



HAL
open science

Optimisation des correcteurs par les métaheuristiques – Application à la stabilisation inertielle de ligne de visée

Philippe Feyel

► **To cite this version:**

Philippe Feyel. Optimisation des correcteurs par les métaheuristiques – Application à la stabilisation inertielle de ligne de visée. Automatique / Robotique. CentraleSupélec, 2015. Français. NNT : . tel-01260457v1

HAL Id: tel-01260457

<https://centralesupelec.hal.science/tel-01260457v1>

Submitted on 22 Jan 2016 (v1), last revised 27 Jul 2016 (v2)

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.



CentraleSupélec



N° d'ordre : 2015-11-TH

CentraleSupélec

ECOLE DOCTORALE STITS

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

Laboratoire des Signaux et Systèmes

THÈSE DE DOCTORAT

DOMAINE : STIC

Spécialité : Automatique

Soutenue le 16 juin 2015

par :

Philippe FEYEL

Optimisation des correcteurs par les métaheuristiques – Application à la stabilisation inertielle de ligne de visée

Composition du jury :

Directeur de thèse :	Gilles Duc	Professeur, CentraleSupélec
Co-encadrant :	Guillaume SANDOU	Professeur, CentraleSupélec
Président du jury :	Michel ZASADZINSKI	Professeur, Université de Lorraine
Rapporteurs :	Pierre APKARIAN	Ingénieur, HDR, ONERA
	Edouard LAROCHE	Professeur, Université de Strasbourg
Examineurs :	Philippe CHEVREL	Professeur, Ecole des Mines de Nantes
	Yohan COMORASSAMY	Ingénieur, Sagem Défense Sécurité
	Hugues MOUNIER	Professeur, Université Paris-Sud

Table des matières

Table des matières	3
Introduction	5
Partie 1 Métaheuristiques pour l'optimisation	13
1.1. Introduction	13
1.2. Algorithme des essaims particuliers (PSO)	14
1.3. Algorithme des essaims particuliers quantiques (QPSO)	19
1.4. Algorithme de l'évolution différentielle (DE)	24
1.5. Algorithme des colonies d'abeilles (ABC)	28
1.6. Algorithme de la recherche Coucou (CS)	33
1.7. Bilan	38
Partie 2 Reformulation des problèmes de commande robuste pour l'optimisation stochastique.	39
2.1. Introduction	39
2.2. Synthèse H_∞ structurée	39
2.2.1. Introduction	39
2.2.2. La limite des inégalités matricielles affines (LMI)	42
2.2.3. Synthèse H_∞ par l'optimisation non lisse	46
2.2.4. Formulation de la synthèse H_∞ pour l'optimisation stochastique	54
2.3. μ -synthèse	86
2.3.1. Le problème de la robustesse en performance	86
2.3.2. Principe de la μ -synthèse	90
2.3.3. μ -synthèse d'ordre plein	91
2.3.4. μ -synthèse structurée	107
Partie 3 Réglage optimal des correcteurs H_∞ structurés vis-à-vis d'exigences de haut-niveau.	117
3.1. Introduction	117
3.1.1. Problématique	117
3.2. La synthèse H_∞ par loop-shaping	125
3.2.1. Principe de l'approche	125
3.2.2. Marge de gain et marge de phase	127
3.2.3. Interprétation 4-blocs de la méthode	128
3.2.4. Mise en œuvre pratique	129
3.2.5. Implémentation des correcteurs	133
3.3. Une méthode générique pour la déclinaison des exigences	136
3.3.1. Principe général	136
3.3.2. Gestion du niveau de priorité sur les exigences	139
3.4. Réglage optimal des pondérations	140
3.4.1. Optimisation sur modèle nominal	140
3.4.2. Optimisation multi-modèles	144
3.4.3. Exemple applicatif – la stabilisation inertielle de ligne de visée	148
3.5. Réglage optimal et simultané des pondérations et du correcteur final à ordre et structure fixés ...	179
3.5.1. Introduction	179
3.5.2. Vers une élimination des pondérations	181
3.5.3. Extension aux correcteurs à 2 degrés de liberté	200
3.5.4. Extension à la μ -synthèse	214
Conclusions et Perspectives	217
Annexe A Une mise sous forme LFT d'un système électromécanique avec modes souples incertains ...	219
A.1. Modèle général et objectif	219

A.2. Mise sous forme d'état du modèle nominal	221
A.2.1. Mise sous forme d'état des blocs élémentaires	221
A.2.2. Forme d'état globale	224
A.3. Mise sous forme LFT des incertitudes	224
A.3.1. Mise sous forme LFT des blocs élémentaires	224
A.3.2. Mise sous forme LFT globale	233
Annexe B Calcul d'une FTM (1D) à partir d'un signal temporel	237
B.1. Principe du calcul	237
Annexe C Choix du nombre d'itérations pour les tests de Compleib	239
Bibliographie	243

Introduction

Depuis de nombreuses années, Sagem est présent sur le marché des dispositifs de visée. Avec des dispositifs optroniques, on cherche à produire une image dont l'orientation et la qualité ne dépendent pas du porteur. Pour cela, des senseurs optroniques sont portés par un dispositif mécatronique asservi inertiellement ce qui confère au viseur certaines fonctionnalités à savoir l'observation, la détection et l'identification. Citons par exemple les viseurs de chars, d'hélicoptères, les périscopes, les autodirecteurs de missile, etc. (Figure I-1)



Figure I-1: Exemples de viseurs (de haut en bas : Strix, BGS 410, Autodirecteur Mica) (source : <http://www.safran-group.com/site-safran/defense/>)

Du fait du principe fondamental de la dynamique pour un solide en rotation, la vitesse absolue Ω_a de la ligne de visée d'inertie J est régie par :

$$J \frac{d\Omega_a}{dt} = C_{util} - C_{ext} \quad (I-1)$$

Ainsi la stabilisation inertielle consiste en un problème de rejet de perturbations : avec un asservissement, on produit un couple utile C_{util} (en général fourni par une motorisation électrique) devant s'opposer aux perturbations externes C_{ext} s'appliquant sur la charge à chaque instant. Plusieurs solutions d'architecture décrites exhaustivement dans [MAS 08] et [HIL 08] sont alors envisagées.

Pour satisfaire à ses fonctionnalités, le viseur doit disposer de la portée suffisante. Cette portée est directement liée à la performance de stabilisation du viseur notamment vis-à-vis d'environnements mécaniques et de conditions d'emploi plus ou moins sévères et doit être compatible avec le dimensionnement des capteurs optroniques ; ces derniers sont caractérisés par un temps d'intégration (T_i) (temps pendant lequel l'image est acquise et générée électroniquement) et par la taille de leurs détecteurs pixéliques (IFOV¹).

Ainsi pour avoir une portée suffisante, il faut limiter la performance angulaire absolue de la ligne de visée $\theta_a = \int \Omega_a$ à une valeur compatible avec l'IFOV et ce, sur le temps d'intégration T_i , en réponse à toutes les perturbations.

Le problème de stabilisation inertielle consiste à rejeter principalement deux types de perturbations externes :

- la première est le couple de frottements Γ_f engendré lors d'un mouvement relatif entre le porteur du viseur et la ligne de visée. Cette perturbation apparaît lorsque l'attitude du porteur du viseur évolue en basses fréquences. Afin d'éviter le bougé d'image lors de l'observation (et ainsi assurer un confort visuel), il convient de rendre le déplacement angulaire dynamique de la ligne de visée en réponse à Γ_f compatible avec l'IFOV. De même, l'erreur statique occasionnée doit être nulle. Plus particulièrement la dynamique temporelle du rejet du frottement doit être compatible avec la dynamique maximale du régime oscillant du porteur qui n'est que de quelques Hz, une dizaine tout au plus. L'erreur en réponse aux frottements doit donc être nulle (ou tout du moins très faible) au bout d'un temps T_f . Pour valider les performances vis-à-vis de Γ_f , on emploie en général une table oscillante simulant les mouvements du porteur en basses fréquences.
- la seconde perturbation Γ_v provient des déformations mécaniques de la structure du viseur qui de par sa flexibilité transmet à la ligne de visée, voire amplifie, une partie de l'environnement mécanique auquel le viseur est soumis. Afin de permettre au viseur d'assurer ses fonctionnalités d'observation, il convient de limiter le flou généré par cette perturbation haute fréquence² à une valeur compatible avec l'IFOV pendant le temps d'intégration T_i . Pour valider les performances vis-à-vis de Γ_v , on emploie en général un pot vibrant simulant les vibrations hautes fréquences du porteur.

En plus de ces exigences de performances, d'autres exigences doivent être prises en considération lors de la conception :

- Parce que le viseur est un équipement embarqué et/ou pour préserver l'intégrité de la motorisation, une exigence sur la consommation énergétique de l'étage de

¹ *Instantaneous Field of View*

² Haute fréquence par rapport à la première perturbation qui est beaucoup plus basse fréquence.

stabilisation est nécessaire (contrainte sur la puissance instantanée maximale, contrainte sur le courant ou la tension maximale, etc...),

- Parce que le viseur est un système de grande série, la conception de la loi de commande doit se faire sous contrainte de robustesse (en stabilité, en performances,...) vis-à-vis des incertitudes

Ainsi l'automaticien doit concevoir une loi de commande unique qu'il validera sur un prototype unique, ayant un degré de robustesse suffisant pour satisfaire un cahier des charges complexe sur un grand nombre de systèmes.

La méthodologie générale de développement des correcteurs d'asservissement sur un viseur prototype peut être résumée sur la Figure I-2. Cette méthodologie est sous-optimale et chronophage, donc couteuse car :

- La loi de commande finale, qui est obtenue à l'étape 7 au travers de simplifications inévitables (linéarisation, etc) et d'un processus expérimental itératif (étapes 5 et 6) couteux, doit satisfaire une spécification système haut-niveau (étape 1) souvent complexe. Ces simplifications sont nécessaires pour l'emploi de techniques de commandes (étape 4), qui de fait nécessitent un certain degré d'expérience de la part de l'ingénieur.
- Basées sur la forme standard, les techniques modernes de type H_∞ employées à l'étape 4 permettent de déterminer un correcteur (potentiellement multi-variable) qui est optimal (au sens de la norme H_∞) vis-à-vis de la tenue de gabarits linéaires (étape 3) sur les transferts en boucle fermée ou sur la boucle ouverte. Ces gabarits sont matérialisés par le réglage de pondérations fréquentielles sur les signaux à surveiller et sur les entrées exogènes (approche standard), ou directement sur les entrées et sorties du système (approche loop-shaping). La tâche la plus complexe de l'étape 4 consiste non pas à calculer le correcteur lui-même, mais à décliner les spécifications simplifiées en un choix judicieux de pondérations : ce réglage est en soit un processus itératif nécessitant un fort degré d'expérience de la part de l'ingénieur et ce, d'autant plus qu'il doit se poser un certain nombre de questions, par exemple :
 - Pour une problématique SISO³, comment choisir la structure des pondérations à régler ? nombre de pôles/zéros et ordre ?
 - Dans le cas MIMO⁴ où on rajoute en général autant de pondérations que de mesures et de commandes disponibles, le nombre de transferts à « gabariser » devient vite très grand de sorte qu'on procède en général à un réglage séquentiel et découplé des pondérations des différents canaux en supposant les pondérations matricielles diagonales : qu'en est-il de l'optimalité du réglage sous ces hypothèses structurelles simplificatrices ? Quelle est la structure la plus adaptée ? diagonale ? pleine ?
 - Quel est le meilleur compromis performance /robustesse atteignable ?
 - Notons que certaines techniques de synthèse robuste conceptuellement très puissantes telles que la μ -synthèse deviennent sous-optimales de par leur méthode de résolution, ce qui peut naturellement être un frein quant à leur emploi.

³ Single Input, Single Output

⁴ Multiple Inputs, Multiple Outputs

- Enfin les méthodes de synthèse modernes générant un correcteur d'ordre souvent élevé, on procède en général à une post-réduction de ce dernier avant son implémentation numérique. Cette post-réduction peut dégrader la performance et la robustesse si bien qu'il faut prendre en compte a priori la contrainte d'ordre. Par ailleurs se pose aussi la question de savoir quelle est la meilleure structure de correction pour le problème posé ? Structure multi-boucle cascadée ? Correction multi-variable en feedback ? Correction décentralisée ?

Nous avons listé là quelques exemples d'interrogations que l'ingénieur est amené à se poser. Cette liste n'est pas exhaustive. La réponse adéquate à ces questionnements dépend du degré d'expérience de l'ingénieur et de son savoir-faire, et la plupart du temps, il procède à des simplifications pour avancer dans le processus de synthèse, simplifications rendant la solution finale couteuse et sous-optimale vis-à-vis de l'objectif premier qu'est la spécification système haut niveau (étape 1).

Ainsi l'enjeu de cette thèse est de rendre la méthodologie de synthèse des correcteurs des asservissements des viseurs plus adaptée car plus directe et donc moins couteuse en temps de développement en calculant un correcteur final (structuré) par une attaque directe de la spécification système haut niveau.

