Feedback system with tuned variables.

For instance, requirements that can be formulated using (3) can be equivalently formulated using the following feasibility criterion:

$$J_i = \max \left( \max_{t>0} (s(\theta, t) - s_{\max}(t)), \max_{t>0} (s_{\min}(t) - s(\theta, t)) \right) \quad (4)$$

The constraint is satisfied if and only if the associated optimal criterion is negative.

Hence, the global optimization problem can be stated by mixing one or more criteria as equation (4) and some constraints like expressions in (3).

## 3. OPTIMIZATION PROBLEM RESOLUTION

The proposed algorithm is based on the descent method. It is a very efficient method when a descent direction can be correctly determined. The most known method to this class of algorithms is the 'gradient method'. It is based on computation of the local gradient which presents the best local choice when it is defined. The descent direction is then determined at each iteration  $k$  as:

$$d_k = -\hat{g}_k \quad \text{and} \quad \hat{g}_k = \nabla \hat{J}(\theta_k) = \left. \frac{\partial \hat{J}}{\partial \theta} \right|_{\theta=\theta_k}$$

This algorithm presents many advantages. Contrary to the other methods with more elevated order (Hessian), it requires less expensive calculations and it is robust with regards to estimation errors of the gradient. However, the convergence, in a reasonable time, depends on the gradient calculations; it must be reliable (with a good precision) and easily tractable (low complexity of calculations).

### 3.1 Criteria and Constraints Computations

For generic specifications, calculations of criteria and constraints are often not so easy. For example; the time response calculations require the use of a unidirectional optimization. Furthermore, inequality constraints, as (3), must be calculated algebraically, by optimization or by gridding. In the general case, all temporal criteria first need an integration of differential equations. Then, only an estimation of trajectories can be obtained that contains a finite number of estimated points. So the numerical estimation of parametric sensitivity (or gradient) of such implicit variables is really difficult using finite difference approach. Thus, the criterion  $J$  can not be exactly computed and an estimated value  $\hat{J} = J + b$  is available only ( $b$  is a numerical noise).

### 3.2 Gradient Calculation by Parametric Sensitivities

Two kinds of methods can be distinguished for the gradient computation. The first one is an approximate method using finite difference techniques. This method is useful but nevertheless leads to unreliable result when the numerical error is strong. For illustration, an analysis of the variance gives:

$$\sigma^2 \left[ \frac{\Delta \hat{J}}{\Delta \theta} \right] = \sigma^2 \left[ \frac{1}{h} [J(\theta + h \cdot v) + b_1 - J(\theta) - b_2] \right] \approx \frac{2\sigma^2(b)}{h^2} \quad (5)$$

where  $v$  denotes a unit vector and  $b_i \rightarrow N(0, \sigma^2(b))$ .

The second method is an exact calculation. It is a precise method but it requires more calculus prior optimization. In order to ameliorate the quality of the gradient estimation, this last method will be considered. In this case, gradient can be accurately computed. Sensitivity function approach is chosen because it does not require more calculations ( $\dim(\theta)+1$  computations of the criterion). This method can be stated for nonlinear systems (NL) as following: Let us consider the NL system:



$$\begin{cases} \frac{dx(t)}{dt} = f[x(t), a] \\ y(t, a) = h[x(t), a] \end{cases} \quad (6)$$

$f$  and  $h$  may also depend on the command  $u(t)$  and the time  $t$ . The vector  $a$  denotes any parameter of the system, its relationship with  $\theta$  will be explained in the sequel. The parametric sensitivities of the output signal  $y$  with respect to arguments  $a$  are defined by the following expression for  $i \in \{1, \dots, \dim(a)\}$  (Rosenwasser, 2000):

$$S_{y/a_i}(t, a) = \frac{\partial}{\partial a_i} y(t, a) \quad (7)$$

This definition applied to the NL model gives:

$$S_{y/a_i}(t, a) = \frac{\partial h[x(t), a]}{\partial x^T} S_{x/a_i}(t, a) + \frac{\partial h[x(t), a]}{\partial a_i} \quad (8)$$

The derivation with respect to parameters  $a$  gives:

$$\frac{d[S_{x/a_i}(t, a)]}{dt} = \frac{\partial f[x(t), a]}{\partial x^T} S_{x/a_i}(t, a) + \frac{\partial f[x(t), a]}{\partial a_i} \quad (9)$$

The computation of the parametric sensitivities of the output signal  $y$  with respect to coefficients  $a$  requires  $\dim(a) + 1$  simulations (the resolution of the system of equations defined by (6) and (9)). This indicates that the direct method is not more complicated anymore than the one by finite differences.

As coefficients  $a$  depend on parameters  $\theta$ , the parametric sensitivities of the output signal  $y$  with respect to parameters  $\theta$  are given by:

$$S_{y/\theta_k} = \sum_{i=1}^{\dim(a)} \frac{\partial a_i}{\partial \theta_k} S_{y/a_i} \quad (10)$$

As an illustrative example, the following feasibility criterion can be considered:

$$J(\theta) = \sum_{i=1}^{t_f} w_i |y(t_i, \theta) - Y_{\max}(t_i)|_+ \quad (11)$$

with  $|f|_+ = \max(f, 0)$ .

The gradient of this criterion is given almost everywhere for  $k \in \{1, \dots, n_\theta\}$  by the following expression:

$$\frac{\partial J}{\partial \theta_k} = \sum_{i=1}^{t_f} \frac{w_i}{2} \cdot S_{y/\theta_k} \cdot \left( 1 + \frac{|y(t_i, \theta) - Y_{\max}(t_i)|}{y(t_i, \theta) - Y_{\max}(t_i)} \right) \quad (12)$$

### 3.3 Nonsmooth Criteria

The last example illustrates a nonsmooth criterion. Its utilization conjointly with nonsmooth constraints leads to a particular class of optimization problems that require special algorithms. An attentive observation of the several criteria formulated on the retuning controller problems shows that the classic specifications are often formulated by nonsmooth criteria and constraints. For example, criteria which

contain any ‘max’ function are nonsmooth when the maximum is reached simultaneously in several points. However, they present the particularity to be nearly differentiable; the space of nonsmooth points has measure zero (Clarke, 1990; Burke, *et al.*, 2004). Nevertheless, the subspace of nonsmooth points is often reached during the optimization process because these points are often local minima in almost all directions. At these locations, determination of descent direction is not possible with the gradient. The classical subgradient is not a good choice too for a descent method because it is typically very hard to get and its direction does not necessarily represent an ascent direction (Clarke, 1990).

A generalized approach using specific subgradient and subdifferential can be used (Clarke, 1990). In order to construct a rigorous procedure, the Clarke subgradient is chosen. It presents an interesting approximation possibility (Burke, *et al.*, 2002).

Formally, it is assumed that  $f$  is locally Lipschitz continuous and continuously differentiable on an open dense subset  $D$  of  $\mathfrak{R}^n$  and there is a point  $\tilde{x} \in \mathfrak{R}^n$  for which the set  $L = \{x / f(x) \leq f(\tilde{x})\}$  is compact. The local Lipschitz hypothesis allows to approximate the Clarke subdifferential as follows:

1) *Clarke subdifferential approximation*: for each  $\varepsilon > 0$ , we define the multifunction  $G_\varepsilon : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  by:

$$G_\varepsilon(x) = Co(\nabla f(x + \varepsilon B) \cap D) \quad (13)$$

where  $B = \{x / \|x\| \leq 1\}$  is the closed unit ball,  $Co(\cdot)$  is the closed convex hull and  $\nabla f(x)$  the gradient at  $x$ . The sets  $G_\varepsilon(x)$  can be used to give the following representation of the Clarke subdifferential of the function  $f$  at a point  $x$ :

$$\bar{\partial} f(x) = \bigcap_{\varepsilon > 0} G_\varepsilon(x) \quad (14)$$

For each  $\varepsilon > 0$ , the Clarke  $\varepsilon$  subdifferential is defined by:

$$\bar{\partial}_\varepsilon f(x) = Co(\bar{\partial} f(x + \varepsilon B)) \quad (15)$$

For:  $0 < \varepsilon_1 < \varepsilon_2$  the following embedded inclusions are verified:

$$\bar{\partial}_{\varepsilon_1} f(x) \subset G_{\varepsilon_1}(x) \subset \bar{\partial}_{\varepsilon_2} f(x) \quad (16)$$

So, the Clarke  $\varepsilon$  subdifferential can be approximated by  $G_\varepsilon(x)$ . Moreover, due to the hypothesis of almost everywhere differentiability of  $f$ ,  $G_\varepsilon(x)$  can be estimated by a finite uniform spatial sampling in  $B$ :

$$G_\varepsilon(x) \approx Co\left(\bigcup_{i=1}^m (\nabla f(x + \varepsilon b_i) \cap D)\right) \quad \text{with } b_i \in B \quad (17)$$

Figure 2 illustrates an example of a Clarke subdifferential approximation. Finally, the introduced notion of the Clarke  $\varepsilon$  subdifferential  $\bar{\partial}_\varepsilon f(x)$  can be approximated by a convex hull  $G_\varepsilon(x)$  which is a result of uniform finite sampling gradients around a point  $x$ .

2) *Clarke  $\varepsilon$  stationary point*: a point  $x$  is Clarke stationary if  $0 \in \bar{\partial}_\varepsilon f(x)$ . In order to measure the proximity to Clarke  $\varepsilon$  stationarity (Burke, *et al.*, 2004) introduce the following distance:

$$\rho_\varepsilon(x) = \text{dist}(0 / G_\varepsilon(x)). \quad (18)$$

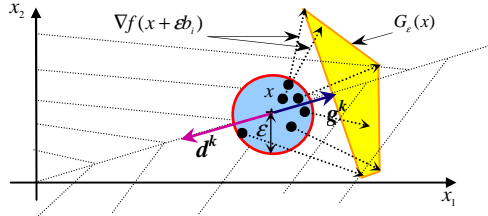


Fig. 2. Approximation of the Clarke  $\varepsilon$  subdifferential.