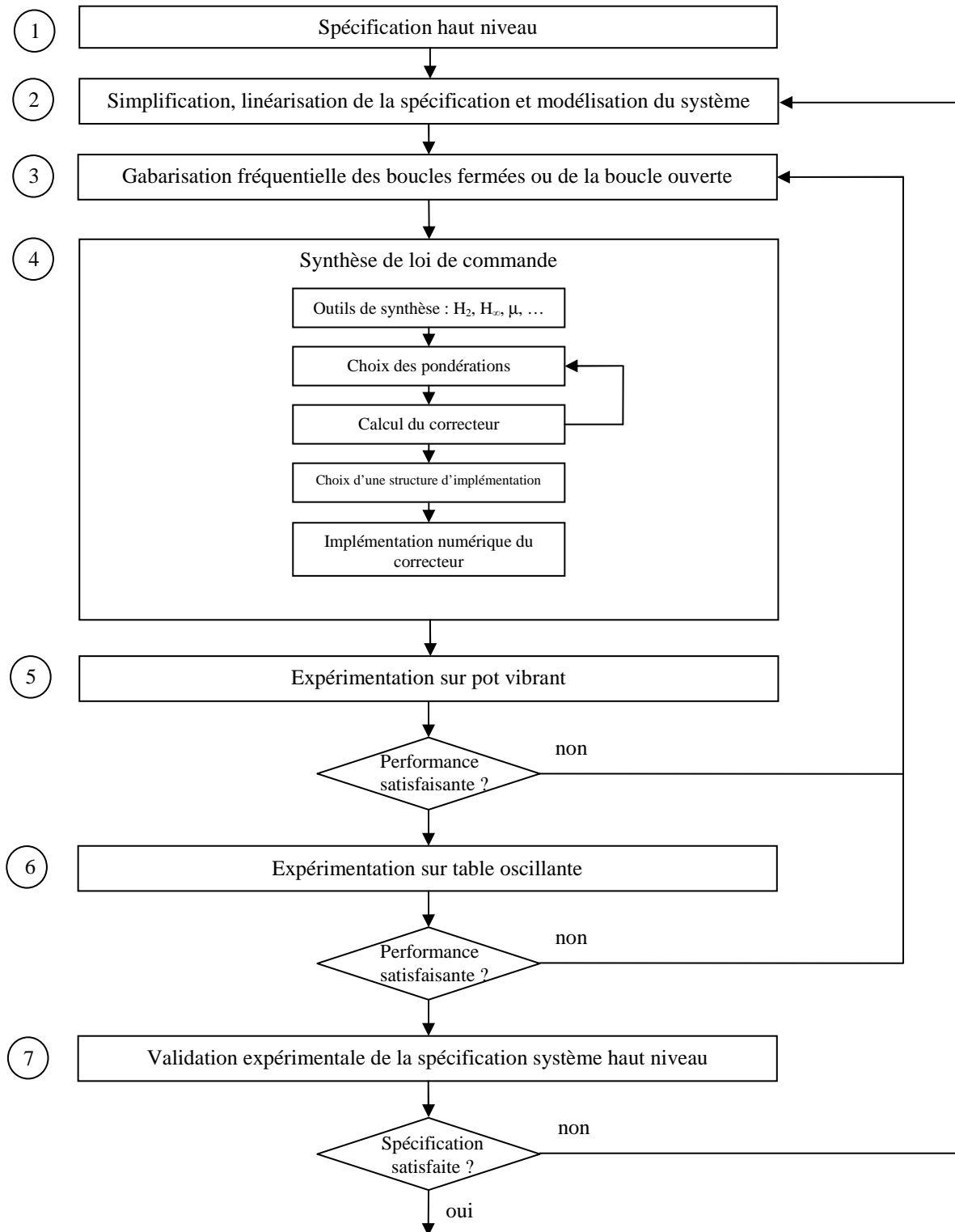


Figure I-2: Méthodologie du développement des correcteurs des asservissements des viseurs

L'optimisation joue toujours un grand rôle pour résoudre les problèmes d'automatique, qui consistent la plupart du temps à calculer un correcteur stabilisant satisfaisant un cahier des charges. Le problème à résoudre peut souvent être énoncé sous la forme générale d'un problème d'optimisation dans lequel on définit une fonction objectif ou fonction de coût (voire plusieurs) $f(x)$ que l'on cherche à minimiser par rapport à tous les paramètres concernés x . La définition d'un tel problème est souvent complétée par la donnée de contraintes portant sur x . Les variables de décision (les paramètres à déterminer) sont alors les variables qui définissent le correcteur à chercher (par exemple les matrices de sa représentation d'état, les coefficients de sa fonction de transfert).

[SCH 97] montre que de nombreux critères de l'automatique s'expriment mathématiquement de manière simple sous la forme d'inégalités matricielles affines (LMI) ; la fonction de coût et les contraintes sont alors des fonctions affines de x . Dans ce cas, le calcul du correcteur optimal est alors réalisé numériquement par des solveurs performants d'optimisation sous contraintes LMI. La convexité des problèmes formulés rend possible la détermination de la solution globale, bien que cela s'accompagne d'un temps de calcul conséquent et de possibles problèmes numériques lorsque le nombre de variables de décisions est important (typiquement quelques milliers).

Lorsque les critères sont plus compliqués à exprimer, la mise en œuvre de telles techniques s'avère la plupart du temps infructueuse car la solution, quand elle est trouvée, n'est optimale que localement. L'emploi de techniques de descente nécessite la formulation du gradient ce qui n'est pas toujours possible. D'autres techniques ne nécessitant pas la formulation du gradient existent telles que [NEL 65] mais elles ne convergent souvent que localement.

Depuis quelques années, on assiste à l'émergence de nombreuses métaheuristiques avec une ambition commune : résoudre au mieux les problèmes d'optimisation difficiles [DRE 06]. Pour tenter de cerner le domaine de « l'optimisation difficile », il est nécessaire de faire la distinction entre deux types de problèmes : les problèmes « discrets » et les problèmes à variables continues. Sans préjuger cette fois de la forme de la fonction $f(x)$ à minimiser, un problème discret consiste à imposer à x un ensemble fini de valeurs possibles ; un problème continu consiste à imposer à x d'appartenir à un intervalle de valeurs.

Ainsi deux sortes de problèmes reçoivent dans la littérature l'appellation comme étant « d'optimisation difficile » :

- certains problèmes d'optimisation discrète, pour lesquels on ne connaît pas d'algorithme exact polynomial (c'est-à-dire dont le temps de calcul est proportionnel à une puissance de n , où n est le nombre d'inconnues du problème). C'est le cas en particulier des problèmes dits « NP difficiles », pour lesquels on conjecture ou on sait qu'il n'existe pas d'algorithme dont le temps de résolution soit borné par un polynôme de degré n .
- certains problèmes d'optimisation à variables continues, pour lesquels on ne connaît pas d'algorithme permettant de repérer un optimum global (c'est-à-dire la meilleure solution possible) à coup sûr et en un nombre fini de calculs.

Les métaheuristiques se révèlent être un outil industriel précieux pour résoudre les problèmes d'optimisation posés ici, car ces méthodes ne préjugent pas de la forme de la fonction de coût $f(x)$ à minimiser, ce qui permet de prendre en compte finalement des spécifications posées sous toute forme, notamment les contraintes sur la dispersion du système, sur n'importe quel système, simplifiant de fait la phase de déclinaison ou de

reformulation du cahier des charges dans un cadre mathématique adapté et réduisant donc l'impact de l'expérience du concepteur sur la robustesse finale : la formulation du gradient n'est pas nécessaire, la seule contrainte étant la capacité à pouvoir évaluer la fonction de coût $f(x)$, qui peut avoir une forme quelconque. A la différence des méthodes type *simplex*, elles sont stochastiques ce qui évite l'explosion combinatoire des possibilités pour trouver le minimum global de $f(x)$. [SIA 02] montre clairement l'avantage à tirer de l'optimisation par métaheuristiques qui permet de remédier au problème du piégeage d'un algorithme itératif « classique » dans un minimum local.

Le principe d'un algorithme classique d'amélioration itérative est le suivant ; on part d'une configuration initiale x_0 , choisie au hasard. On essaie alors une modification élémentaire, souvent appelé mouvement, et on compare les valeurs de la fonction objectif $f(x)$ avant et après modification. Si le changement conduit à une diminution de la fonction objectif $f(x_1) < f(x_0)$, alors il est accepté et la configuration x_1 sert de point de départ pour le prochain essai. Dans le cas contraire, on revient à la condition précédente avant de refaire une tentative. Le processus est itéré jusqu'à ce que tout essai rende le résultat moins bon. La Figure I-3 montre que les méthodes classiques ou méthode de descente ou méthode de la plus grande pente ne conduisent souvent qu'à un minimum local x_n , qui constitue la meilleure solution possible compte tenu de la condition initiale x_0 . Même en appliquant plusieurs fois la procédure avec des conditions initiales différentes, on ne peut pas être certain de trouver la configuration optimale x_{opt} .

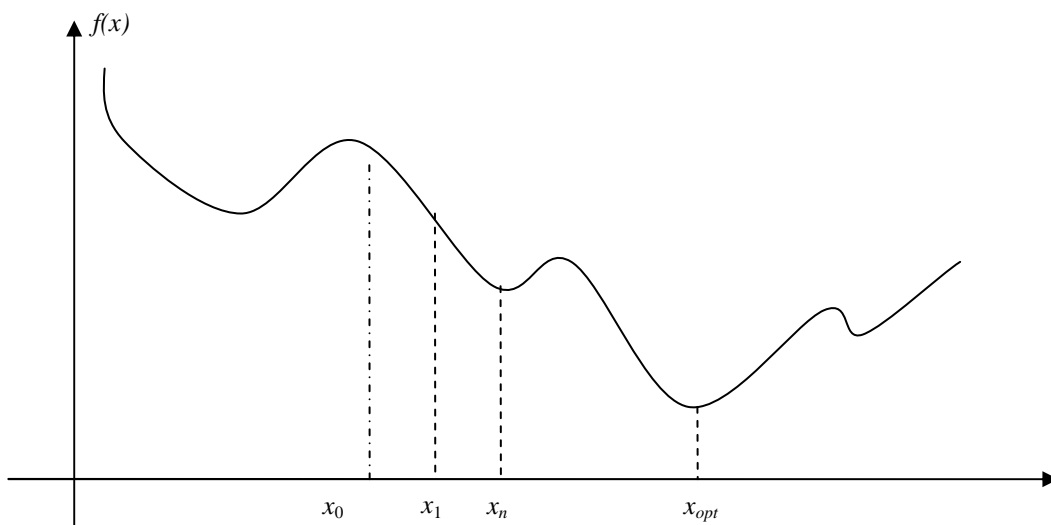


Figure I-3: Une fonction multimodale à minimiser

Pour surmonter l'obstacle des minimums locaux, une autre idée s'est montrée très fructueuse au point qu'elle est à la base de nombreuses métaheuristiques : il s'agit d'autoriser de temps en temps des remontées, c'est-à-dire d'accepter une dégradation temporaire de la situation lors du changement de la configuration courante. C'est le cas si l'on passe de x_n à x_n' (Figure I-4). Un mécanisme de contrôle des dégradations, spécifique à chaque métaheuristique, permet d'éviter la divergence du procédé. Il devient dès lors possible de s'extraire du piège que représente un minimum local, pour partir explorer une autre « vallée » plus prometteuse.

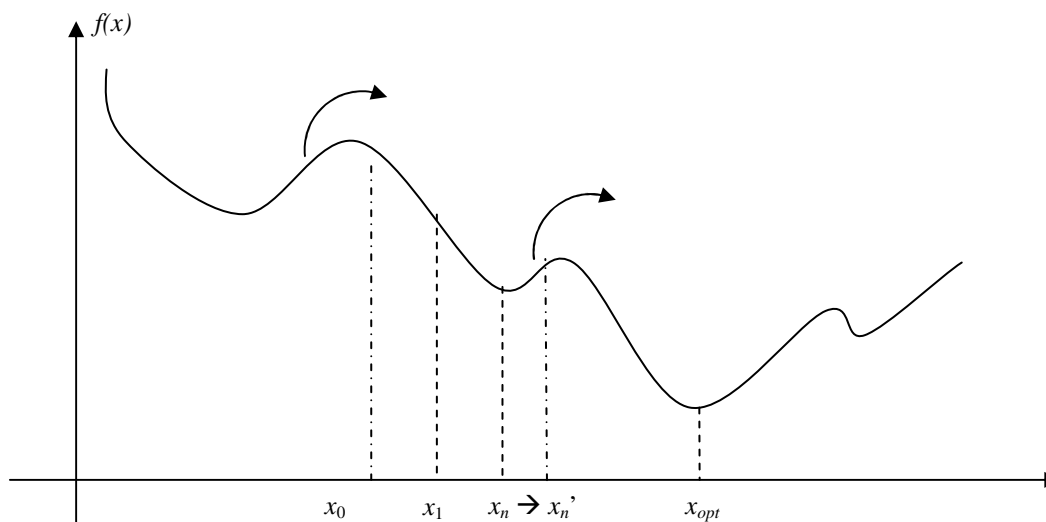


Figure I-4: Extraction des minimums locaux

C'est ce principe que nous souhaitons exploiter pour déterminer un correcteur vis-à-vis d'une spécification système complexe haut-niveau.

A titre d'exemple, [SAN 10] montre une application intéressante et prometteuse de la méthode des essais particuliers pour le réglage automatique des pondérations d'une schématique de synthèse 4-blocs H_∞ . Les travaux montrent notamment comment prendre en compte simultanément des critères temporels et fréquentiels pour le réglage des pondérations : la métaheuristique considérée est alors vue comme une surcouche de la méthode de synthèse employée.

La thèse est organisée de la manière suivante : dans un premier temps on décrit les principales métaheuristiques employées dans la thèse. Puis on reformule, dans une seconde partie, les problèmes de commande moderne pour l'optimisation stochastique : en abordant les problèmes structurés de la synthèse H_∞ et de la μ -synthèse, nous montrons la grande flexibilité qu'offre l'emploi de l'optimisation stochastique pour la prise en compte de contraintes complexes (synthèse multi-modèle, synthèse mixte H_2/H_∞ , contraintes de placement de pôles, etc...). Enfin une troisième partie, basée sur les résultats obtenus, présente une méthode générique pour synthétiser des correcteurs robustes et structurés vis-à-vis de critères de n'importe quelle forme. Après avoir traduit le critère haut niveau sous la forme d'une fonction de coût générique, on propose la résolution de deux problèmes duaux : d'une part l'optimisation des pondérations de synthèse dans le cadre de la synthèse H_∞ par loop-shaping, d'autre part l'optimisation directe du correcteur structuré par la mise en œuvre de scalings statiques en lieu et place des pondérations fréquentielles. L'ensemble de l'exposé est illustré d'exemples industriels basés sur le problème de stabilisation inertielle de ligne de visée.

Partie 1

Métaheuristiques pour l'optimisation

1.1. Introduction

L'optimisation stochastique par les métaheuristiques est adaptée à la résolution de problèmes pour lesquels il est difficile de trouver un optimum global ou de bons optimums locaux par des méthodes plus classiques. Ce type d'optimisation dispose de trois caractéristiques souvent décisives en optimisation globale :

- l'optimisation par les métaheuristiques ne nécessite pas de connaître le gradient de la fonction à minimiser, la seule contrainte étant de pouvoir évaluer cette dernière qui peut donc avoir une forme quelconque ;
- il n'est pas nécessaire de disposer d'un « bon » point initial, l'initialisation étant réalisée de manière aléatoire dans l'espace de recherche ;
- enfin ce type d'optimisation est stochastique ce qui permet de faire face à l'explosion combinatoire des possibilités et ce qui limite le piègeage dans les optima locaux.

On trouvera dans [DRE 06]-[YAN 10a]-[FIS 12] des descriptions des principales métaheuristiques inspirées par des analogies avec la physique, avec l'éthologie, avec la biologie...

Malgré le principal inconvénient qui concerne la pertinence de la solution trouvée, évaluable statistiquement par plusieurs exécutions successives (ce qui occasionne un temps de calcul élevé en général), ce type d'optimisation s'avère intéressant pour les problèmes d'optimisation difficile et paraît adaptée ici à la synthèse de correcteur.

Concernant le temps de calcul, ce dernier peut s'avérer important car la plupart de ces algorithmes procèdent à l'évaluation de la fonction de coût pour chaque individu d'une population à chaque itération. Un moyen de réduire les temps de calculs est de paralléliser ces derniers à chaque itération avec des outils numériques dédiés [MAT 10] – [LUC 09] ; on peut alors répartir la charge de calcul sur plusieurs processeurs en parallèle en affectant à chaque processeur un groupe d'individus de la population.

On présente dans ce chapitre les métaheuristiques retenues pour la suite du document car elles ont donnée de « bons résultats » pour l'ensemble des problèmes d'automatique envisagés dans ce document. On entend par « bon résultat » le fait que l'optimisation ait un bon degré de reproductibilité, un bon temps de convergence et fournisse un bon optimum.

D'autres algorithmes ont été testés mais sans véritable succès (soit parce qu'ils n'ont pas convergé, soient parce qu'ils sont trop complexes à régler). Citons par exemple :

- L'algorithme des lucioles (FireFly – FF) [YAN 09b],
- L'algorithme de chauves-souris (Bat Algorithm – BA) [YAN 10b],
- L'algorithme électromagnétique (Electromagnetic-like Algorithm - ElA) [BIR 03],
- L'algorithme de la recherche harmonique (Harmony Search – HS) [LEE 05].

Pour introduire ces algorithmes, on considère le problème de trouver X_{opt} tel que $X_{opt} = \underset{X \in \Psi}{\operatorname{argmin}}(f(X))$ où $\Psi \subset \mathbf{R}^n$ est l'espace de recherche pour X et n est la dimension du problème. Dans la suite on pose $\Psi = [X_{\min}; X_{\max}]$. La fonction $rnd(x,y)$ désigne un nombre aléatoire généré par une distribution uniforme dans l'intervalle $[x,y]$.

1.2. Algorithme des essais particuliers (PSO)

Principe

L'algorithme des essais particuliers (Particle Swarm Optimization – PSO) [CLE 06]-[BRA 07]-[KEN 95] est une métaheuristique qui s'inspire du comportement social des nuées d'oiseaux (Figure 1-1) ou des bancs de poissons (Figure 1-2).



Figure 1-1: Nuage d'oiseaux



Figure 1-2: Banc de poissons

N_p particules se déplacent à travers l'espace de recherche avec leur propre vitesse. Elles sont capables de se rappeler où elles ont eu leur meilleure performance. Dans la version *LocalBest* de PSO présentée ici, on définit pour une particule donnée son voisinage (ou particules amies) comme un sous-ensemble de particules avec lesquelles elle peut communiquer. Ainsi à chaque instant, chaque particule connaît les caractéristiques de la meilleure de ses voisines. Les notations suivantes sont employées :

- $X_p^{(k)}$ (resp. $V_p^{(k)}$) : position (resp. vitesse) de la particule p à l'itération k ,
- $B_p^{(k)} = \arg \min (f(X_p^{(k-1)}), f(X_p^{(k)}))$: meilleure position trouvée par la particule p jusqu'à l'itération k ,
- $\Omega(X_p^{(k)}) \subset \{1, 2, \dots, N_p\}$: ensemble de particules définissant le voisinage de la particule p ,
- $G_p^{(k)} = \arg \min_{X \in \{B_i^{(k)}, i \in \Omega(X_p^{(k)})\}} f(X)$: meilleure position trouvée par le voisinage de la particule p jusqu'à l'itération k .

Chaque particule se déplace dans l'espace de recherche suivant la loi de transition suivante :

$$\begin{aligned} V_p^{(k+1)} &= w.V_p^{(k)} + c_1 \otimes (B_p^{(k)} - X_p^{(k)}) + c_2 \otimes (G_p^{(k)} - X_p^{(k)}) \\ X_p^{(k+1)} &= X_p^{(k)} + V_p^{(k+1)} \end{aligned} \quad (1-1)$$

où :

- \otimes est le produit terme à terme,

- w est le coefficient d'inertie,
- $c_1 = rnd(0, \bar{c}_1)$, $c_2 = rnd(0, \bar{c}_2)$ sont les coefficients d'accélération.

Ainsi chaque particule de l'essaim se déplace en fonction de sa vitesse propre, de sa meilleure position obtenue jusqu'alors mais aussi de la position de la meilleure de ses voisins (Figure 1-3).

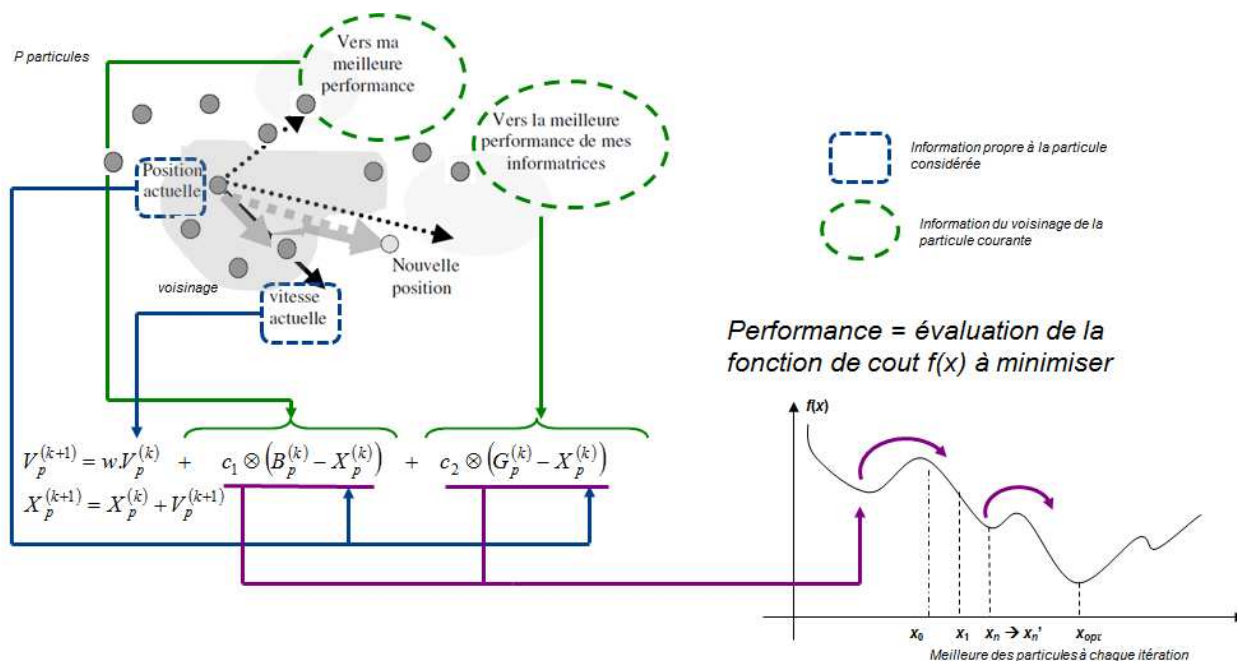


Figure 1-3: Principe de PSO

Dans le cas de la version *GlobalBest* de l'algorithme, $G_p^{(k)}$ est remplacée par $G^{(k)} = \underset{X \in \{B_i^{(k)}, i=\{1,2,\dots,N_p\}\}}{\text{arg min}} f(X)$ qui est la meilleure position que l'essaim a trouvé à l'itération k .

Alors que la version *GlobalBest* favorise la convergence, la version *LocalBest* favorise l'exploration et s'avère plus adaptée aux problèmes de grande dimension : c'est donc avec cette version que nous travaillerons ici.

Confinement des particules

Pour assurer la convergence de l'algorithme, la position et la vitesse des particules doivent être confinées :

$$\begin{aligned} V_p^{(k+1)} &= \min(\max(V_p^{(k+1)}, V_{\min}), V_{\max}) \\ X_p^{(k+1)} &= \min(\max(X_p^{(k+1)}, X_{\min}), X_{\max}) \end{aligned} \tag{1-2}$$

où :

- $[X_{\min}, X_{\max}]$ est l'intervalle de recherche,
- $[V_{\min}, V_{\max}]$ limite la vitesse maximale de déplacement d'une particule et :

$$\begin{aligned} V_{\max} &= \delta(X_{\max} - X_{\min}), \delta \in]0,1] \\ V_{\min} &= -V_{\max} \end{aligned} \quad (1-3)$$

Initialisation de l'algorithme

L'initialisation de l'algorithme consiste à choisir les particules aléatoirement dans l'intervalle de recherche. Il en est de même pour leur vitesse :

$$\begin{aligned} X_p^{(0)} &= X_{\min} + rnd(0,1)(X_{\max} - X_{\min}) \\ V_p^{(0)} &= V_{\min} + rnd(0,1)(V_{\max} - V_{\min}) \end{aligned} \quad (1-4)$$

Pseudo-code de l'algorithme

Le pseudo code de l'algorithme est donné ci-dessous.

Génération aléatoire des positions et des vitesses des particules dans l'espace de recherche à l'aide de (1-4).

Tant que le critère d'arrêt n'est pas satisfait, répéter :

Pour $p = 1$ à N_p particules, faire :

Mettre à jour la vitesse V_p de la particule,

$$V_p = \min(\max(V_p, V_{\min}), V_{\max}),$$

Mettre à jour la position X_p de la particule,

$$X_p = \min(\max(X_p, X_{\min}), X_{\max})$$

Si $f(X_p) < f(B_p)$ alors $B_p = X_p$, fin si

Fin pour

$$\text{Calculer } G_p = \arg \min_{X \in \{B_i, i \in \Omega(X_p)\}} f(X), p = 1 \text{ à } N_p$$

$$\text{Calculer } G = \arg \min_{X \in \{B_i, i = \{1, 2, \dots, N_p\}\}} f(X)$$

Fin Tant que

(1-5)

Réglage canonique de l'algorithme

On définit ci-après la version standard de PSO (*Std_PSO*) avec les réglages suivants (dits canoniques) pour lesquels on obtient en général de bons résultats :

- Taille de l'essaim $N_p = \text{floor}(10 + \sqrt{n})$, où n est le nombre de variables d'optimisation, c'est-à-dire la dimension du problème ;
- $w = 0,729$; $\bar{c}_1 = \bar{c}_2 = 1,494 = 2,05 \times 0,729$;
- $\dim(\Omega(X_p^{(k)})) = 3$;
- $\delta = 0,9$.