### 3.4 The $\varepsilon$ Subdifferential Algorithm

The proposed algorithm may be applied to any function  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$  that is continuous and differentiable almost everywhere on  $\mathfrak{R}^n$ . The constraint problems can be handled by an exact penalty where the nonsmooth weightings can also be used. This algorithm is different from the one developed in (Burke, *et al.*, 2004). Basis differences are: addition of tests for managing numerical degenerated cases and introduction of an isotropic point sampling of the hyperball for a better estimation of the  $\varepsilon$  subdifferential.

#### Notations:

- $\varepsilon_k$  Sampling radius at the  $k^{\text{th}}$  iteration.
- $v_k$  Optimality tolerance at the  $k^{\text{th}}$  iteration.
- $\beta$  Armijo parameter.
- $\gamma$  Backtracking reduction factor.
- $\delta$  Optimality tolerance reduction factor.
- $\mu$  Sampling radius reduction factor.

#### The Algorithm:

##### Step 0: (Initialization)

Let  $x^0 \in L \cap D$ ,  $\gamma \in ]0,1[$ ,  $\beta \in ]0,1[$ ,  $\varepsilon_0 > 0$ ,  $v_0 \geq 0$ ,  $\mu \in ]0,1[$ ,  $\delta \in ]0,1[$ ,  $k = 0$  and  $m \in \{n+1, n+2, n+3, \dots\}$ .

##### Step 1: (Approximation of the Clarke $\varepsilon$ subdifferential)

Let  $u^{k1}, u^{k2}, \dots, u^{km}$  be sampled independently and uniformly from  $B$ , and set:  $x^{k0} = x^k$  and

$$x^{kj} = x^k + \varepsilon_k u^{kj} \text{ for } j = 1, \dots, m.$$

If one of the samples points ( $j = 1, \dots, m$ ) verifies  $x^{kj} \notin D$  then, go to Step 1.

Otherwise  $G_k = \text{Co}\{\nabla f(x^{k0}), \nabla f(x^{k1}), \dots, \nabla f(x^{km})\}$ .  $\nabla f(x^{kj})$  is calculated by sensitivity method.

##### Step 2: (Compute a descent direction $d^k$ )

Let  $g^k \in G_k$  be a solution of the positive quadratic problem  $g^k = \arg \min_{g \in G_k} \text{dist}(0 / G_k)$

If  $v_k = \|g^k\| = 0$ , Stop ( $\varepsilon$  stationarity).

Else

→ If  $\|g^k\| \leq v_k$  then, set

$t_k = 0$ ,  $v_{k+1} = \delta v_k$ ,  $\varepsilon_{k+1} = \mu \varepsilon_k$  and go to Step 4.

→ Else  $v_{k+1} = v_k$ ,  $\varepsilon_{k+1} = \varepsilon_k$ ,  $d^k = -g^k / \|g^k\|$

**Step 3:** (Compute a step length  $t_k$ )

$$t_k = \max_{s \in \{0,1,2,\dots\}} \gamma^s / f(x^k + \gamma^s d^k) < f(x^k) - \beta \gamma^s \|g^k\|$$

##### Step 4: (Update)

If  $x^k + t_k d^k \in D$  then,  $x^{k0} = x^k + t_k d^k$ ,  $k = k+1$  and go to Step 1,

Else, let  $\hat{x}^k \in \hat{x}^k + \varepsilon_k B$  satisfying  $\hat{x}^k + t_k d^k \in D$  and  $f(\hat{x}^k + \gamma^s d^k) < f(x^k) - \beta \gamma^s \|g^k\|$  and then,  $x^{k0} = \hat{x}^k + t_k d^k$ ,  $k = k+1$ . Go to step 1.

## 4. APPLICATION TO A DRILLING SYSTEM

For this application, the backstepping control has already been described in (Abdulgalil and Siguerdidjane, 2005). The main contribution of this paper is pointed out and concerns especially how to appropriately adjust the backstepping parameters for satisfying specifications and particularly in order to handle the stick-slip oscillations in oil rotary drilling system.

### 4.1 Rotary Drilling Modeling

A drilling assembly essentially consists of a series of hollow cylindrical steel pipes connected to form a long flexible drillstring to which is attached to a short heavier segment containing a cutting device at the free end called the drill bit as shown in Fig. 3. This segment may contain a stabilizing fins designed to reduce lateral vibrations during the drilling and together with the drill-bit constitute the bottom-hole assembly (BHA). The BHA consists of thick-walled tubulars loaded in compression, and, without buckling; it provides weight on the bit (WOB) required for creating sufficient cutting force.

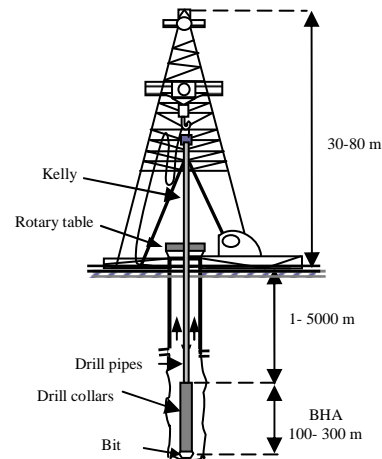


Fig. 3. Drilling equipment

The drillstrings are mainly consisting of a number of relatively thin walled pipes, and they are driven by a rotary table in the top end, often by means of an

electric motor and gearbox, the top-drive. During the process drilling, drilling fluid (mud) is continuously circulated to the bottom of the hole and back to surface to remove cuttings from the bottom of the hole, to cool and lubricate the bit, and to control downhaul pressures.

The drilling system may be modeled as follows: the main components of the model are two damped inertias mechanically coupled by an elastic intertialess shaft (drillstring); as displayed in Fig. 4.

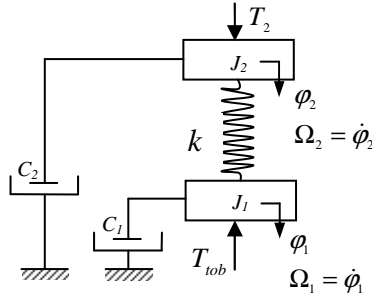


Fig. 4. Drilling rotary model

The detailed model can be found in (Serrarens, *et al.*, 1998). Then, the state equations of the rotary drilling system are:

$$\begin{cases} \dot{\Omega}_1 = -\frac{1}{J_1}(C_1\Omega_1 - k\Phi + T_{tob}(\Omega_1)) \\ \dot{\Phi} = \Omega_2 - \Omega_1 \\ \dot{\Omega}_2 = -\frac{1}{J_2}(C_2\Omega_2 - C_2\Omega_{ref} + k\Phi - u) \end{cases} \quad (19)$$

where  $J_1$  and  $J_2$  represent, respectively, the inertia of the upper part (rotary table and the electric motor) and the lower part (the collars and the drillpipes),  $C_1$  and  $C_2$  denote the equivalent viscous damping coefficients,  $k$  is the stiffness coefficient of the linear torsional spring. The state variables are:  $\Omega_1$  the angular velocity of the bit,  $\Omega_2$  the angular velocity of the rotary table and  $\Phi$  the difference angular displacement between the upper and the lower part of the system.  $T_{tob}$  is the friction torque, given by the following nonlinear function:

$$T_{tob}(\Omega_1) = T_{tobdyn} \frac{2}{\pi} (\alpha_1 \Omega_1 e^{-\alpha_2 |\Omega_1|} + \arctan(\alpha_3 \Omega_1)) \quad (20)$$

with  $T_{tobdyn} = 0.5 \text{ kNm}$ ,  $\alpha_1 = 9.5$ ,  $\alpha_2 = 2.2$  and  $\alpha_3 = 35.0$ .

#### 4.2 Backstepping Control Approach

The backstepping design methodology has become increasingly popular in nonlinear control systems. Complete methodology can be found in (Kanellakopoulos and Kokotovic, 1995). Roughly speaking the main idea of the strategy is characterized by a step-by-step procedure interlacing, at each step, a coordinate transformation and the design of a virtual control via a classical Lyapunov function technique, with the definition of a tuning function, and at the last step obtaining the true control expression. This procedure has been formally implemented through

Mathematica. We have developed a program that permits to directly get the control law. This program also allows us to choose the number of parameters and then the controller structure which amounts to have other degrees of freedom in the optimization procedure.

Now, we briefly describe steps calculation down by the Mathematica program.

#### 4.3 Backstepping Control Design

**Step 1:** Let us start by looking at the first subsystem consisting of the state equation for  $\Omega_1$ . This system can be stabilized by using the angular displacement  $\Phi$  as the virtual control law.

$$\dot{\Omega}_1 = -\frac{1}{J_1}(C_1\Omega_1 - k\Phi + T_{tob}(\Omega_1)) \quad (21)$$

In order to find the virtual control law  $\Phi^d(\Omega_1)$ , we introduce the control Lyapunov function (CLF)

$$V_1(\Omega_1) = \frac{1}{2}\varepsilon^2 \quad (22)$$

where  $\varepsilon$  is the error signal  $\varepsilon = \Omega_1 - \Omega_1^d$ .