Plusieurs topologies existent pour le choix du voisinage $\Omega(X_p^{(k)})$ mais par souci de simplicité, on utilise souvent un voisinage cyclique (*social ring*) défini à l'initialisation une fois pour toutes. Ainsi le voisinage de la particule i est défini par l'ensemble de particules suivant :

$$\Omega(X_i) = (X_{(i-1) \bmod (N_p)} ; X_i ; X_{(i+1) \bmod (N_p)}) \quad (1-6)$$

On trouvera des justifications de ces réglages dans [CLE 02]-[CLE 12]-[CLE 04].

Critères d'arrêt de l'algorithme

Plusieurs critères d'arrêt peuvent être employés. On retiendra :

- Le nombre d'itérations maximum, sachant qu'à chaque itération, la fonction de cout doit être évaluée N_p fois,
- Un nombre maximum d'évaluations de la fonction de coût,
- La convergence prématurée de l'algorithme qui est détectée lorsque toutes les particules tendent à être identiques, c'est-à-dire lorsque la relation suivante est satisfaite :

$$\exists G : \frac{\max_{i=1, \dots, N_p} \|X_i^{(k)} - G\|}{\|X_{max} - X_{min}\|} < \varepsilon \quad (1-7)$$

On choisira dans la plupart des cas $\varepsilon = 10^{-5}$.

Variante de l'algorithme

La métaheuristique précédente constitue une base à partir de laquelle de nombreuses versions ont été développées [SED 09]. L'idée étant de ne pas substituer à un problème de réglage de correcteur un autre problème de réglage de paramètres, il est de bon aloi de ne pas complexifier davantage l'algorithme.

Afin de limiter la convergence prématurée de l'algorithme, on retiendra la version *TVRndIW_PSO*⁵ dans laquelle on fait varier le facteur d'inertie à chaque itération suivant la loi [EBE 01]:

$$w_k = \text{rnd}\left(\frac{1}{2}, 1\right) \quad (1-8)$$

La valeur moyenne du facteur d'inertie reste donc autour de 0,75. En modifiant légèrement le facteur d'inertie à chaque itération, on crée ainsi de la diversité ce qui rend l'algorithme plus efficace pour les problèmes de grande dimension. Les autres paramètres de réglages demeurant inchangés par rapport à la version canonique.

Versions retenues

⁵ TVRndIW = Time Varying Randomly Inertia Weight

Nous retiendrons les deux versions de PSO (version *LocalBest*) qui se sont avérées être les plus performantes dans notre étude : *Std_PSO* et *TVRndIW_PSO*.

1.3. Algorithme des essais particuliers quantiques (QPSO)

Principe

Une propriété importante de PSO [CLE 02] est que la convergence de l'algorithme est atteinte si la $i^{\text{ème}}$ particule converge vers son attracteur local, dont les coordonnées sont définies dans un espace à n dimensions (n étant la dimension du problème) par $P_i^{(k)} = (p_{i1}^{(k)}, p_{i2}^{(k)}, \dots, p_{ij}^{(k)}, \dots, p_{in}^{(k)})$ avec :

$$p_{ij}^{(k)} = \frac{c_1 b_{ij}^{(k)} + c_2 g_j^{(k)}}{c_1 + c_2} = \phi b_{ij}^{(k)} + (1 - \phi) g_j^{(k)} \quad (1-9)$$

ϕ est un nombre aléatoire distribué uniformément dans $[0,1]$ et :

$$\begin{aligned} B_i^{(k)} &= (b_{i1}^{(k)}, b_{i2}^{(k)}, \dots, b_{ij}^{(k)}, \dots, b_{in}^{(k)}) \\ G^{(k)} &= (g_1^{(k)}, g_2^{(k)}, \dots, g_j^{(k)}, \dots, g_n^{(k)}) \end{aligned} \quad (1-10)$$

Dans l'algorithme des essais particuliers quantique (Quantum Particle Swarm Optimization – PSO) [SUN 12a], l'essaim est un système quantique, et chaque particule a un comportement quantique avec son état quantique modélisé par une fonction d'onde $\psi(\vec{X}, t)$ qui doit satisfaire l'équation de Schrödinger :

$$i\hbar \frac{\partial \psi(\vec{X}, t)}{\partial t} = \hat{H}(\vec{X}) \psi(\vec{X}, t) \quad (1-11)$$

où \hbar est la constante de Planck et $i = \sqrt{-1}$. L'opérateur Hamiltonien est défini pour une particule de masse m dans un champ de potentiel $V(\vec{X})$ par :

$$\hat{H}(\vec{X}) = -\frac{\hbar^2}{2m} \nabla^2 + V(\vec{X}) \quad (1-12)$$

Par ailleurs, $Q = |\psi|^2$ représente la fonction de densité de probabilité de la particule d'être en \vec{X} et est telle que :

$$\int Q(\vec{X}) d\vec{X} = 1 \quad (1-13)$$

Dans le cas d'un mouvement dans un espace monodimensionnel et dans le cas où le point d'abscisse x_0 est le centre du puits de potentiel, alors :

$$\begin{aligned}
V(X) &= -\gamma\delta(X - x_0) = -\gamma\delta(Y) \\
\delta(Y) &= 1 \text{ si } Y = 0 \\
\delta(Y) &= 0 \text{ si } Y \neq 0
\end{aligned} \tag{1-14}$$

où γ est l'intensité du puits de potentiel et la fonction $\delta(x)$ vaut 1 quand x vaut 0, et 0 sinon. En appelant E l'énergie de la particule ($E < 0$), alors l'Hamiltonien s'écrit :

$$\hat{H}(Y) = -\frac{\hbar^2}{2m} \frac{d^2}{dY^2} - (\gamma\delta(Y) + E) \tag{1-15}$$

Ainsi à l'équilibre l'état de la particule satisfait l'équation stationnaire de Schrödinger suivante (on retire la dépendance temporelle de (1-11)) :

$$\hat{H}(Y)\psi(Y) = 0 \tag{1-16}$$

C'est-à-dire :

$$\frac{d^2\psi}{dY^2} + \frac{2m}{\hbar^2} (\gamma\delta(Y) + E)\psi(Y) = 0 \tag{1-17}$$

Une solution telle que $\psi \rightarrow 0$ quand $|Y| \rightarrow \infty$ est :

$$\psi(Y) = \frac{1}{\sqrt{L}} e^{-|Y|/L}, \quad L = \frac{2E}{\gamma}, \quad E = \frac{\hbar^2}{2m} \tag{1-18}$$

et :

$$Q(Y) = \frac{1}{L} e^{-2|Y|/L} \tag{1-19}$$

On vérifie que :

$$\begin{aligned}
\int_{-\infty}^{+\infty} \frac{1}{L} e^{-2|Y|/L} dY &= \int_{-\infty}^0 \frac{1}{L} e^{2Y/L} dY + \int_0^{+\infty} \frac{1}{L} e^{-2Y/L} dY \\
&= \left[\frac{1}{2} e^{2Y/L} \right]_{-\infty}^0 + \left[-\frac{1}{2} e^{-2Y/L} \right]_0^{+\infty} \\
&= 1
\end{aligned} \tag{1-20}$$

La distribution de probabilité correspondante s'écrit :

$$F(Y) = 1 - e^{-2|Y|/L} \tag{1-21}$$

La $i^{\text{ème}}$ particule a pour coordonnées $X_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in})$.

Inspiré par l'analyse de la convergence du PSO traditionnel, la $i^{\text{ème}}$ particule se déplace à l'itération k autour d'un puits de potentiel centré sur $p_{ij}^{(k)}$ pour la $j^{\text{ème}}$ dimension (Figure 1-4).

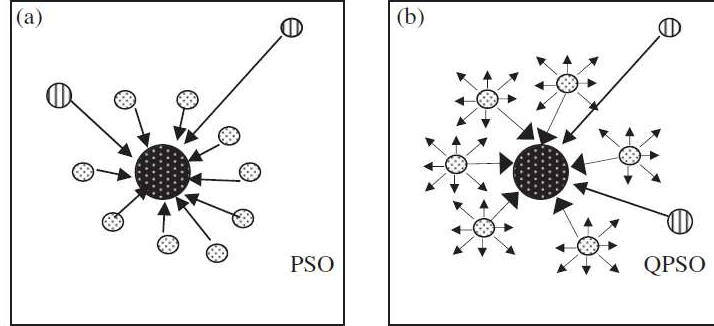


Figure 1-4: Le mouvement des particules dans PSO (a) et QPSO (b)

La fonction d'onde correspondante à l'itération $k+1$ s'écrit :

$$\Psi(x_{ij}^{(k+1)}) = e^{\frac{-|x_{ij}^{(k)} - p_{ij}^{(k)}|}{L_{ij}^{(k)}}} \frac{1}{\sqrt{L_{ij}^{(k)}}} \quad (1-22)$$

La distribution de probabilité correspondante s'écrit :

$$F(x_{ij}^{(k+1)}) = 1 - e^{-2 \frac{|x_{ij}^{(k)} - p_{ij}^{(k)}|}{L_{ij}^{(k)}}} \quad (1-23)$$

D'où l'on tire que :

$$x_{ij}^{(k)} = p_{ij}^{(k)} \pm \frac{L_{ij}^{(k)}}{2} \ln \left(\frac{1}{1 - F(x_{ij}^{(k+1)})} \right) \quad (1-24)$$

Imaginons que l'on fasse des tirages uniformément distribués pour F alors $F \approx U(0,1)$ et donc $1 - F \approx U(0,1)$. En posant $u = 1 - F$, il vient :

$$x_{ij}^{(k)} = p_{ij}^{(k)} \pm \frac{L_{ij}^{(k)}}{2} \ln \left(\frac{1}{u} \right), \quad u = U(0,1) \quad (1-25)$$

Ainsi dans [SUN 12], on propose de remplacer la loi d'évolution du PSO traditionnel (1-1) par :

$$x_{ij}^{(k+1)} = p_{ij}^{(k)} \pm \beta |C_j^{(k)} - x_{ij}^{(k+1)}| \ln\left(\frac{1}{u_{ij}^{(k+1)}}\right) \quad (1-26)$$

Où $C^{(k)} = (C_1^{(k)}, C_2^{(k)}, \dots, C_n^{(k)})$ est la moyenne des meilleures positions trouvées par les particules à l'itération k , c'est-à-dire :

$$C^{(k)} = \left(\frac{1}{P} \sum_{i=1}^P b_{i1}^{(k)}, \frac{1}{P} \sum_{i=1}^P b_{i2}^{(k)}, \dots, \frac{1}{P} \sum_{i=1}^P b_{in}^{(k)} \right) \quad (1-27)$$

$u_{ij}^{(k+1)}$ est un nombre aléatoire distribué uniformément dans $[0,1]$ et β est le coefficient de contraction-expansion (CEC) et est employé pour contrôler la vitesse de convergence de l'algorithme.

Confinement des particules

Pour assurer la convergence de l'algorithme, la position des particules doit être confinée dans l'espace de recherche :

$$X_p^{(k+1)} = \min\left(\max\left(X_p^{(k+1)}, X_{\min}\right), X_{\max}\right) \quad (1-28)$$

Initialisation de l'algorithme

L'initialisation de l'algorithme consiste à choisir les particules aléatoirement dans l'intervalle de recherche :

$$X_p^{(0)} = X_{\min} + \text{rnd}(0,1)(X_{\max} - X_{\min}) \quad (1-29)$$

Pseudo-code de l'algorithme

Le pseudo-code de l'algorithme est donné ci-dessous.

Génération aléatoire des positions des particules dans l'espace de recherche
Tant que le critère d'arrêt n'est pas satisfait, répéter :

Calculer la meilleure position moyenne C à l'aide de (1-27),

Pour $i = 1$ à N_P particules, faire :

si $f(X_i) < f(B_i)$ alors $B_i = X_i$ fin si

Pour $j = 1$ à n dimensions, faire :

$\phi = \text{rnd}(0,1)$; $u = \text{rnd}(0,1)$;

$p_{ij} = \phi \cdot b_{ij} + (1-\phi) \cdot g_j$

si $\text{rnd}(0,1) > 0,5$

$x_{ij} = p_{ij} + \beta \cdot \text{abs}(C_j - x_{ij}) \cdot \log(1/u)$

sinon

(1-30)

```

       $x_{ij} = p_{ij} - \beta \cdot \text{abs}(C_j - x_{ij}) \cdot \log(1/u)$ 
    fin si
       $x_{ij} = \min(\max(x_{ij}, X_{\min\_j}), X_{\max\_j})$ 
    Fin Pour
  Fin Pour
  Identifier  $G = \arg \min_{X \in \{B_i, i=\{1,2,\dots,N_P\}\}} f(X)$ 
  Fin Tant que

```

Réglage de l'algorithme

Un avantage de cet algorithme est son faible nombre de paramètres de réglage. Le seul paramètre à régler est β . L'inconvénient est que le comportement de l'algorithme est très sensible à β . Cependant, en général, une valeur de β constante autour de 0,75 conduit à de bons résultats [SUN 12a]. C'est la version *StaticCEC_QPSO*.

Suivant la même idée que le PSO traditionnel, la taille de l'essaim est fixé à $N_P = \text{floor}(10 + \sqrt{n})$, où n est le nombre de variables d'optimisation, c'est-à-dire la dimension du problème.

Critères d'arrêt de l'algorithme

Plusieurs critères d'arrêt peuvent être employés. On retiendra :

- Le nombre d'itérations maximum, sachant qu'à chaque itération, la fonction de cout doit être évaluée N_P fois,
- Un nombre maximum d'évaluations de la fonction de coût,
- La convergence prématurée de l'algorithme qui est détectée lorsque toutes les particules tendent à être identiques, c'est-à-dire lorsque la relation suivante est satisfaite :

$$\exists G : \frac{\max_{i=1,\dots,N_P} \|X_i^{(k)} - G\|}{\|X_{\max} - X_{\min}\|} < \varepsilon \quad (1-31)$$

On choisira dans la plupart des cas $\varepsilon = 10^{-5}$.

Variante de l'algorithme

De nombreuses versions de QPSO existent [SUN 12a]. La plupart consistent à faire varier β suivant des lois particulières. Par exemple, on peut créer de la diversité à chaque itération en faisant varier aléatoirement β comme suit :

$$\beta(k) = \text{rnd}(\beta_{\min}, \beta_{\max}) ; \beta_{\max} = 1,0 ; \beta_{\min} = 0,5 \quad (1-32)$$

On retiendra la version *TVDdecCEC_QPSO* [SUN 12a] [SUN 12b], dans laquelle β ne fait que décroître lors d'un processus d'optimisation à N itérations suivant la loi suivante :

$$\beta(k) = \beta_{\max} - \frac{\beta_{\max} - \beta_{\min}}{N} \times k ; \beta_{\max} = 1,0 ; \beta_{\min} = 0,5 \quad (1-33)$$

Versions retenues

Au final nous retiendrons les deux versions de QPSO qui se sont avérées être les plus performantes dans notre étude : *StaticCEC_QPSO* et *TVDecCEC_QPSO*.

1.4. Algorithme de l'évolution différentielle (DE)

Principe

L'évolution différentielle (Differential Evolution – DE) [STO 95] est une métaheuristique récente qui appartient à la classe des algorithmes évolutionnaires (Figure 1-5), lesquels s'inspirent de la théorie de l'évolution des espèces (Figure 1-6).

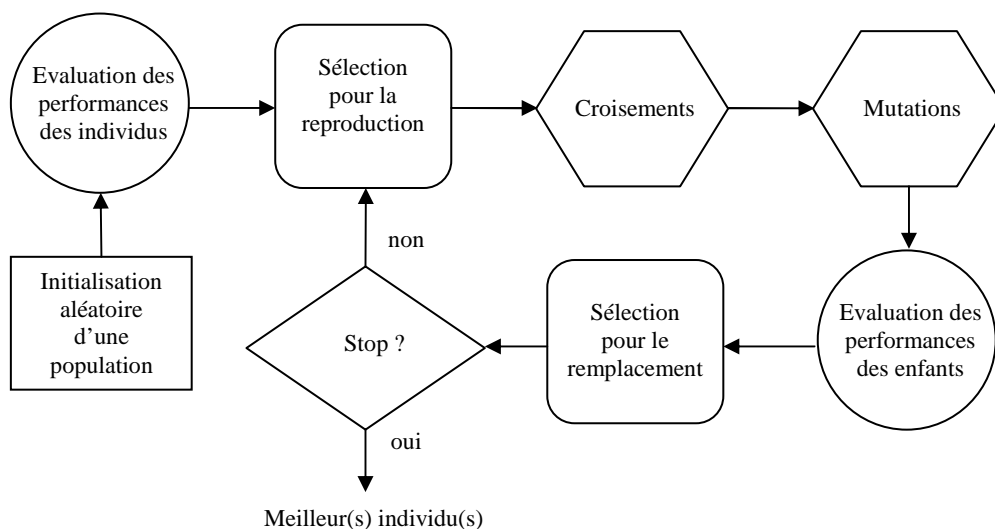


Figure 1-5: Principe des algorithmes évolutionnaires

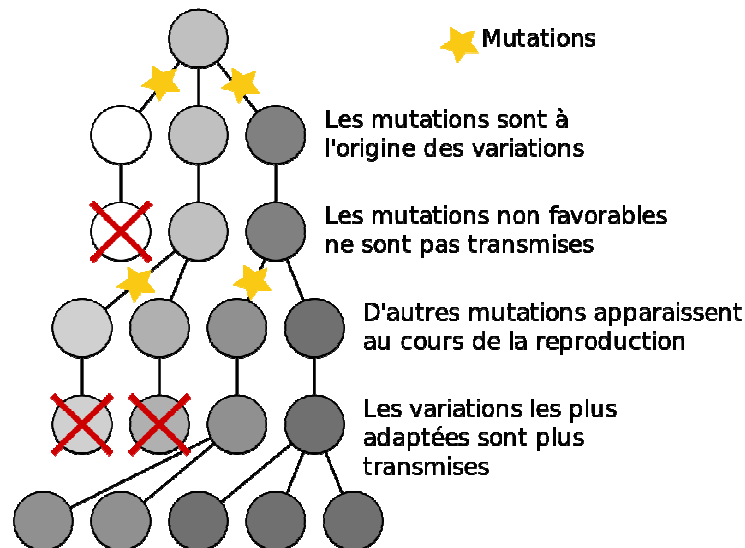


Figure 1-6: Théorie de Darwin

On décrit ci-dessous la version classique de l'algorithme.

Initialisation de l'algorithme

On construit une population initiale P de N individus :

$$P = \{X_1, \dots, X_i, \dots, X_N\} \quad (1-34)$$

Chaque individu est défini par ses n gènes :

$$X_i = (x_{i1}, \dots, x_{ij}, \dots, x_{in}) \quad (1-35)$$

n est donc la dimension du problème.

En supposant que $x_j \in [X_{\min j}, X_{\max j}]$, le $j^{\text{ème}}$ gène du $i^{\text{ème}}$ individu est choisi aléatoirement dans son intervalle de définition :

$$x_{ij}^{(0)} = X_{\min j} + \text{rnd}(0,1)(X_{\max j} - X_{\min j}), \quad j = 1, \dots, n, \quad i = 1, \dots, N \quad (1-36)$$

La $k^{\text{ème}}$ itération consiste à réaliser successivement les 3 étapes décrites ci-dessous.

Description de l'algorithme

- Etape 1 : mutation

Usuellement deux schémas de mutation sont employés : le schéma *rand* et le schéma *best*. Ils sont décrits ci-dessous.

$$\begin{aligned} \text{DE/Rand/1} &\rightarrow U_i^{(k+1)} = X_{a_i}^{(k)} + F(X_{b_i}^{(k)} - X_{c_i}^{(k)}) \\ \text{DE/Best/1} &\rightarrow U_i^{(k+1)} = X_{best}^{(k)} + F(X_{a_i}^{(k)} - X_{b_i}^{(k)}) \end{aligned} \quad (1-37)$$

Où $X_{a_i}, X_{b_i}, X_{c_i}$ sont différents et choisis aléatoirement dans la population et X_{best} est le meilleur individu obtenu depuis le début de l'optimisation. $F \in [0,2]$ est un nombre appelé facteur de mutation. Ainsi à chaque itération, un nombre N de mutants U_i sont générés selon un des schémas précédents (choisi une fois pour toutes au début de l'optimisation).

Le schéma *rand* favorise la diversité alors que le schéma *best* encourage la convergence. Parce que les problèmes auxquels nous nous confrontons ici sont de grande dimension, nous privilégierons ici l'exploration et donc le schéma *rand*.

- Etape 2 : croisement

En croisant $U_i^{(k+1)}$ avec $X_i^{(k)}$ un nouvel individu $V_i^{(k+1)}$ est généré avec des gènes définis de la manière suivante :

$$v_{ij}^{(k+1)} = \begin{cases} u_{ij}^{(k+1)} & \text{si } \text{rnd}(0,1)_{ij} \leq C_r \text{ ou si } j = j_i \\ x_{ij}^{(k)} & \text{sinon} \end{cases} \quad (1-38)$$

où $C_r \in [0,1;0,9]$ est le taux de croisement et n un nombre entier choisi aléatoirement dans $\{1,2,\dots,N\}$.

- Etape 3 : sélection

La population est mise à jour avec des individus définis par :

$$X_i^{(k+1)} = \begin{cases} V_i^{(k+1)} & \text{si } f(V_i^{(k+1)}) \leq f(X_i^{(k)}) \\ X_i^{(k)} & \text{sinon} \end{cases} \quad (1-39)$$

Si le critère d'arrêt est satisfait alors renvoyer la meilleure solution trouvée jusqu'alors, sinon retourner à l'étape 1.

Pseudo-code de l'algorithme

Le pseudo code de l'algorithme est donné ci-dessous.

Génération d'une population initiale.

Tant que le critère d'arrêt n'est pas satisfait, répéter :

Pour $i = 1$ à N individus, faire :

Choisir aléatoirement 3 individus $X_{a_i}, X_{b_i}, X_{c_i}$

Mutation :

Créer un mutant U_i à partir de l'individu X_i et/ou des

(1-40)

$X_{a_i}, X_{b_i}, X_{c_i}$ (en fonction du schéma de mutation choisi au début)

Croisement :

Créer l'individu test V_i à partir du mutant U_i et de X_i

Sélection :

Si $f(U_i) \leq f(X_i)$ alors on remplace $X_i \leftarrow U_i$

Fin pour

Calculer X_{best}

Fin Tant que

Critère d'arrêt de l'algorithme

Plusieurs critères d'arrêt peuvent être employés. On retiendra :

- Le nombre d'itérations maximum, sachant qu'à chaque itération, la fonction de cout doit être évaluée P fois,
- Un nombre maximum d'évaluations de la fonction de coût,
- La convergence prématurée de l'algorithme qui est détectée lorsque toutes les particules tendent à être identiques, c'est-à-dire lorsque la relation suivante est satisfaite :

$$\exists X_{best} : \frac{\max_{i=1, \dots, P} \|X_i^{(k)} - X_{best}\|}{\|X_{max} - X_{min}\|} < \varepsilon \quad (1-41)$$

On choisira dans la plupart des cas $\varepsilon = 10^{-5}$.

Réglage de l'algorithme

Un des avantages de l'algorithme est son faible nombre de paramètres de réglage. Nous emploierons ici les réglages classiques de [STO 96] : $F = 0,75$ et $C_r = 0,8$.

Bien qu'une taille de population N entre $5n$ et $10n$ est généralement conseillé, l'algorithme a donné ses meilleurs résultats en employant la même idée que dans PSO c'est-à-dire un nombre d'individus égal à :

$$N = \text{floor}(10 + \sqrt{n}) \quad (1-42)$$

Variantes de l'algorithme

Dans la suite $X_{a_i}, X_{b_i}, X_{c_i}, X_{d_i}, X_{e_i}$ sont des individus différents et choisis aléatoirement dans la population et X_{best} est le meilleur individu de la population obtenu depuis le début de l'optimisation.

Il existe de nombreuses variantes de DE [MEZ 08] qui consistent la plupart du temps en la définition de schémas de mutation particuliers. Citons par exemple :

- DE/Best/2 :

$$\text{DE/Best/2} \rightarrow U_i^{(k+1)} = X_{best}^{(k)} + F \left(X_{a_i}^{(k)} - X_{b_i}^{(k)} + X_{c_i}^{(k)} - X_{d_i}^{(k)} \right) \quad (1-43)$$

- DE/Rand/2 :

$$\text{DE/Rand/2} \rightarrow U_i^{(k+1)} = X_{a_i}^{(k)} + F \left(X_{b_i}^{(k)} - X_{c_i}^{(k)} + X_{d_i}^{(k)} - X_{e_i}^{(k)} \right) \quad (1-44)$$

- DE/Current-to-Rand/1 :

$$\text{DE/Current - to - rand/1} \rightarrow U_i^{(k+1)} = X_i^{(k)} + F \left(X_{a_i}^{(k)} - X_{b_i}^{(k)} \right) + F' \left(X_{c_i}^{(k)} - X_i^{(k)} \right) \quad (1-45)$$

- DE/Current-to-Best/1 :

$$\text{DE/Current - to - best/1} \rightarrow U_i^{(k+1)} = X_i^{(k)} + F \left(X_{a_i}^{(k)} - X_{b_i}^{(k)} \right) + F' \left(X_{best}^{(k)} - X_i^{(k)} \right) \quad (1-46)$$

- DE/Rand-to-Best/1 :

Le schéma *rand* favorise la diversité (donc l'exploration de l'espace de recherche) alors que le schéma *best* encourage la convergence mais conduit souvent à des solutions sous optimales. C'est pourquoi, dans le but de tirer parti des avantages de ces 2 schémas de base, nous proposons d'employer le schéma de mutation (1-47) que nous dénommons DE/Rand-to-best/1 :

$$\text{DE/Rand - to - best/1} \rightarrow U_i^{(k+1)} = X_{a_i}^{(k)} + F \left(X_{b_i}^{(k)} - X_{c_i}^{(k)} \right) + F' \left(X_{best}^{(k)} - X_{a_i}^{(k)} \right) \quad (1-47)$$

F' est un autre facteur de mutation que nous choisissons égal à $1-F$, soit ici $F' = 0,25$. Cela constitue une bonne balance entre la diversité et la convergence.