From the time derivative of (22), it comes out:

$$\dot{V}_1(\Omega_1) = \varepsilon \left( -\frac{1}{J_1}(C_1\Omega_1 - k\Phi + T_{tob}(\Omega_1)) \right) \quad (23)$$

We can strictly assign the CLF  $V_1$  by taking:

$$\Phi^d(\Omega_1) = \frac{1}{k}(C_1\Omega_1 + T_{tob}(\Omega_1) - \gamma_1 J_1 \varepsilon) \quad \text{with } \gamma_1 > 0.$$

**Step 2:** Let us now extend the first system to the subsystem.

$$\begin{cases} \dot{\varepsilon} = -\gamma_1 \varepsilon + \frac{k}{J_1} z \\ \dot{z} = \Omega_2 - \Omega_1 + \dot{\Phi}^d \quad \text{with } z = \Phi - \Phi^d \end{cases} \quad (24)$$

By introducing  $\Omega_2$  as a virtual control input, one can stabilize this system by the following CLF  $V_2$ :

$$V_2(\Omega_1, \Phi) = V_1 + \frac{1}{2} z^2 \quad (25)$$

As in step 1, we can assign the CLF  $V_2(\Omega_1, \Phi)$  by setting:

$$\Omega_2^d(\Omega_1, \Phi) = \Omega_1 - \dot{\Phi}^d - \frac{k}{J_1} \varepsilon - \gamma_2 z \quad \text{with } \gamma_2 > 0$$

**Step 3:** Now, let us consider the whole system which is given by the following equations:

$$\begin{cases} \dot{\varepsilon} = -\gamma_1 \varepsilon + \frac{k}{J_1} z \\ \dot{z} = y - \gamma_2 z - \frac{k}{J_1} \varepsilon \\ \dot{y} = -\frac{1}{J_2}(k\Phi + C_2(\Omega_2 - \Omega_{ref}) - u) - \dot{\Omega}_2^d \quad \text{with } y = \Omega_2 - \Omega_2^d \end{cases}$$

We select the Lyapunov function candidate  $V$  for the total system to be

$$V = V_2 + \frac{1}{2} y^2 \quad (26)$$

Finally, the control law  $u$  which stabilizes the overall system can be found in the same way as in the preceding steps.

$$u = k\Phi + C_2(\Omega_2 - \Omega_{ref}) - J_2(-\dot{\Omega}_2^d + z + \gamma_3 y) \quad (27)$$

with  $\gamma_3 > 0$ .

#### 4.4 Simulations results

The parameters, used for the simulations are shown in Table 1, and are taken from (Serrarens, *et al.*, 1998). They represent a typical case in oil well drilling operations. The desired rotary table speed is chosen as 10 (rad/s) which is within the common operating range for oil well drilling.

The effects of large torsional vibrations on the bit are very danger and even small amplitude stick-slip vibrations are thought to be a major cause of bit wear. The control objective is to bring the bit speed back to the desired speed while considerably minimizing these vibrations.

TABLE 1: NUMERICAL VALUES OF DRILLING SYSTEM AND MOTOR

Symbol	Description	Values
$\Omega_{ref}$	The reference angular velocity	10 [rad/s]
$J_1$	BHA+ 1/3 drill-string inertia	374 [kg m <sup>2</sup> ]
$J_2$	Rotary table + drive inertia	2122 [kg m <sup>2</sup> ]
$C_1$	BHA damping	0-150 [Nms/rad]
$C_2$	Rotary table damping	425[Nms/rad]
$k$	Drillstring stiffness	473[Nms/rad]

For this reason, a first set of requirements is expressed in terms of maximum overshoot, settling time at 2% and the limitation of control amplitude. These specifications are formulated as follows:

$$\min_{\theta} T_s / \{D \leq 10\% \text{ and } |u(t)| \leq 12 \cdot 10^5 \text{ Nm}\} \quad (28)$$

Using the  $\varepsilon$  subdifferential algorithm, the last problem is solved and the optimal settling time found is equal to 2 seconds. The optimal parameters are:  $[\gamma_1^{opt}, \gamma_2^{opt}, \gamma_3^{opt}] = [4.3538, 2.5952, 2.6089]$ .

Figure 5 shows the closed-loop system responses with the control input  $u$ . We observe that specifications are validated but the angular velocity of the rotary table  $\Omega_2$  has a large overshoot.

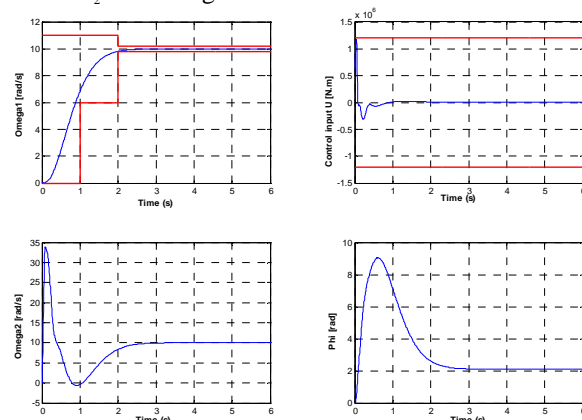


Fig. 5. Closed loop responses without  $\Omega_2$  constraints.

In order to ameliorate this response, we change specifications. So, the required performance for  $\Omega_1$  are expressed by the overshoot (<10 %), the settling

time at 60% (< 3 s), the response time at 2% (< 5 s) and the tracking error accuracy. For  $\Omega_2$ , demands are expressed by the overshoot (<80 %), the settling time at 80% (< 3 s), the response time at 2% (< 5 s) and the tracking error accuracy. The limitation of the control input is  $|u| < 12 \cdot 10^5$  (Nm). These specifications are formulated using template forms and the general optimization criterion has the following form:

$$J(\theta) = \sum_{i=1}^{n_i} (|y(\theta, t_i) - Y_{max}(t_i)|_+ + |Y_{min}(t_i) - y(\theta, t_i)|_+) \quad (29)$$

$$+ \sum_{i=1}^{n_i} (|u(\theta, t_i) - U_{max}(t_i)|_+ + |U_{min}(t_i) - u(\theta, t_i)|_+)$$

In figure 6, it may be observed that simulation results are in concordance with the required performance and the new optimal parameters are shown to be:  $[\gamma_1^{opt}, \gamma_2^{opt}, \gamma_3^{opt}] = [5.3397, 2.1204, 0.3980]$ .

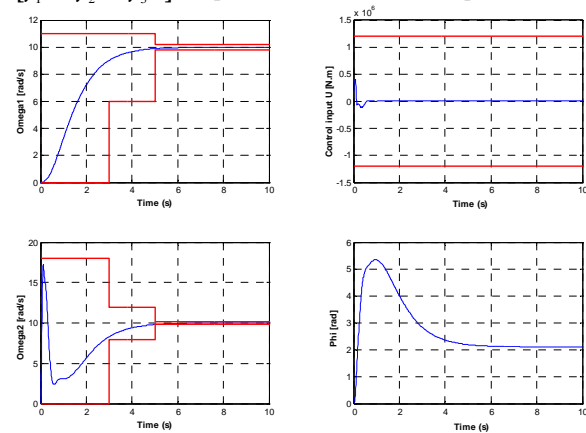


Fig. 6. Closed loop responses with  $\Omega_2$  constraints.

## 5. CONCLUSION

In order to design an optimal backstepping controller which verify temporal specifications, an exact formulation of the demands must be used. Corresponding criteria are generally nonsmooth, which often lead to complex optimization problems. In our case, backstepping is not only used to stabilize the rotary drilling system but also to reach given performance. A direct computation of the backstepping control law may be obtained through the developed Mathematica program. This program permits to manage controller structure and consequently the degree of freedom in optimization problem. The presented  $\varepsilon$  subdifferential algorithm can represent an effective tool in computer-aided design and for controllers retuning.

## REFERENCES

- Abdulgalil, F. and H. Siguerdidjane (2005), "Backstepping Design for Controlling Rotary Drilling System," in *Proc. of Conference on Control Applications*, Toronto, Canada, 2005, pp. 120-124.
- Boyd, S. and C. Barrat (1991), *Linear Controller Design: Limits of Performance*. Englewood Cliffs. New Jersey: Prentice-Hall.
- Burke, J.V., A.S. Lewis and M.L. Overton (2002), "Approximating subdifferential by random sampling of

- 
- gradients," *Mathematics of Operations Research*, pp. 567-584.
- Burke, J.V., A.S. Lewis and M.L. Overton (2004), "A robust gradient sampling algorithm for nonsmooth, nonconvex optimization," Revised version to appear in *SIAM Journal on Optimization*.
- Clarke, F.H. (1990), *Optimization and Nonsmooth Analysis*. John Wiley, New York, 1983. Reprinted by SIAM, Philadelphia.
- Doyle, J.C., K. Zhou and B. Bodenheimer (1989), "Optimal control with mixed  $H_2$  and  $H_\infty$  performance objective," in *Proc. of American Control Conference*, Pittsburgh, Pennsylvania, 1989, pp. 2065-2070.
- Kanellakopoulos M. I and P. V. Kokotovic (1995), *Nonlinear and Adaptive Control Design*. New York: Wiley.
- Rosenwasser, E. and R. Yusupov (2000), *Sensitivity of Automatic Control Systems*. CRC Press Control Series, Boca Raton.
- Serrarens, A.F.A, M.J.G. van de Molengraft, J.J. Kok and L. Van den Steen (1998).  $H_\infty$  Control for suppressing stick slip in oil well drillstrings *IEEE Control systems*, Vol. 18, No. 2, pp. 19-30.