Comme nous le verrons plus loin, ce schéma s'avère très performant pour la résolution des problèmes de commande.

Versions retenues

Nous retiendrons les deux versions de DE qui se sont avérées être les plus performantes dans notre étude : ***Rnd_DE*** et ***Rnd2Bst_DE*** qui correspondent respectivement aux deux schémas de mutation DE/Rand1/ et DE/Rand-to-best/1.

1.5. Algorithme des colonies d'abeilles (ABC)

Principe

L'optimisation par colonies d'abeilles artificielles (Artificial Bee Colony – ABC) [KAR 07]-[KAR 08] est une métaheuristique récente qui s'inspire du modèle naturel du comportement des abeilles mellifères lors de la recherche de leur nourriture.

Le processus de recherche de nourriture chez les abeilles est fondé sur un mécanisme de déplacement très efficace. Il leur permet d'attirer l'attention d'autres abeilles de la colonie aux sources alimentaires trouvées dans le but de collecter des ressources diverses. En fait, les abeilles utilisent un ensemble de danses frétilantes comme moyen de communication entre elles. Ces danses permettent aux abeilles de partager des informations sur la direction, la distance et la quantité du nectar avec ses congénères. La collaboration et la connaissance collective des abeilles de la même colonie sont basées sur l'échange d'information sur la quantité du nectar dans la source de nourriture trouvée par les différents membres.

Dans un algorithme d'optimisation par colonies d'abeilles, une source de nectar correspond à une solution possible au problème à traiter. La colonie d'abeilles artificielles est composée de trois types d'abeilles: les ouvrières, les spectatrices et les scouts :

- L'ouvrière exploite la source de nourriture trouvée. Elle se base sur sa mémoire et essaye d'apporter des modifications à sa position (solution) actuelle pour découvrir une nouvelle position (i.e. source de nourriture).
- La spectatrice attend le retour des ouvrières au champ de danse pour observer leurs danses et recueillir des informations sur les sources de nectar qu'elles ont trouvées.
- L'abeille scout exploite l'espace de recherche en lançant une recherche aléatoire d'une nouvelle source de nourriture.

Initialisation de l'algorithme

La population initiale de solution est constituée d'un nombre N_{FS} de sources de nourriture générées aléatoirement dans l'espace de recherche. En supposant que la $i^{\text{ème}}$ source de nourriture de la population soit représentée par $X_i = (x_{i1}, \dots, x_{ij}, \dots, x_{in})$ où n est la dimension du problème, alors chaque source de nourriture est générée par :

$$x_{ij} = X_{\min j} + \text{rnd}(0,1)(X_{\max j} - X_{\min j}), \quad j = 1, \dots, n, \quad i = 1, \dots, N_{FS} \quad (1-48)$$

Ces sources de nourriture sont assignées aléatoirement aux N_O ouvrières de la ruche et les valeurs des fonctions de coût correspondantes sont évaluées. Nous supposons ici que $N_O = N_{FS}$.

De même nous supposons que le nombre d'abeilles spectatrices N_S est égal à N_O .

Enfin chaque source de nourriture dispose d'un compteur d'abandon initialisé à 0.

La $k^{\text{ème}}$ itération consiste à réaliser successivement les 3 phases décrites ci-dessous.

Description de l'algorithme

- Phase ouvrière :

Dans cette phase, chaque abeille employée génère une nouvelle source de nourriture x_{new} choisie au voisinage de la position actuellement trouvée :

$$x_{newij} = x_{ij} + rnd(-1,1)(x_{ij} - x_{kj}), \quad i = 1, \dots, N_O \quad (1-49)$$

Où $k \in \{1, 2, 3, \dots, N_{FS}\}$ tel que $k \neq i$ et $j \in \{1, 2, 3, \dots, n\}$ sont choisis aléatoirement.

On procède ensuite à une sélection après évaluation de la fonction de coût pour cette nouvelle solution. X_{new} remplace X_i dans la population si $f(X_{new}) \leq f(X_i)$ (et son compteur d'abandon est réinitialisé) sinon X_i est conservée et son compteur d'abandon est incrémenté.

- Phase spectatrice :

Dans cette étape les spectatrices récupèrent de la part des ouvrières de l'information sur la quantité de nectar de la source x_i . La probabilité p_i que les spectatrices choisissent la source x_i est déterminée comme suit :

$$p_i = \frac{fit_i}{\sum_{k=1}^{N_{FS}} fit_k} \quad (1-50)$$

Où fit_i est la quantité de nectar de la $i^{\text{ème}}$ source de nourriture X_i .

On trouve aussi dans la littérature l'expression suivante [OZT 12] :

$$p_i = 0,1 + \frac{0,9 fit_i}{\max_k (fit_k)} \quad (1-51)$$

La quantité de nectar associée à la source X_i est déterminée par :

$$fit_i = \begin{cases} \frac{1}{1 + f(X_i)}, & \text{si } f(X_i) \geq 0 \\ 1 + |f(X_i)|, & \text{si } f(X_i) < 0 \end{cases} \quad (1-52)$$

D'après (1-50) et (1-51), il est clair que plus fit_i est élevée, plus la probabilité de sélectionner la source X_i correspondante est importante.

Une fois que la $i^{\text{ème}}$ source est sélectionnée par les spectatrices, elles en réalisent une modification en utilisant (1-49).

Si la source ainsi modifiée est meilleure que X_i alors la source modifiée remplace X_i dans la population (son compteur d'abandon est alors réinitialisé), sinon X_i est conservé et son compteur d'abandon est incrémenté.

- Phase scoute :

La $i^{\text{ème}}$ source de nourriture est abandonnée si elle ne peut être améliorée au bout d'un certain nombre prédéterminé d'essais T_{limit} (c'est-à-dire si son compteur d'abandon dépasse T_{limit}) et l'abeille ouvrière correspondante devient alors une abeille scoute. L'abeille scoute génère alors sa source de nourriture aléatoirement :

$$x_{ij} = X_{\min j} + \text{rnd}(0,1)(X_{\max j} - X_{\min j}), \quad j = 1, \dots, n \quad (1-53)$$

Pseudo-code de l'algorithme

Le pseudo code de l'algorithme est donné ci-dessous.

Pour chaque ouvrière
 Générer une solution initiale X_i pour chaque ouvrière avec (1-48),
 Evaluer la fonction de cout $f(X_i)$
 Evaluer la quantité de nectar (fit_i) associée en utilisant (1-52),
 Initialiser un compteur d'abandon $C_i = 0$.

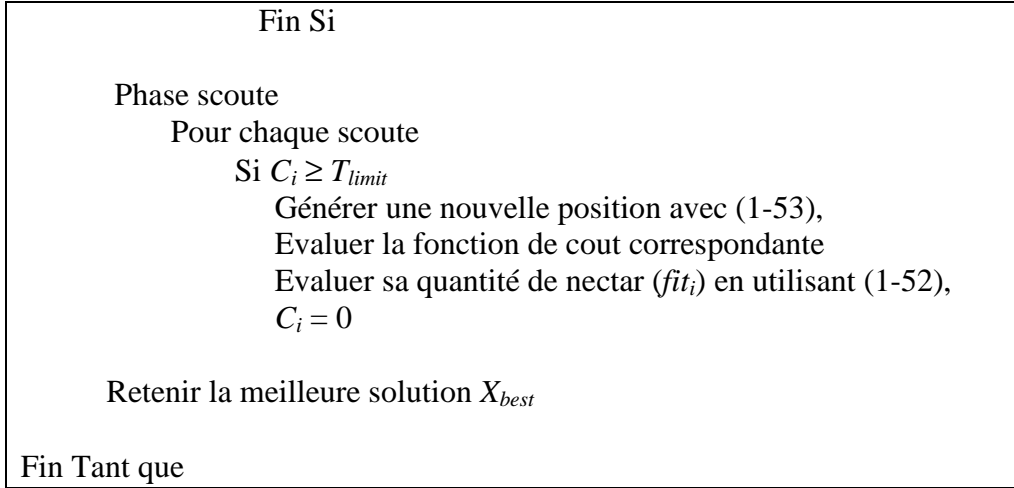
Tant que le critère d'arrêt n'est pas satisfait, répéter :

Phase ouvrière
 Pour chaque ouvrière
 Générer une nouvelle position X_{new} avec (1-49),
 Evaluer la fonction de cout correspondante $f(X_{new})$,
 Evaluer sa quantité de nectar (fit_{new}) en utilisant (1-52),
 Si $f(X_{new}) \leq f(X_i)$, alors
 $X_i \leftarrow X_{new}$
 $C_i = 0$
 Sinon
 $C_i = C_i + 1$
 Fin Si
 Fin Pour

Evaluer la probabilité de transmission d'information associée à chaque source p_i l'aide de (1-50),

Phase spectatrice
 Pour chaque spectatrice
 si $\text{rnd}(0,1) < p_i$
 Générer une nouvelle position X_{new} avec (1-49),
 Evaluer la fonction de cout correspondante $f(X_{new})$,
 Evaluer sa quantité de nectar (fit_{new}) en utilisant (1-52)
 Si $f(X_{new}) \leq f(X_i)$, alors
 $X_i \leftarrow X_{new}$
 $C_i = 0$
 Sinon
 $C_i = C_i + 1$

(1-54)



Critère d'arrêt de l'algorithme

Plusieurs critères d'arrêt peuvent être employés. On retiendra :

- Le nombre d'itérations maximum, sachant qu'à chaque itération, la fonction de cout doit être évaluée P fois,
- Un nombre maximum d'évaluations de la fonction de coût,
- La convergence prématurée de l'algorithme qui est détectée lorsque toutes les particules tendent à être identiques, c'est-à-dire lorsque la relation suivante est satisfaite :

$$\exists X_{best} : \frac{\max_{i=1, \dots, P} \|X_i^{(k)} - X_{best}\|}{\|X_{max} - X_{min}\|} < \varepsilon \quad (1-55)$$

On choisira dans la plupart des cas $\varepsilon = 10^{-5}$.

Variante de l'algorithme

Les paramètres de base de l'algorithme sont le nombre de sources de nourriture N_{FS} , le nombre d'ouvrières N_O , le nombre de spectatrices N_S ainsi que la limite T_{limit} à partir de laquelle la source de nourriture doit être abandonnée.

La version standard de l'algorithme consiste à prendre :

$$N_{FS} = N_O = N_S \quad (1-56)$$

Il ne reste alors que 2 paramètres à régler.

Bien qu'une taille de population N_{FS} de l'ordre $10n$ soit généralement conseillée, l'algorithme a donné ses meilleurs résultats en employant la même idée que dans PSO c'est-à-dire :

$$N_{FS} = \text{floor}(10 + \sqrt{n}) \quad (1-57)$$

Enfin dans la littérature, il est préconisé de choisir [KAR 08] :

$$T_{limit} = n \times N_{FS} \quad (1-58)$$

1.6. Algorithme de la recherche Coucou (CS)

Principe

La recherche coucou (Cuckoo Search - CS) est une métaheuristique récente [YAN 09a] qui s'inspire du mode de reproduction de certaines espèces de coucous. En effet leur stratégie de reproduction a pour particularité le fait que les femelles pondent leurs œufs dans le nid d'autres espèces (dont les œufs ressemblent (Figure 1-7)). Ces œufs peuvent alors être couvés par des parents de substitution. Par ailleurs, quand les œufs de coucou parviennent à éclore dans le nid hôte (ils éclosent plus rapidement), les poussins coucou ont le réflexe d'éjecter les œufs de l'espèce hôte en dehors du nid et imitent même le cri des poussins hôtes dans le but d'être nourris par l'espèce hôte.



Figure 1-7: Un œuf de coucou (gris) dans un nid d'accenteur mouchet

Il peut cependant arriver que les œufs de coucous soient découverts ; dans ce cas, les parents de substitution les sortent du nid, ou abandonnent le nid et commencent leur couvée autre part. Cette métaheuristique est donc basée sur ce comportement parasite des espèces de coucous associé à une logique de déplacement de type « Levy flight » propres à certains oiseaux et certaines espèces de mouches.

L'algorithme CS est basé sur les règles suivantes :

- Chaque coucou pond seulement un œuf à la fois et le place dans un nid choisi aléatoirement,
- Les meilleurs nids avec des œufs (solutions) de haute qualité sont conservés pour les prochaines générations,
- Le nombre de nids hôtes est fixé et l'œuf pondu par un coucou peut être découvert par l'espèce hôte avec une probabilité $p_a \in [0,1]$. Dans ce cas, l'oiseau

hôte soit sort l'œuf du nid, soit quitte le nid et en construit un nouveau. Pour simplifier, cette dernière hypothèse peut être approchée par le remplacement d'une fraction p_a de n nids par des nouveaux.

Dans CS, chaque œuf dans un nid représente une solution et chaque coucou peut pondre un seul œuf (qui représente une solution), le but est d'utiliser la nouvelle et potentiellement meilleure solution pour remplacer une moins bonne solution dans un nid. Bien que l'algorithme puisse être étendu au cas plus complexe où chaque nid contient plusieurs œufs représentant un ensemble de solutions, on emploie ici la version la plus simple où chaque nid ne contient qu'un seul œuf. Dans ce cas, il n'y a plus de distinction entre œuf, nid ou coucou, et chaque nid correspond à un œuf qui représente aussi un coucou.