# Références Bibliographiques

(A)

- [Abd06] Abdulgalil. F., *Commande non linéaire dans les systèmes de forage pétrolier : contribution à la suppression du phénomène de “stick-slip”*, Thèse de Doctorat, Université Paris-Sud XI, 2006.
- [Ack02] Ackermann, J., *Robust control: the parameter space approach*. 2<sup>nd</sup> edition, Springer-Verlag, Berlin, Germany, 2002.
- [Aey99] Aeyels. D, F. Lamnabhi-Lagarrigue, and A. J. van der Schaft. *Stability and stabilization of nonlinear systems*. Springer -Verlag, London, 1999.
- [Ala99] Alazard D., Cumer C., Apkarian P., Gauvrit M. and Ferreres G., *Robustesse et Commande Optimale*. Cépaduès, 1999.
- [And90] Anderson. B and J. B. Moore., *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, 1990.
- [Apk95] Apkarian. P and P. Gahinet, A convex characterization of gain-scheduled  $H_\infty$  controllers, *IEEE Trans. Aut. Control*, vol. 40, pp. 853–864. See also pp. 1681, 1995.
- [Apk03] Apkarian. P, D. Noll, and H. D. Tuan, Fixed-order  $H_\infty$  control design via a partially augmented Lagrangian method, *Int. J. Robust Nonlinear Control*, vol. 13, pp. 1137-1148, 2003.
- [Apk04] Apkarian. P, D. Noll, J.-B. Thevenet, and H. D. Tuan, A spectral quadratic-SDP method with applications to fixed-order  $H_2$  and  $H_\infty$  synthesis, in *Proceedings Asian Control Conf.*, Melbourne, Australia, 2004.
- [Apk05] Apkarian. P and D. Noll., Controller Design via Nonsmooth Multidirectional Search, *SIAM J. Control Opt.* vol. 44(6), pp.1923-1949, 2005.
- [Apk06] Apkarian. P and D. Noll. Nonsmooth  $H_\infty$  synthesis. *IEEE Transactions on Automatic Control*, vol. 51(1), pp. 71–86, 2006.
- [Arm66] Armijo. L., Minimization of Functions having Lipschitz Continuous First Partial Derivatives. *Pacific Journal of Mathematics*, vol. 16, pp. 1-3, 1966.
- [Ast98] Åström K. J, H. Panagopoulos and T. Hägglund, Design of PI Controllers based on nonconvex optimization, *Automatica*, vol. 34(5), pp. 585-601, 1998.
- [Ast05] Åström K. J and T. Hägglund., *Advanced PID Control, The Instrumentation, Systems, and Automation Society*, Research Triangle Park, 2005.
- [Ast00] Åström K. J., Limitations on control system performance. *Europ. J. Control*, vol. 6(1), pp. 2-20. 2000.
- [Ath66] Athans M. and P. Falb., *Optimal Control*, McGraw Hill, 1966.

[Aut08] <http://www.autodiff.org/>.

(B)

- [Bal96] Balas. G. J., R. Lind, and A. Packard. Optimal scaled H1 full information control synthesis with real uncertainty. *AIAA J. Guidance, Control and Dynamics*, vol. 19(4), pp. 854-862, 1996.
- [Big71] Biggs M.C. Minimization Algorithms Making Use of Nonquadratic Properties of the Objective Function. *J. Inst. Maths. Applics.*, vol. 8, pp. 315-327, 1971.
- [Bla34] Black H. S., Stabilised Feedback Amplifiers, *The Bell System Technical Journal*, vol 13, pp. 1-18, 1934.
- [Blo94a] Blondel. V., *Simultaneous Stabilization of Linear Systems*. Lecture Notes in Control and Information Sciences 191. Springer, Berlin, 1994.
- [Blo94b] Blondel. V and M. Gevers, Simultaneous stabilizability question of three linear systems is rationally undecidable, *Mathematics of Control, Signals, and Systems*, vol. 6, pp. 135–145, 1994.
- [Blo97] Blondel. V. and J. H. Tsitsiklis, NP-hardness of some linear control design problems, *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118-2127, 1997.
- [Blo99] Blondel. V., Simultaneous stabilization of linear systems and interpolation with rational functions. In Vincent D. Blondel, Eduardo D. Sontag, M. Vidyasagar, and Jan C. Willems, editors, *Open Problems in Mathematical Systems and Control Theory*, pages 53–56. Springer Verlag, London, 1999.
- [Blo00] Blondel, V. D. and J. N. Tsitsiklis., A survey of computational complexity results in systems and control. *Automatica*, vol. 36(9), pp. 1249-1274., 2000.
- [Bod40] Bode H. W., Relation between Attenuation and Phase Feedback Amplifier Design, *The Bell System Technical Journal*, vol 19, pp. 421-454, 1940.
- [Bod45] Bode H. W., *Networks Analysis and Feedback Amplifiers Design*, Van Nostrand, New York, 1945.
- [Bon97] Bonnans J-F., et J-C. Gilbert, C. Lemaréchal et C. Sagastizàbal, *Optimisation Numérique* Springer, 1997.
- [Boy90] Boyd. S., Baratt C. and Norman, S., Linear Controller Design: Limits of Performance via convex optimization, *Proceedings of the IEEE*, vol 78-3, pp. 529-574, 1990.
- [Boy91] Boyd. S. and C. Barratt., *Linear Controller Design: Limits of Performance*, Prentice and Hall, Englewood Cliffs, New Jersey, 1991.
- [Boy04] Boyd. S and L. Vandenberghe., *Convex Optimization*, Cambridge University Press, 2004.
- [Bre73] Brent. R., *Algorithms for Minimization without Derivatives*, Prentice-Hall, Inc. 1973.
- [Bry75] Bryson A. E. and Y. C. Ho., *Applied Optimal Control*, Hemisphere Publishing, 1975.
- [Bun71] Bunch J. R. and Parlett B. N., Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations, *SIAM Journal on Numerical Analysis*, vol. 8(4), 639-655, 1971.

- [Bur02] Burke, J.V., Lewis, A.S. and Overton M.L. Approximating Subdifferentials by Random Sampling of Gradients. *Mathematics of Operations Research*, pp. 567-584, 2002.
- [Bur03] Burke, J.V., A. S. Lewis, and M. L. Overton, A nonsmooth, nonconvex optimization approach to robust stabilization by static output feedback and low-order controllers, in *Proceedings ROCOND*, Milan, 2003.
- [Bur04] Burke, J.V., Lewis, A.S. and Overton M.L. A Robust Gradient Sampling Algorithm for Nonsmooth, Nonconvex Optimization. *Revised version to appear In SIAM Journal on Optimization*, June 2004.
- [Bur06] Burke, J. V, D. Henrion, A. S. Lewis, M. L. Overton, HIFOO: A Matlab Package for Fixed-order Controller Design and H-infinity Optimization, *Proceedings of the IFAC Symposium on Robust Control Design*, Toulouse, July 2006.
- [Bur06a] Burke, J. V, D. Henrion, A. S. Lewis, M. L. Overton, Stabilization via Nonsmooth, Nonconvex Optimization, *IEEE Transactions on Automatic Control*, Vol. 51, No. 11, pp. 1760-1769, November 2006.

(C)

- [Car61] Carroll, C. W., The created response surface technique for optimizing nonlinear restrained systems. *Operations Research*, vol. 9, pp. 169-184, 1961.
- [Cau47] Cauchy, A. L., Méthode Générale pour la Résolution des Systèmes d'Equations Simultanées. *Comptes Rendus de l'Académie des Sciences de Paris*, vol. 25, pp. 536-538, 1847
- [Che98] Chen, B. M.,  $H^\infty$  Control and Its Applications, vol. 235 of Lectures Notes in Control and Information Sciences, Springer Verlag, New York, Heidelberg, Berlin, 1998.
- [Cla75] Clarke, F.H., Generalized gradients and applications. *Trans. Amer. Math. Soc.*, vol. 205, pp. 247-262, 1975.
- [Cla83] Clarke, F.H., *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York, 1983.
- [Cla90] Clarke, F.H., *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983. Republished as vol. 5 of Classics in Applied Mathematics, SIAM, 1990.
- [Cor02] Cortes, H. R., *Sur la commande non linéaire via assignation de l'interconnexion et de l'amortissement*. Thèse de Doctorat, Université Paris-Sud XI, 2002.
- [Cul94] Culioli, J.-C., Introduction à l'Optimisation. Edition Ellipses, 1994.

(D)

- [Dav90] Davison E. J., (ed.), Benchmark problems for control system design, tech. rep., Oxford, Pergamon Press, IFAC Technical Committee Reports, 1990.
- [Del05] Delmond, F, C. Cumer and D. Alazard, Controller Gain Adjustments for Closed-loop Modal Requirements. *IFAC World Congress*, Prague, 4-8 July 2005.
- [Den91] Dennis, J.E., Torczon, V. Direct Search Methods on Parallel Machines. *SIAM Journal on Optimization*, vol. 1(4), pp. 448-474, 1991.



- [Den96] Dennis. J.E. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.
- [Des75] Desoer. C. A. and M. Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press, New York, 1975.
- [Des96] Desai. R. and Patil R. SALO: Combining Simulated Annealing and Local Optimization for Efficient Global Optimization. *In Proceedings of the 9<sup>th</sup> Florida AI Research Symposium (FLAIRS-96)*, pp. 233-237, June 1996.
- [Dev02] Devaud. E. and H. Siguerdidjane, A missile autopilot based on feedback linearization, *European Journal of Control*, vol. 8, 2002.
- [Dol03] Dolan, E.D., Lewis, R.M., Torczon, V. On the Local Convergence of Pattern Search. *SIAM Journal on Optimization*, vol. 14(2), pp. 567-583, 2003.
- [Doy81] Doyle J., Stein G., Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis, *IEEE Transactions on Automatic Control*, vol. AC-26, no 1, pp. 4-16, 1981.
- [Doy89a] Doyle J. C., Glover K., Khargonekar P. P., and Francis B. A., State-Space Solutions to Standard  $H_2$  and  $H_\infty$  control problems, *IEEE Transactions on Automatic Control*, vol. 34, no 8, p. 831-847, 1989.
- [Doy89b] Doyle J. C., K. Zhou and B. Bodenheimer, Optimal Control with Mixed  $H_2$  and  $H_\infty$  Performance Objective, *In Proc. of American Control Conference*, Pittsburgh, Pennsylvania, pp. 2065-2070, 1989.
- [Duc99] Duc. G and S. Font., *Commande  $H_\infty$  et  $\mu$ -Analyse : Des Outils pour la Robustesse*, édition Hermès, 1999.

(E)

- [Elg97] El Ghaoui. L., F. Oustry, and M. Ait Rami, A cone complementarity linearization algorithm for static output-feedback and related problem, *IEEE Trans. on Automatic Control*, vol. 42, no. 8, pp. 1171-1176, 1997.

(F)

- [Fan89] Fan. M. K. H., L. Wang., J. Joninckx, and A. Tits, Software package for Optimization-Based Design with user-supplied Simulators, *IEEE Cont. Syst. Mag* , vol. 9, no. 1, pp. 66-71, 1989.
- [Far01] Fares. B., P. Apkarian, and D. Noll, An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory, *Int. J. Control*, vol. 74, no. 4, pp. 348-360, 2001.
- [Far02] Fares. B., D. Noll, and P. Apkarian, Robust control via sequential semidefinite programming, *SIAM J. Control Optim.*, vol. 40, no. 6, pp. 1791-1820, 2002.
- [Fer95] Ferreres. G. De l'utilisation des outils de robustesse pour la commande adaptative. PhD thesis, Institut National Polytechnique de Grenoble, France, January 1995.
- [Fer96] Ferreres. G, G. Scorletti and V. Fromion. Advanced computation of the robustness margin. *In Proc. IEEE Conf. on Decision and Control*, December 1996.

- [Feu78] Feurer. A. and Morse. A. S., Adaptive control of single-input, single-output linear systems, *IEEE Transactions on Automatic Control*, vol. 23(4), pp.557-569, 1978.
- [Fle87] Fletcher R., *Practical Methods of Optimization*. John Wiley & Sons, Chichester (second edition), 1987.
- [Fon95] Font. S, *Méthodologie pour Prendre en Compte la Robustesse des Systèmes Asservis: Optimisation  $H_\infty$  et Approche Symbolique de la Forme Standard*. Thèse de Doctorat en Automatique, Université Paris-Sud et Supélec, 1995.
- [Fra84] Francis B. A. and G. Zames., On  $H_\infty$ -optimal Sensitivity Theory for SISO feedback systems, *IEEE Transactions on Automatic Control*, vol 29, pp. 9-16, 1984.
- [Fra87] Francis B. A., A Course in  $H_\infty$  Control Theory. *Lecture Notes in Control and Information Sciences*, no. 88, 1987.
- [Fra86] Franklin G. F., J. D. Powell, and A. Emami-Naeni. *Feedback Control of Dynamic Systems*. Addison-Wesley, 1986.
- [Fre94] Freeman. A. R. and P. V. Kokotovic., Tools and procedures for robust control of nonlinear systems, *Proceedings of IEEE Conference on Decision and Control*, pp. 3458-3463, 1994.
- [Fre95] Freeman. A. R. and P. V. Kokotovic. Robust integral control for a class of uncertain nonlinear systems, *Proceedings of IEEE Conference on Decision and Control*, pp. 2245-2250, 1995.
- [Fre88] Freudenberg J. S. and D. P Looze., Frequency Domain Properties of Scalar and Multivariable Feedback Systems. *Lecture Notes in Control and Information Sciences* 104, Springer-Verlag, Berlin, etc., 1988.
- [Fri55] Frisch. K. R., *The Logarithmic Potential Method for Convex Programming*, Memorandum, Institute of Economics, University of Oslo, Oslo, Norway, 1955.
- [Fro95a] Fromion. V., *Une Approche Incrémentale de la Robustesse Non Linéaire ; Application au Domaine de l'Aéronautique*. Thèse de Doctorat en Automatique, Université Paris-Sud et Supélec, 1995.
- [Fro95b] Fromion. V, S. Monaco, and D. Normand-Cyrot, A Possible Extension of  $H_\infty$  Control to The Nonlinear Context. In *Proc. IEEE Conf. on Decision and Control*, December 1995.
- [Fro97] Fromion. V, G. Scorletti, and G. Ferreres. Nonlinear performance of a PI controlled missile: a simple explanation. In *Proc. IEEE Conf. on Decision and Control*, December 1997.
- [Fuj90] Fujita. M., F.Matsumura, and M. Shimizu.,  $H_\infty$  robust control design for a magnetic suspension system. In *Proceedings of the 2<sup>nd</sup> Int. Symp. on Magnetic Bearings*, pp. 349-356, 1990.

(G)

- [Gah93] Gahinet. P and P. Apkarian, A LMI-based parametrization of all  $H_\infty$  controllers with applications, in *Proc. IEEE Conf. on Decision and Control*, San Antonio, Texas, pp. 656–661, 1993.

- [Gah94] Gahinet. P and P. Apkarian, Explicit Controller Formulas for LMI-based  $H_\infty$  Synthesis, *American Control Conference*, pp. 2396-2400, Maryland, 1994.
- [Gan86] Gangassas. D., K. Bruce, J. Blight, and U. Ly, Application of Modern Synthesis to Aircraft Control: Three Case studies, *IEEE Transactions on Automatic Control*, vol AC-31, pp. 995-1014, 1986.
- [Gar04] Garcia. D, A. Karimi and R. Longchamp, Robust PID controller tuning with specification on modulus Margin, *American Control Conference*, Boston, Massachusetts, pp. 3297-3302, 2004.
- [Gas88] Gazzina. A., How to control unstable missile airframes: methodology and limitations. In *AGARD conf. Proc.*, 1988.
- [Ger00] Germundsson, R., *Mathematica* Version 4. *Mathematica Journal*, vol. 7, pp. 497-524, 2000.
- [Gil91] Gilbert J.C., G. Le Vey and J. Masse, La différentiation automatique des fonctions représentées par des programmes. Rapport de Recherche INRIA N° 1557, Rocquencourt, 1991.
- [Gil92] Gilbert J.C. and Nocedal J., Global Convergence Properties of Conjugate Gradient Methods for Optimization. *SIAM Journal on Optimization*, vol. 2, pp. 21-42, 1992.
- [Gil07] Gilbert J.C., Éléments d'Optimisation Différentiable - Théorie et Algorithmes. Notes de cours ENSTA. (disponible sur [www-rocq.inria.fr/~gilbert/ensta/optim.html](http://www-rocq.inria.fr/~gilbert/ensta/optim.html))
- [Gil81] Gill. P. E, W. Murray, and M. Wright, *Practical Optimization*. London, UK, Academic Press, 1981.
- [Glo84] Glover .K., All optimal Hankel-norm approximations of linear multivariable systems and their  $L_\infty$ -error bounds, *Int. J. Control*, vol. 39, pp. 1115–1193, 1984.
- [Glo88] Glover K., and Doyle J.C., State-space Formulae for all Stabilizing Controllers that Satisfy an  $H_\infty$  Norm Bound and Relations to Risk Sensitivity. *Syst. Cont. Letters*, vol. 11, pp. 167-172, 1988.
- [Glo93] Glover F., Taillard E., and de Werra D. A User's Guide to Tabu Search. *Annals of Operations Research*, vol. 41, pp. 3-28, 1993.
- [Gol89] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [Gol65] Goldstein. A. A., On steepest descent. *SIAM Journal on Control*, vol. 3, pp. 147-151, 1965.
- [Gol77] Goldstein. A. A., Optimization of Lipschitz Continuous functions. *Mathematical Programming*, vol. 13, pp. 14-22, 1977.
- [Gri91] Griewank. A. and G. Corliss., *Automatic Differentiation of Algorithms : Theory, Implementation and Application*, SIAM, Philadelphia 1991.
- [Gri96] Grigoriadis. K. M. and R. Skelton, Low-order control design for LMI problems using alternating projection methods, *Automatica*, vol. 32, no. 8, pp. 1117-1125, 1996.