Levy flight

Dans le cadre de CS, le pas de déplacement des coucous est déterminé par le Levy Flight (Figure 1-8). Le Levy flight est une marche aléatoire dans laquelle les pas ont une longueur ayant une certaine distribution de probabilité (distribution de Levy), la direction des pas étant isotrope et aléatoire. Le Levy Flight est une classe de marche aléatoire dans laquelle les sauts sont distribués selon la distribution de Levy qui consiste en une loi en puissance à variance et moyenne infinies du type :

$$Levy(\beta) \sim y = x^{-\beta}, 1 < \beta \leq 3 \quad (1-59)$$

Dans le cas de CS, l'emploi du Levy Flight améliore et optimise la recherche : des solutions nouvelles sont générées par une marche aléatoire de Levy autour de la meilleure solution obtenue jusqu'alors, ce qui accélère la recherche globale.

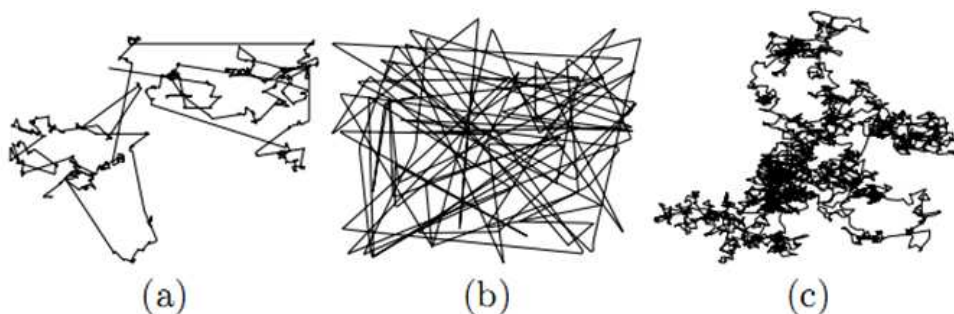


Figure 1-8: Levy flight (a), marche aléatoire simple (b) et mouvement brownien (c)

D'un point de vue implémentation, la génération d'un nombre aléatoire avec le Levy Flight suit deux étapes : le choix d'une direction aléatoire et la génération du pas qui doit obéir à la distribution de Levy. La génération d'une direction peut être réalisée à partir d'une distribution uniforme, alors que la génération des pas est plus délicate. Il y a plusieurs méthodes pour y parvenir mais une des plus simples et des plus efficaces consiste à employer les formules de Mantegna pour déterminer le pas s :

$$s = \frac{u}{|v|^{1/\beta}} \quad (1-60)$$

où u et v sont des distributions gaussiennes centrées telles que :

$$u = N(0, \sigma_u^2), \quad v = N(0, \sigma_v^2) \quad (1-61)$$

avec :

$$\sigma_u^2 = \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{(\beta-1)/2}}, \quad \sigma_v^2 = 1 \quad (1-62)$$

où $\Gamma(z)$ est la fonction Gamma :

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt \quad (1-63)$$

Initialisation de l'algorithme

La population initiale est constituée d'un nombre N_H de nids hôtes générés aléatoirement dans l'espace de recherche. On rappelle que le nombre de nids hôtes, de coucous et d'œufs sont égaux.

En supposant que le $i^{\text{ème}}$ nid hôte soit représenté par $X_i = (x_{i1}, \dots, x_{ij}, \dots, x_{in})$ où n est la dimension du problème, alors chaque nid est généré par :

$$x_{ij} = X_{\min j} + \text{rnd}(0,1)(X_{\max j} - X_{\min j}), \quad j = 1, \dots, n, \quad i = 1, \dots, N_H \quad (1-64)$$

Les valeurs des fonctions de coût correspondantes sont évaluées.

Description de l'algorithme

- Etape 1 : recherche globale

La première étape consiste à réaliser une recherche globale. Pour cela, on génère de nouveaux coucous (solutions) tests à partir des coucous existants (solutions) en réalisant des Levy Flight. Cela consiste à employer la loi d'évolution suivante pour chaque coucou :

$$\left. \begin{aligned} X_{new i} &= X_i + \alpha \otimes \text{Levy}(\beta) \\ F_{new i} &= f(X_{new i}) \end{aligned} \right\}, \quad i = 1, \dots, N_H \quad (1-65)$$

où $\alpha > 0$ est la taille du pas de déplacement qui dépend du problème considéré et tel que $\alpha \approx O(L/10)$, où L est l'échelle caractéristique du problème considéré. Le produit \otimes est le produit terme à terme.

Afin de savoir où/si le nouveau $i^{\text{ème}}$ coucou va déposer un œuf, il choisit aléatoirement un nid hôte X_j ($j \neq i$) de fonction de coût F_j .

Le nouveau coucou pond si :

$$F_{new\ i} < F_j \quad (1-66)$$

Dans ce cas, la solution $X_{new\ i}$ remplace la solution X_j dans la population.

- Etape 2 : recherche locale

Les nouveaux œufs de coucous ont une probabilité p_a d'être découvert.

Si c'est le cas ($rnd(0,1) < p_a$), de nouveaux œufs sont générés. Plusieurs méthodes peuvent être envisagées pour cela. On retiendra pour CS la suivante où on procède à une hybridation entre éléments de la population choisie aléatoirement :

$$\begin{aligned} x'_{new\ ij} &= x_{ij} + rnd(0,1)(x_{pj} - x_{mj}), \quad j = 1, \dots, n, \quad i = 1, \dots, N_H \\ F'_{new\ i} &= f(X'_{new\ i}) \end{aligned} \quad (1-67)$$

où $m \in \{1, 2, 3, \dots, N_H\}$ et $p \in \{1, 2, 3, \dots, N_H\}$ tels que $p \neq m$ sont choisis aléatoirement.

Une sélection identique à celle de l'étape précédente est alors réalisée.

Pseudo-code de l'algorithme

Le pseudo code simplifié de l'algorithme est donné ci-dessous.

Génération d'une population initiale de N_H nids hôtes X_i
 Tant que le critère d'arrêt n'est pas satisfait, répéter :
 Générer un coucou par Levy Flight (1-65) et évaluer sa fonction de coût F_i ,
 Choisir aléatoirement un nid X_j parmi les N_H nids hôtes,
 Si $F_i < F_j$
 Remplacer F_j par cette nouvelle solution,
 Fin Si
 Une fraction p_a des plus mauvais nids est abandonné et d'autres sont générés à la place (1-67), et faire une sélection.
 Identifier la meilleure solution X_{best}
 Fin Tant que

(1-68)

Réglage de l'algorithme

On retiendra les réglages standards suivant pour CS [YAN 09a] :

- Probabilité de découverte des œufs : $p_a = 0,25$
- Taille du pas de déplacement : $\alpha = \frac{\|X_{\min} - X_{\max}\|}{10}$
- Dimensionnement du Levy Flight : $\beta = 3/2$

Il n'y a pas de règles empiriques quant au choix du nombre de nids. Toutefois, suivant les mêmes principes que PSO, l'algorithme a donné ses meilleurs résultats avec un nombre de nids égal à :

$$N_H = \text{floor}(10 + \sqrt{n}) \quad (1-69)$$

Variantes de l'algorithme

Les différentes versions existantes consistent en la modification de la phase d'exploration locale et/ou globale. On retiendra la version CS/DE [WAN 12] qui consiste à hybrider l'algorithme avec l'évolution différentielle (DE). Plus précisément, le principe est d'employer dans la phase de recherche globale de CS les opérations de mutation et de croisement de DE/Rand/1 (voir § 1.4) pour respectivement générer les nouvelles solutions à la place du Levy Flight et pour éventuellement les sélectionner. Ce mécanisme de génération/sélection est donné en (1-70).

```

Pour i = 1 à NH, faire
  Sélectionner aléatoirement r1 ≠ r2 ≠ r3 ≠ i,
  Xv = Xr1 + F × (Xr2 - Xr3),
  r4 = floor(1 + rnd(0,1) × NH)
  Pour j = 1 à n, faire
    Si rnd(0,1) ≤ Cr ou j == r4 alors
      Xu(j) = Xv(j)
    Sinon
      Xu(j) = Xi(j)
    Fin si
  Fin Pour

  Si F(Xu) < F(Xr4) alors
    Xr4 = Xu
    F(Xr4) = F(Xu)
  Fin Si
Fin Pour

```

(1-70)

On retrouve les opérateurs de mutation et de croisement de DE. On retiendra les réglages standards suivants : $F = 0,75$ et $C_r = 0,8$.

Le pseudocode de CS/DE est donné ci-dessous :

```

Génération d'une population initiale de NH nids hôtes Xi
Tant que le critère d'arrêt n'est pas satisfait, répéter :
  Générer une nouvelle population de coucou par (1-70)

```

(1-71)

Procéder à la recherche locale (étape 2)
 Identifier la meilleure solution X_{best}
 Fin Tant que

Versions retenues

Nous retenons *CS* et *CS/DE*.

1.7. Bilan

Les algorithmes et leurs variantes retenus peuvent être rassemblés dans le tableau suivant :

Nom de l'algorithme	Description
Std_PSO	Algorithme des essaims particuliers, version <i>LocalBest</i> , réglages canoniques.
TVRndIW_PSO	Algorithme des essaims particuliers, version <i>LocalBest</i> , réglages canoniques avec facteur d'inertie stochastique autour de la valeur canonique.
StaticCEC_QPSO	Algorithme des essaims particuliers quantiques classiques avec coefficient de contraction-expansion constants.
TVDdecCEC_QPSO	Algorithme des essaims particuliers quantiques classiques avec coefficient de contraction-expansion décroissant dans le temps.
DE	Algorithme de l'évolution différentielle en version DE/Rand/1.
Rnd2Bst_DE	Algorithme de l'évolution différentielle en version DE/Rand-2-best/1
ABC	Algorithme des colonies d'abeilles standard.
CS	Algorithme de la recherche coucou classique.
CS/DE	Algorithme de la recherche coucou hybridé avec DE/Rand/1

Partie 2

Reformulation des problèmes de commande robuste pour l'optimisation stochastique.

2.1. Introduction

On s'intéresse dans cette partie à un premier inconvénient majeur inhérent aux méthodes de synthèse robuste avancées modernes : l'ordre des correcteurs synthétisés est grand. Ceci est d'autant plus le cas avec la synthèse H_∞ par *loop-shaping* car le correcteur final a au moins 2 fois l'ordre des pondérations. Cet aspect de la synthèse peut devenir vite problématique ; en effet, des performances demandées aux asservissements toujours plus accrues associées à un besoin de robustesse toujours plus important poussent l'ingénieur à la mise en œuvre de plus en plus de capteurs. La nécessité de développer des systèmes toujours à moindre coût implique l'emploi de calculateurs aux capacités limitées si bien que l'ingénieur est nécessairement contraint à limiter l'ordre des lois de commandes qu'il implémente. La post-réduction des correcteurs est possible mais se pose alors la question de la robustesse du correcteur réduit : il convient donc de contraindre l'ordre des correcteurs *a priori* lors de leur synthèse.

2.2. Synthèse H_∞ structurée

2.2.1. Introduction

On considère la forme standard pour la commande de la Figure 2-1 où :

- l'entrée e rassemble les m_1 entrées exogènes ;
- la sortie z rassemble les p_1 signaux à surveiller ;
- y rassemble les p_2 mesures employées par le correcteur ;
- u rassemble les m_2 commandes élaborées par le correcteur ;

$$e(t) \in \mathbf{R}^{m_1}, u(t) \in \mathbf{R}^{m_2}, z(t) \in \mathbf{R}^{p_1}, y(t) \in \mathbf{R}^{p_2}$$

$$\begin{pmatrix} z(s) \\ y(s) \end{pmatrix} = P(s) \begin{pmatrix} e(s) \\ u(s) \end{pmatrix} = \begin{pmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{pmatrix} \begin{pmatrix} e(s) \\ u(s) \end{pmatrix} \quad (2-1)$$

Le problème H_∞ standard à ordre fixé consiste à déterminer un correcteur $K(s)$ d'ordre n_k qui :

- stabilise la boucle fermée $F_l(P,K)$ de la Figure 2-1 de manière interne ;
- minimise γ tel que :

$$\|F_l(P, K)\|_\infty < \gamma \quad (2-2)$$

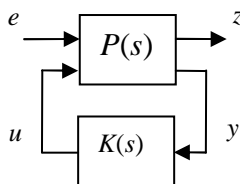


Figure 2-1: Forme standard pour la synthèse H_∞

$P(s)$ dispose de la représentation d'état suivante :

$$\begin{pmatrix} \dot{x}(t) \\ e(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{pmatrix} \begin{pmatrix} x(t) \\ w(t) \\ u(t) \end{pmatrix} \quad (2-3)$$

et $\dim(A) = n_p$.