- [Gri99] Grigoriadis. K. M. and E. B. Beran, Alternating projection algorithms for linear matrix inequalities problems with rank constraints," in *Advances on Linear Matrix Inequality Methods in Control*, L. El Ghaoui and S.-I. Niculescu, Eds. SIAM, pp. 251-267, 1999.
- [Gri97] Grippo L. and Lucidi S. A Globally Convergent Version of the Polak-Ribière Conjugate Gradient Method. *Mathematical Programming*. 78A, No.3, pp.375-391, 1997.
- (H)
- [Haf93] Haftka. R.T. and Z Gürdal., *Elements of Structural Optimization*. Kluwer Academic Publishers, Third edition. (1993).
- [Hba02a] Hbaïeb. S., *Analyse de Cahier des Charges en Automatique par Optimisation Convexe*, Thèse de Doctorat en Automatique, Université Paris-Sud et Supélec, 2002.
- [Hba02b] Hbaïeb. S, S. Font, P. Bendotti, C-M. Falinower, Convex Optimal Control Design via Piecewise Linear Approximation, *15<sup>th</sup> IFAC World Congress*, Barcelona, Espagne, 2002.
- [Hen04] Henni. B. A., *Analyse de la robustesse de la stabilité pour des systèmes non linéaires dynamiques*, Thèse de Doctorat en Automatique, Université Paris-Sud et Supélec, 2004.
- [Hen05] Henrion. D, J. Löfberg, M. Kocvara, M. Stingl Solving polynomial static output feedback problems with PENBMI, *Proceedings of the joint IEEE Conference on Decision and Control and European Control Conference*, Sevilla, Spain, December 2005.
- [Hen06] Henrion. D., Solving Static Output Feedback Problems by Direct Search Optimization, *Proceedings of the joint IEEE CCA/CACSD/ISIC Conference*, Munich, Germany, October 2006.
- [Hig93] Higham. N. J., Optimization by direct search in matrix computations, *SIAM J. Matrix Anal. Appl*, 14(2): 317-333, 1993.
- [Hig02a] [www.maths.man.ac.uk/~higham/mctoolbox](http://www.maths.man.ac.uk/~higham/mctoolbox).
- [Hig02b] Higham. N. J., *Accuracy and Stability of Numerical Algorithms*, Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002; sec. 20.5.
- [Hin98] Hindi. H, B. Hassidi, S. Boyd., Multiobjective  $H_2/H_\infty$ -optimal Control via Finite Dimensional Q-Parameterization and LMI, *American Control Conference*, pp. 3244-3248, 1998.
- [Hin05] Hindmarsh. A. C, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," *ACM Transactions on Mathematical Software*, 31(3), pp. 363-396, 2005. Also available as LLNL technical report UCRL-JP-200037.
- [Hin86] Hinrichsen. D and A. J. Pritchard, \Stability radii of linear systems," *Systems and Control Letters*, vol. 7, pp. 1-10, 1986.
- [Hir93] Hiriart-Urruty. J.-B., C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Grundlehren der mathematischen Wissenschaften 305-306. Springer-Verlag, 1993.

- [Hor74] Horisberger. H. P. and P. R. Belanger, Solution of the optimal constant output feedback problem by conjugate gradients, *IEEE Trans. Aut. Control*, vol. 19, pp. 434–435, 1974.
- [Hor63] Horowitz I. M., *Synthesis of Feedback Systems*, Academic Press, New York, 1963.
- [Hor82] Horowitz I. M., Quantitative Feedback Theory, *IEE Proc.*, vol 129-D, pp. 215-226, 1982.
- [Hua95] Huang. Y. and W.M. Lu. An LFT approach to autopilot design for missiles. *In Proc. American Control Conf.*, pp. 2990-2994, 1995.
- [Hun82] Hung. Y. S. and A. G. J. MacFarlane, Multivariable feedback: A classical approach, *Lectures Notes in Control and Information Sciences*, Springer Verlag, New York, Heidelberg, Berlin, 1982.

(I)

- [Iba01] Ibaraki. S and M. Tomizuka, Rank minimization approach for solving BMI problems with random search," in *Proceedings American Control Conference*, pp. 25-27, 2001.
- [Iso95] Isidori. A and W. Kang. H<sub>∞</sub> Control via Measurement Feedback for General Nonlinear Systems. *IEEE Transactions on Automatic Control*, vol. 40(3), pp. 466-472, 1995.
- [Iso95a] Isidori. A., *Nonlinear control systems*. Springer Verlag, 1995.
- [Iwa94] Iwasaki. T and R. E. Skelton., All Controllers for the General H<sub>∞</sub> Control Problem: LMI Existence Conditions and State-Space Formulas, *Automatica*, vol. 30, no. 8, pp. 1307-117, 1994.

(J)

- [Jac77] Jacobson. D.H., *Extensions of Linear-Quadratic Control, Optimization and Matrix Theory*. Academic Press, New York, 1977.
- [Jur78] Jurdjević. V. and J.P. Quinn. Controllability and Stability. *Journal of Differential Equations*, vol. 28, pp. 381-389, 1978.

(K)

- [Kal61] Kalman. R. E. and R.S. Bucy, New Results in Linear Filtering and Prediction Theory, *Transactions. ASME, Series D*, vol.83, pp. 95-107, Dec, 1961.
- [Kan04] Kanev. S., C. Scherer, M. Verhaegen, and B. De Schutter, Robust output-feedback controller design via a local BMI optimization, *Automatica*, vol. 40, no. 7, pp. 1115-1127, 2004.
- [Kan91] Kanellakopoulos. I., P. V. Kokotović., A. S. Morse., Systematic design of adaptive controllers for feedback linearizable systems, *IEEE Transaction On Automatic Control*, vol. 36(11), pp. 1241-1253, 1991.
- [Kee88] Keel. L. H., S. P. Bhattacharyya, and J. W. Howze, Robust control with structured perturbations, *IEEE Trans. Aut. Control*, vol. 36, pp. 68–77, 1988.
- [Kel99] Kelley, C.T. Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, 10(1), 43-45 (1999).

- [Kha02] Khalil H. K., *Nonlinear Systems*, Third Edition, Prentice Hall, 2002.
- [Knu97] Knuth. D. E., *The Art of Computer Programming: Seminumerical Algorithms*, Third Edition Reading, Massachusetts: Addison-Wesley, 1997.
- [Kok89] Kokotović, P. V. and Sussmann, H. J., A positive real condition for global stabilization of nonlinear systems, *Systems and Control Letters*, vol. 13, pp. 125-133, 1989.
- [Kol03] Kolda, T.G., R.M. Lewis., and V. Torçzon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Journal on Optimization*, 45(3), 385-482, 2003.
- [Krs95] Krstić. M, I. Kanellakopoulos, and P. V. Kokotović. *Nonlinear and Adaptive Control Design*. Wiley, New York, 1995.
- [Kuc74] Kučera. V., *Discrete Linear Control: The Polynomial Equation Approach*, John Wiley & Sons, 1974.
- [Kwa72] Kwakernaak H. and R. Sivan., *Linear Optimal Control Systems*, John Wiley & Sons, 1972.
- [Kwa91] Kwakernaak H. and R. Sivan., *Modern Signals and Systems*. Prentice and Hall, Englewood Cliffs, New Jersey, 1991.
- (L)
- [Lag98] Lagarias, J.C., J.A. Redds., M.H. Wright., P.E Wright., Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, vol. 9(1), pp. 112-147, 1998.
- [Lei01] Leibfritz. F., An LMI-based algorithm for designing suboptimal static  $H_2/H_\infty$  output feedback controllers," *SIAM J. Control Optim.*, vol. 39, no. 6, pp. 1711-1735, 2001.
- [Lei02] Leibfritz. F and E. M. E. Mostafa, An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems, *SIAM J. Optim.*, vol. 12, no. 4, pp. 1048-1074, 2002.
- [Lei03] Leibfritz. F., *COMPLeib, Constraint Matrix-Optimization Problem Library: A Collection of Test Examples for Nonlinear Semidefinite Programs, Control System Design and Related Problems*, Technical report, Universität Trier, 2003.
- [Lar93] P. de Larminat, *Commande des systèmes linéaires*. Hermès, 1993.
- [Las06] Lassami. B, F. Abdulgalil, S. Font and H. Siguerdidjane, Parametric Adjustment of a Backstepping Controller by Nonsmooth Optimization: Application to a Rotary Drilling System, *International Journal of Tomography and Statistics*, vol. 6, No. S07, pp. 134-139, 2006.
- [Lem89] Lemaréchal. C., *Méthode Numérique d'Optimisation*, Fascicule INRIA, Collection Didactique, No.6, 1989.
- [Lem02] Lemauff. F and G. Duc., Design of Structured Controllers with Respect to Various Specifications using Robust Control and Genetic Algorithms, *American Control Conference*, May 2002.

- [Lew99] Lewis, R.M., Torczon, V. Pattern Search Methods for Bound Constrained Minimization. *SIAM Journal on Optimization*, vol. 9(4), pp. 1082-1099, 1999.
- [Lew00] Lewis, R.M., Torczon, V., Trosset, M.W. Direct Search Methods: Then and Now. *Journal of Computational and Applied Mathematics*, vol. 124(1-2), pp. 191-207, 2000.
- [Lew02] Lewis, R.M., Torczon, V. A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds. *SIAM Journal on Optimization*, vol. 12(4), pp. 1075-1089, 2002.
- [Lev96] Lévine, J., J. Lottin, and J-C. Ponsart. A nonlinear approach to the control of magnetic bearings. *IEEE Transactions on Automatic Control*, vol. 4(5), pp.524-544, 1996.
- [Lue04] Luersen, M.A., Le Riche, R., Guyon, F. A constrained, globalized, and bounded Nelder-Mead method for engineering optimization. *Structural and Multidisciplinary Optimization*, 27, 43-54 (2004).
- [Luk90] Luksàn L. Computational Experience with Improved Variable Metric Methods for Unconstrained Minimization. *Kybernetika*, vol. 26, pp. 415-431, 1990.
- [Luk94] Luksàn L. Computational Experience with Known Variable Metric Updates. *J. Optimization Theory Appl.* Vol.83, No.1, pp. 27-47, 1994.
- [Luk95] Luksàn L. Simple Scaling for Variable Metric Updates. Technical Report V-611, Institute of Computer Science, Academy of Sciences of the Czech Republic, May 1995.
- [Lyn67a] Lyness, J. N., and C. B. Moler, Numerical differentiation of analytic functions, *SIAM J. Numer. Anal.*, Vol. 4, pp. 202-210, 1967.
- [Lyn67b] Lyness, J. N., Numerical algorithms based on the theory of complex variables, *Proc. ACM 22nd Nat. Conf.*, Thompson Book Co., Washington DC, pp. 124-134, 1967.
- [Lyu83] Ly. U., E. Bryson, and R. H. Cannon, Design of Low-order Compensators using Parameter Optimization, in *Applications of Nonlinear Programming to Optimization and Control*, Laxenburg, Austria, IFAC, 1983.
- (M)
- [Mac77] MacFarlane A. G. J. and B. Kouvaritakis., A Design Technique for Linear Multivariable Feedback Systems, *Int. J. Control*, vol.25, no 6, pp. 875-883, 1977.
- [Mac79] MacFarlane A. G. J., *Frequency Response Methods in Control Systems*, Selected Reprints, IEEE Press, 1979.
- [Mac89] Maciejowski. J., *Multivariable Feedback Design*, Addison-Wesley, Wokingham, England, 1989.
- [Mag02] Magni. J. F., *Robust Modal Control with a Toolbox for Use with MATLAB®*, Kluwer Academics/Plenum publishers,, 2002.
- [Mak87] Mäkilä. P. M. and H. T. Toivonen, Computational Methods for Parametric LQ Minimization: A survey, *IEEE Transactions on Automatic Control*, vol AC-32, pp. 658-671, 1987.

- [Mar00] Martins. J., I. Kroo, and J. Alonso. An automated method for sensitivity analysis using complex variables. In *Proceedings of the 38th Aerospace Sciences Meeting*, Reno, USA, 2000.
- [Mar01] Martins. J., P. Sturdza, and J. Alonso. The connection between the complex-step derivative approximation and algorithmic differentiation. In *Proceedings of the 39th Aerospace Sciences Meeting*, Reno, USA, 2001.
- [Mat90] Matsumura. F., M. Fujita, and K. Okawa. Modelling and control of magnetic bearing systems achieving a rotation around the axis of inertia. in *Proceedings of the 2nd Int. Symp. on Magnetic Bearings*, pp. 273-280, 1990.
- [May70] Mayr O., *The Origins of Feedback Control*. Cambridge, MA: MIT Presse, 1970.
- [Mck98] McKinnon, K.I.M. Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, 9(1), 148-158 (1998).
- [Met53] Metropolis. N, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21, 1087-1091, 1953.
- [Min86] Minoux. M., *Mathematical Programming: Theory and Algorithms*, John Wiley & Sons, 1986.
- [Mor94] Moré J. J. and Thuente D.J. Line Search Algorithms with Guaranteed Sufficient Decrease. *ACM Transactions on Mathematical Software*, vol. 20(3), pp. 286-307, 1994.
- [Mor85] Morgenstein, J. How to compute fast a function and all its derivatives, a variation of the Theorem of Baur-Strassen, vol. 16, pp. 60-62. *SIGACT News*, 1985.
- [Mou02] Mouyon. P, C. Cumer and Y. Lossier, Retouche de Correcteurs, In *Conférence Internationale Francophone d'Automatique*, Nantes, France 2004.
- [Muh91] Muhlenbein H., M. Sconish and J. Born., The Parallel Genetic Algorithm as Function Optimizer. *Parallel Computing*, vol. 17, pp. 619-632, 1991.
- [Muh93] Muhlenbein H., and D. Schlierkamp-Voosen., Predictive Models for the Breeder Genetic Algorithm: Continuous Parameter Optimization. *Evolutionary Computation*, vol.1, 1993.

(N)

- [Naz94] Nazareth. L., The Newton-Cauchy Framework: A Unified Approach to Unconstrained Nonlinear Minimization, *Lecture Notes in Computer Science 769*. Springer-Verlag, 1994.
- [Nel65] Nelder, J.A and R. Mead. , A simplex for function minimization. *Computer Journal*, vol. 7, pp. 308-313, 1965.
- [Nic93] Nichols R.A., R.T. Reichert, and W.J. Rugh. Gain scheduling for H-Infinity controllers: A flight control example. *IEEE Trans Control Sys Tech.*, vol. 1(2), pp.69-69, 1993.
- [Nij90] Nijmeijer, H., Van der Shaft. A.J., *Nonlinear dynamical control systems*, Springer Verlag, 1990.



- [Nem94] Nemirovskii. A, Several NP-Hard problems arising in robust stability analysis, *Mathematics of Control, Signals, and Systems*, vol. 6 , pp. 99–105, 1994.
- [Neu71] Neuman. C. P. and A. K. Sood., Sensitivity functions for multi-input, linear time-invariant systems, *Int. J. Control*, vol. 13, pp. 1137-1150, 1971.
- [Neu72] Neuman. C. P. and A. K. Sood., Parameter sensitivity in linear systems, a controllability viewpoint, *Proc. IEE*, vol. 119. pp. 1217-1219, 1972.
- [Noc99] Nocedal. J. and Wright S.J., *Numerical Optimization*, Springer-Verlag, 1999.
- [Nol04] D. Noll, M. Torki, and P. Apkarian, Partially augmented Lagrangian method for matrix inequality constraints, *SIAM J. Optim.*, vol. 15, no. 1, pp. 161-184, 2004.
- [Non94] Nonami. K. and T. Ito.  $\mu$ -synthesis of a flexible rotor magnetic bearing system. *In Proceedings of the 4th Int. Symp. on Magnetic Bearings*, 1994.
- [Nyg32] Nyquist H., Regeneration theory, *The Bell System Technical Journal*, vol. 11, p. 126-147, 1932.

(O)

- [Ore74] Oren S.S. On the Selection of Parameters in Self-Scaling Variable Metric Algorithms. *Mathematical Programming*, vol. 7, pp. 351-367, 1974.
- [Ors04] Orsi. R., U. Helmke, and J. B. Moore, A Newton-like method for solving rank constrained linear matrix inequalities, in *Proc. 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, pp. 3138-3144, 2004.
- [Ous94] Oustaloup A., *La robustesse: Analyse et synthèse de commandes robustes*. Hermès, Paris, 1994.

(P)

- [Par95] Pardalos P.M., Pitsoulis L., Mavridou T., and Resende M.G.C. Parallel Search for Combinatorial Optimization: Genetic Algorithms, Simulated Annealing, Tabu Search and GRAS. In A. Ferreira and J. Rolin, Editors, *Parallel Algorithms for Irregularly Structured Problems*, Proceedings of the Second International Workshop-Irregular'95, vol. 980 of Lectures Notes in Computer Science, pp. 317-331. Springer-Verlag, 1995.
- [Pat02] Patel. V. V., G. Deodhare, T. Viswanath. Some applications of randomized algorithms for control system design. *Automatica*, vol. 38, pp. 2085-2092, 2002.
- [Per93] Perttunen, C.D., Jones, D.R., Struckman, B.E. Lipschitzian Optimization without a Lipschitz Constant. *Journal of Optimization Theory and Applications*, vol. 79(1), pp. 157-181, 1993.
- [Pol84] Polak. E, D. Q. Mayne, and D. Stimler, Control System Design via Semi-infinite Optimization: A Review, *Proc. IEEE*, 72(12): 1777-1794, December 1984.
- [Pol85] Polak. E, P. Siegel, T. Wu, W. T. Nye, and D. Q. Mayne, DELIGHT.MIMO: An Interactive Optimization based Multivariable Control System Design Package, *In Computer Aided Control Systems Engineering*, M. Jamshidi and C. J Herget, Eds. Amsterdam, The Netherlands: North-Holland, 1985. Reprinted from *IEEE Cont.Syst. Mag.*, no. 4, 1982.

- [Pol87] Polyak, B.T. *Introduction to Optimization*. Translations Series in Mathematics and Engineering, Optimization Software Inc, New York, 1987.
- [Pow64] Powell, M.J.D. An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *Computer Journal*, vol. 7, pp. 155-162, 1964.
- [Pra96] Praly. L., Survey on Lyapunov designs of stabilizing state and output feedback. *Lecture notes*, 1996.
- [Pre92] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 1992.

(Q)

- [Que96] Queirioz M. de. and D. Dawson. Nonlinear control of active magnetic bearings: A backstepping approach. *IEEE Transactions on Control Systems Technology*, vol. 4(5), pp. 545-552, 1996.

(R)

- [Rad94] Radcliffe N.J. and P.D. Surry., Formal Memetic Algorithms. Technical Report, Edinburgh Parallel Computing Center, 1994.
- [Rag58] Raggazini. J. R and G. F. Franklin, *Sampled-Data Control Systems*, New York McGraw-Hill, 1958.
- [Rei92] Reichert. R.T., Dynamic scheduling of modern robust control autopilot design for missiles. *IEEE Control Syst. Mag.*, pp. 35-42, 1992.
- [Ren94] Renders, J.-M., Bersini, H. Hybridizing Genetic Algorithms with Hill-Climbing Methods for Global Optimization: Two Possible Ways. *In Proceedings of the first IEEE Conf on Evolutionary Programming* (1994).
- [Res98] Resende M.G.C. Greedy Randomized Adaptive Search Procedures (GRASP). *AT & T Labs Research Technical Report: 98.41.1*, December 1998.
- [Roc95] Rochat Y. and Taillard E., Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of heuristics*, vol 1, pp. 147-167, 1995.
- [Rod00a] Rodriguez. H., R. Ortega, and H. Siguerdidjane. Passivity-based control of magnetic levitation systems: theory and experiments. *Mathematical Theory and Network Systems*, 2000.
- [Rod00b] Rodriguez. H., H. Siguerdidjane, and R. Ortega. Experimental comparison of linear and nonlinear controllers for a magnetic suspension. *In Proceedings of the 2000 IEEE International Conference on Control Applications*, pp. 715-719, 2000.
- [Roc70] Rockafellar. R. T., *Convex Analysis*. Princeton Mathematics Ser. 28. Princeton University Press, Princeton, New Jersey, 1970.
- [Ros60] Rosenbrock, H. H., An Automatic Method for Finding the Greatest or Least Value of a Function, *Computer Journal*, vol. 3, 175-184, 1960.
- [Rot03] Rotella. F., Observation, *Notes de cours de l'Ecole Nationale d'Ingénieurs de Tarbes*, 2003. <http://www.clubeea.org/documents/mediatheque/observateurs.pdf>.

- [Run96] Rundell. A., S. Drakunov, and R. DeCarlo. A sliding mode observer and controller for stabilization of rotational motion of a vertical shaft magnetic bearing. *IEEE Transactions on Control Systems Technology*, vol. 4(5), pp.598-608, 1996.
- (S)
- [Saf80] Safanov M. G., *Stability and Robustness of Multivariable Feedback Systems*, MIT Press, Cambridge, 1980.
- [Sch92] van der Schaft A. J.,  $L_2$ -gain Analysis of Nonlinear Systems and Nonlinear State Feedback  $H_\infty$  Control. *IEEE Transactions on Automatic Control*, vol. 37(6), pp.770–784, 1992.
- [Sch00] van der Schaft A. J.,  *$L_2$ -Gain and Passivity Techniques in Nonlinear Control*, Second Edition, Springer, 2000.
- [Sch95] Scherer. C. W., Multiobjective  $H_2/H_\infty$  Control, *IEEE Transactions on Automatic Control*, vol. 40, pp. 1054-1062, 1995.
- [Sch91] Schnabel R. B. and Eskow E., A new Modified Cholesky Factorization, *SIAM Journal on Scientific and Statistical Computing*, vol. 11:1136-1158, 1991.
- [Sch80] Schumacher J. M., Compensator Synthesis Using (c, a, b)-Pairs. *IEEE Transactions on Automatic Control*, vol. 25(6), pp. 1133-1138, 1980.
- [Sco03] Scorletti. G, V. Fromion et S. Font. Automatique fréquentielle : des critères graphiques à l'optimisation LMI. *Journal Européen des Systèmes Automatisés*, vol. 37/2. 2003.
- [Sco97] Scorletti. G. Approche unifiée de l'analyse et de la commande des systèmes par formulation LMI, Thèse de Doctorat en Automatique, Université Paris-Sud et Supélec, 1997.
- [Sep97] Sepulchre. R., M. Janković and P. Kokotović, *Constructive Nonlinear Control*, Springer Verlag, 1997.
- [Ser02] Serrarens. A. F. A.,  $H_\infty$  Control as applied to torsional drillstring dynamics. Master's thesis, Eindhoven University of Technology, 2002.
- [Ser05] Serban. R, and A. C. Hindmarsh, *CVODES: the Sensitivity-Enabled ODE Solver in SUNDIALS*, Proceedings of IDETC/CIE 2005, Long Beach, Sept. 2005, CA. Also available as LLNL technical report UCRL-JP-200039.
- [Ser06] Serban. R, *SUNDIALS<sup>TB</sup> v2.2.0, a MATLAB Interface to SUNDIALS*, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, *UCRL-SM-212121, 2005*. <http://www.llnl.gov/CASC/sundials/documentation/documentation.html>.
- [Sho85] Shor. N. Z., *Minimization Methods for Non-Differentiable Functions*, vol. 3, Springer-Verlag, 1985.
- [Sha93] Shamma J.S. and J.R. Cloutier. Gain scheduled missile autopilot design using Linear Parameter Varying transformation. *AIAA J. Guidance, Control and Dynamics*, vol. 16(2), pp.256-263, 1993.
- [Sia03] Siarry. P., *Métaheuristiques pour l'Optimisation Difficile*, Edition Eyrolles. 2003.

- [Smi95] Smit. A. T, *Using of optimal control techniques to dampen torsional drillstrings vibrations*. Master's thesis, University of Twente, March 1995.
- [Son89] Sontag. E. D., A universal construction of Artestin's theorem on nonlinear stabilization, *System and Control Letters*, vol. 13, pp.117-123, 1989.
- [Spe80] Speelpening. B., *Compiling Fast Partial Derivatives of Functions Given by Algorithms*, PhD thesis, Department of Computer Science, University of Illinois at Urbana Champaign.
- [Spe62] Spendley, W., G.R. Himsworth, F.R. Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. *Technometrics*, 4, 441-461 (1962).
- [Squ98] Squire, W., and G. Trapp, Using Complex Variables to Estimate Derivatives of Real Functions, *SIAM Review*, Vol. 10, No. 1, pp. 100-112, March 1998.
- [Sun07] Sundials: <http://www.llnl.gov/CASC/sundials/main.html>.
- [Swa90] Swann M. and W. Michaud. Active magnetic bearing performance standard specification. *Proceedings of the 2nd Int. Symp. on Magnetic Bearings*, 1990.

(T)

- [Tal91] Talagrand. O., The use of adjoint equations in numerical modelling of atmospheric circulation, *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. A. Griewank and G. Corliss, Eds, SIAM, Philadelphia, pp. 169-180. 1991.
- [Tan98] Tan. K. C, K. K. Tan. Y. Ci., Robust Controller Design for Linear Systems via Evolutionary Computation, *Proceeding of The 5<sup>th</sup> Conference on Control, Automation, Robotics and Vision*, pp. 434-439, Singapore, 1998.
- [Tar99] Real-Time Windows Target. *The math works inc.* Edition, 1999.
- [The04] Thevenet. J.-B., D. Noll, and P. Apkarian, Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design, in *Proceedings ICINCO*, Portugal, 2004.
- [Tor89a] Törn, A.A. and A. Zilinskas., *Global Optimization*. Springer-Verlag, Berlin, 1989.
- [Tor89b] Torçzon. V. J., Multi-directional search: A direct search algorithm for parallel machines, Ph.D. Thesis, Rice University, Houston, Texas, 1989.
- [Tor91] Torçzon, V. On the convergence of the multidirectional search Algorithms. *SIAM Journal on Optimization*, 1(1), 123-145 (1991).
- [Tor97] Torçzon, V. On the Convergence of Pattern Search Algorithms. *SIAM Journal on Optimization*, 7(1), 1-25 (1997).
- [Tor98] Torres. M. and R. Ortega. *Feedback linearization, integrator backstepping and passivity-based controller design : A comparison example. Chapitre du livre de D. Normand-Cyrot. Perspectives in Control : Theory and Application*. Springer-Verlag, London, 1998.

- [Tsa97] Tsang. E and C. Voudouris., Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Operational Research Letter*, vol. 20(3), pp. 119-127, 1997.
- [Tsi89] Tsiniias. J., Sufficient Lyapunov-like conditions for stabilization, *Math. Contr. Signal. Syst.*, vol. 2, pp.343-357, 1989.
- [Tsu89] Tsuda. M., Y. Nakamura, and T. Higuchi. Design and control of magnetic servo levitation. *Proceedings of the 4th Int. Conf. Ind. Robots*, 1989.
- [Tuc99] Tucker. R. W. and C. Wang. On the effective control of torsional vibrations in drilling systems. *Journal of Sound and Vibration*, vol. 224(1), pp.101–122, 1999.
- [Tuw02] Tu. W, W. Mayne., Studies of Multi-start Clustering for Global Optimization. *International Journal for Numerical Methods in Engineering*, vol. 53, pp. 2239-2252, 2002.

(V)

- [Vid85] Vidyasagar. M., *Control System Synthesis: A Factorization Approach*. MIT Press, 1985.
- [Vou95] Voudouris C. and Tsang E. Function Optimization using Guided Local Search. Technical Report CSM-249, Department of Computer Science, University of Essex, 1995.

(W)

- [Wal94] Walter E. and L. Pronzato., *Identification de Modèles Paramétriques à partir de Données Expérimentales*, Edition Masson, Paris, 1994.
- [Wei94] Weidemann. B. and W. Xiao. A nonlinear fuzzy controller for magnetic bearings without premagnetization. *Proceedings of the 4<sup>th</sup> Int. Symp. on Magnetic Bearings*, 1994.
- [Wil69] Wilkie. D. F. and W. R. Perkins., Essential Parameters in sensitivity analysis, *Automatica*, vol. 5, pp. 191-198, 1969.
- [Wol95] Wolkowicz H. and Zhao Q. An All Inclusive Efficient Trust Region of Updates for Least Change Secant Methods. *SIAM Journal of Optimization*, vol. 5, No.1, 172-191, 1995.
- [Wol97] Wolpert. D. H. and W. G. Macready., The No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, vol. 1(1), pp. 67-82, 1997.
- [Wri96] Wright, M.H. Direct Search Methods: One Scorned, Now Respectable. *Dundee Biannual Conference in Numerical Analysis*, Harlow, UK, pp. 191-208 (1996).

(Y)

- [You76] Youla. D. C, H. A. Jabr, J. J. Bongiorno., Modern Wiener Hopf Design of Optimal Controller, Part II: The multivariable case., *IEEE Transactions on Automatic Control*, vol. 21, pp.319-338, 1976.
- [Ypm95] Ypma. T. J., Historical development of the Newton-Raphson method. *SIAM Review*, vol. 37, pp. 531-551, 1995.

(Z)

- [Zam66] Zames G., On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems (Part I, II), *IEEE Transactions on Automatic Control*, vol. 11, 1966.
- [Zam81] Zames G., Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses, *IEEE Transactions on Automatic Control*, vol. AC-26, no 2, pp. 301-320, 1981.
- [Zam83] Zames G. and B. A. Francis., Feedback, minimax Sensitivity and Optimal Robustness, *IEEE Transactions on Automatic Control*, vol 28, pp. 585-601, 1984.
- [Zho95] Zhou K., Doyle J., Glover K., *Robust and Optimal Control*, Prentice Hall, New Jersey, 1995.
- [Zin99] Zinober. A. S. I. and M. Rios-Bolivar, Symbolic Algebra Toolbox for the Design of Dynamical Adaptive Backstepping Controllers, *IEE Colloquium on Symbolic Computation for Control*, 1999.