On note de plus :

$$K(s) := \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right] \quad (2-4)$$

On montre que la forme standard de la Figure 2-1 bouclée par $K(s)$ dispose de la représentation d'état suivante :

$$\begin{aligned}
 T(s) &= F_l(P(s), K(s)) \\
 &= \left[\begin{array}{cc|c} A + B_2 M_1 D_k C_2 & B_2 M_1 C_k & B_1 + B_2 M_1 D_k D_{21} \\ B_k M_2 C_2 & A_k + B_k M_2 D_{22} C_k & B_k M_2 D_{21} \\ \hline C_1 + D_{12} M_1 D_k C_2 & D_{12} M_1 C_k & D_{11} + D_{12} M_1 D_k D_{21} \end{array} \right] \quad (2-5) \\
 M_1 &= (I - D_k D_{22})^{-1} \\
 M_2 &= (I - D_{22} D_k)^{-1}
 \end{aligned}$$

Pour des raisons de lisibilité, on se limite souvent au cas où $D_{22} = 0$ ce qui n'est en rien restrictif en vertu de la procédure de *loop-shifting* [ZHO 96] :

$$T(s) = \left[\begin{array}{cc|c} A + B_2 D_k C_2 & B_2 C_k & B_1 + B_2 D_k D_{21} \\ B_k C_2 & A_k & B_k D_{21} \\ \hline C_1 + D_{12} D_k C_2 & D_{12} C_k & D_{11} + D_{12} D_k D_{21} \end{array} \right] = \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array} \right] \quad (2-6)$$

Dans le cas où le correcteur est statique, alors la boucle fermée devient :

$$T(s) = \left[\begin{array}{c|c} A + B_2 K C_2 & B_1 + B_2 K D_{21} \\ \hline C_1 + D_{12} K C_2 & D_{11} + D_{12} K D_{21} \end{array} \right] \quad (2-7)$$

En posant:

$$\Theta_k = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix} \quad (2-8)$$

Alors on constate que (2-6) est équivalente à :

$$T(s) = \left[\begin{array}{c|c} \bar{A} + \bar{B}_2 \Theta_k \bar{C}_2 & \bar{B}_1 + \bar{B}_2 \Theta_k \bar{D}_{21} \\ \hline \bar{C}_1 + \bar{D}_{12} \Theta_k \bar{C}_2 & \bar{D}_{11} + \bar{D}_{12} \Theta_k \bar{D}_{21} \end{array} \right] \quad (2-9)$$

Avec :

$$\begin{aligned}
 \bar{A} &= \begin{pmatrix} A & 0_{n_p \times n_k} \\ 0_{n_k \times n_p} & 0_{n_k \times n_k} \end{pmatrix}, \bar{B}_1 = \begin{pmatrix} B_1 \\ 0_{n_k \times n_e} \end{pmatrix}, \bar{B}_2 = \begin{pmatrix} 0_{n_p \times n_k} & B_2 \\ I_{n_k} & 0_{n_k \times m} \end{pmatrix} \\
 \bar{C}_1 &= \begin{pmatrix} C_1 & 0_{n_z \times n_k} \end{pmatrix}, \bar{D}_{11} = D_{11}, \quad \bar{D}_{12} = \begin{pmatrix} 0_{n_z \times n_k} & D_{12} \end{pmatrix} \\
 \bar{C}_2 &= \begin{pmatrix} 0_{n_k \times n_p} & I_{n_k} \\ C_2 & 0_{p \times n_k} \end{pmatrix}, \bar{D}_{21} = \begin{pmatrix} 0_{n_k \times n_e} \\ D_{21} \end{pmatrix}
 \end{aligned} \quad (2-10)$$

En rapprochant (2-7) et (2-9) on constate que tout retour dynamique de sortie peut formellement se mettre sous la forme d'un retour statique de sortie sur le système augmenté défini par (2-10). Ainsi le problème de la synthèse d'un retour dynamique de sortie d'ordre quelconque $K(s) = [A_k, B_k, C_k, D_k]$ peut être réduit à celui de la synthèse d'un retour statique de sortie Θ_k défini par (2-8).

2.2.2. La limite des inégalités matricielles affines (LMI)

Le calcul d'un correcteur d'ordre plein (c'est-à-dire d'ordre $n_k = n_p$) est possible par une approche basée sur la résolution d'équations de Riccati [DOY 89] mais celle-ci ne permet pas de synthétiser des correcteurs d'ordre fixé. Apparue plus récemment, la synthèse H_∞ par LMI constitue une alternative intéressante. Elle est moins restrictive et est basée sur la forme standard. Il en existe plusieurs versions et on retient ici celle qui se décompose en deux étapes [GAH 94] : d'abord l'étude de la faisabilité du problème H_∞ standard puis la reconstruction du correcteur H_∞ -optimal.

2.2.2.1. Faisabilité du problème standard

La faisabilité du problème standard est testée au moyen du théorème suivant. En supposant (A, B_2) stabilisable et (C_2, A) détectable, le problème H_∞ standard admet une solution si et seulement si il existe 2 matrices R et S , vérifiant les 3 LMI suivantes :

$$\begin{pmatrix} N_R & 0 \\ 0 & I_{n_e} \end{pmatrix}^T \begin{pmatrix} AR + RA^T & RC_1^T & B_1 \\ C_1 R & -\gamma I_{n_z} & D_{11} \\ B_1^T & D_{11}^T & -\gamma I_{n_e} \end{pmatrix} \begin{pmatrix} N_R & 0 \\ 0 & I_{n_e} \end{pmatrix} < 0 \quad (2-11)$$

$$\begin{pmatrix} N_S & 0 \\ 0 & I_{n_z} \end{pmatrix}^T \begin{pmatrix} A^T S + SA & SB_1 & C_1^T \\ B_1^T S & -\gamma I_{n_e} & D_{11}^T \\ C_1 & D_{11} & -\gamma I_{n_z} \end{pmatrix} \begin{pmatrix} N_S & 0 \\ 0 & I_{n_z} \end{pmatrix} < 0 \quad (2-12)$$

$$\begin{pmatrix} R & I_{n_p} \\ I_{n_p} & S \end{pmatrix} \geq 0 \quad (2-13)$$

où N_R et N_S constituent une base des noyaux de $(B_2^T \ D_{12}^T)$ et $(C_2 \ D_{21})$ respectivement.

Par ailleurs, la valeur minimale de γ s'obtient en résolvant le problème d'optimisation convexe suivant :

$$\min_{R=R^T, S=S^T} \gamma \quad (2-14)$$

Sous contraintes (2-11)-(2-13)

2.2.2.1.1. Reconstruction du correcteur optimal

Des correcteurs d'ordre $n_k \leq n_p$ existent si et seulement si les LMI (2-11) à (2-13) sont vérifiées par des matrices R et S satisfaisant la condition :

$$\text{rang} \begin{pmatrix} R & I_{n_p} \\ I_{n_p} & S \end{pmatrix} \leq n_p + n_k \Leftrightarrow \text{rang} (I_{n_p} - RS) \leq n_k \quad (2-15)$$

Le correcteur final s'obtient en appliquant le lemme borné réel [BOY 94] sur le système (2-6) : $\|T(s)\|_\infty \leq \gamma$ si et seulement si il existe une matrice X symétrique définie positive vérifiant :

$$\begin{pmatrix} A_{cl}^T X + X A_{cl} & X B_{cl} & C_{cl}^T \\ B_{cl}^T X & -\gamma I_{n_e} & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma I_{n_z} \end{pmatrix} < 0 \quad (2-16)$$

qui est une inégalité matricielle bilinéaire en X, A_k, B_k, C_k, D_k . Une matrice X qui convient est [GAH 94] :

$$X = \begin{pmatrix} S & N \\ N^T & -M^+ RN \end{pmatrix} \quad (2-17)$$

Où M^+ désigne la pseudo inverse de M (c'est-à-dire $M^+M = I$) et où M, N s'obtiennent en effectuant la décomposition en valeurs singulières suivante :

$$MN^T = I_{n_p} - RS \quad (2-18)$$

L'inégalité (2-18) devient alors une LMI en les variables A_k, B_k, C_k, D_k d'où on déduit $K(s)$ après sa résolution.

2.2.2.2. Observations

Lorsque l'ordre de la forme standard est important, cette approche peut souffrir de problèmes numériques. Toutefois cette approche dispose d'un intérêt certain car :

- elle est moins restrictive que l'approche par Riccati ;
- la contrainte d'ordre est explicitement reliée à une contrainte de rang : on peut donc théoriquement imposer l'ordre du correcteur par ce biais, bien qu'en

- réalité, comme nous le verrons plus loin, la contrainte de rang n'est pas convexe ;
- cette approche fournit un cadre beaucoup plus large pour la résolution de problèmes plus complexes tels que la synthèse multi-objectif [SCH 97].

Dans [VIN 97] on trouve une approche alternative, employant elle aussi le formalisme LMI, basée sur la stabilisation robuste sous contrainte d'ordre d'un système modélisé par ses factorisations premières.

2.2.2.3. Les limites de la synthèse H_∞ par les LMI et la contrainte d'ordre

D'après (2-15), pour disposer *a priori* d'un correcteur d'ordre inférieur à n_p , il suffit d'imposer une contrainte additionnelle aux LMI (2-11) à (2-13) sur le rang de la matrice $I_{n_p} - RS$ lors de l'étude de faisabilité du problème H_∞ standard. Ainsi le problème H_∞ d'ordre réduit n_k admet une solution si et seulement si il existe 2 matrices R et S , vérifiant :

$$\text{rang} \begin{pmatrix} R & I_{n_p} \\ I_{n_p} & S \end{pmatrix} \leq n_p + n_k \quad (2-19)$$

Sous contraintes (2-11)-(2-13)

Or, la contrainte de rang n'est pas convexe ; cependant, un cas particulier intéressant est le retour statique de sortie, car :

- tout problème de synthèse de retour de sortie dynamique peut se réduire à celui de la synthèse d'un retour de sortie statique sur un système augmenté ;
- dans ce cas la contrainte de rang devient :

$$\text{rang} \begin{pmatrix} R & I_{n_p} \\ I_{n_p} & S \end{pmatrix} \leq n_p \Leftrightarrow \text{rang} (I_{n_p} - RS) = 0 \Leftrightarrow I_{n_p} - RS = 0 \quad (2-20)$$

Il n'y a qu'en de très rares cas où la relation $R = S^{-1}$ peut être explicitement prise en compte dans l'étape de faisabilité [PRE 04].

Dans [APK 03], on utilise les lagrangiens augmentés pour résoudre localement le problème suivant :

$$\begin{aligned} \min f(x) &:= \gamma \\ \text{sous contrainte } h(x) &:= I - RS = 0 \\ x &\in \mathcal{X} \end{aligned} \quad (2-21)$$

Où :

$$x = (R, S, \gamma) \quad (2-22)$$

Et :

$$\chi := \{x \text{ tel que (2-11)-(2-13)}\} \quad (2-23)$$

Les auteurs proposent alors de résoudre à la place de (2-21) une série de problèmes relaxés de la forme (2-24).

$$\min_{R=R^T, S=S^T, \gamma} g(R, S, \gamma) \quad (2-24)$$

Sous contraintes (2-11)-(2-13)

Où $g(R, S, \gamma)$ est le Lagrangien partiellement augmenté :

$$g(R, S, \gamma) = \gamma + \text{trace}(\Lambda^T (RS - I)) + \frac{c}{2} \text{trace}((RS - I)^T (RS - I)) \quad (2-25)$$

c : coefficient de pénalité

Λ : multiplicateur matriciel de Lagrange

Cette approche fournit des résultats intéressants mais pour des formes standards de petits ordres.

Enfin, certains travaux plus conservatifs ont été réalisés dans le but d'attaquer la contrainte de rang de manière plus directe mais sans plus de garanties de convergence globale. On retiendra par exemple [ELG 97] où on « sature » la contrainte de rang (2-19) en résolvant le problème d'optimisation suivant :

$$\min_{R=R^T, S=S^T} \text{trace}(RS) \quad (2-26)$$

Sous contraintes (2-11)-(2-13)

Il existe un retour statique de sortie si et seulement si le minimum global du problème (2-26) est n . Une méthode itérative employant une approximation linéaire de $\text{trace}(RS)$ est alors employée :

$$\Phi_{\text{lin}}(R, S) = \text{constante} + \text{trace}(S_0 R + R_0 S) \quad (2-27)$$

Une mise en œuvre de cet algorithme peut être trouvée dans [PRE 05].

Signalons enfin l'existence de certaines heuristiques sous-optimales intéressantes telles que la X - K itérations [ELG 94] où on propose de résoudre la LMI (2-43) du lemme borné alternativement suivant X et Θ_k .

$$\begin{pmatrix} A_{cl}^T X + X A_{cl} & X B_{cl} & C_{cl}^T \\ B_{cl}^T X & -\gamma I_{n_e} & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma I_{n_z} \end{pmatrix} < 0 \quad (2-28)$$

Où :

$$T(s) = \left[\begin{array}{c|c} \overline{A} + \overline{B}_2 \Theta_k \overline{C}_2 & \overline{B}_1 + \overline{B}_2 \Theta_k \overline{D}_{21} \\ \hline \overline{C}_1 + \overline{D}_{12} \Theta_k \overline{C}_2 & \overline{D}_{11} + \overline{D}_{12} \Theta_k \overline{D}_{21} \end{array} \right] = \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array} \right] \quad (2-29)$$

En effet :

- à X fixé, (2-43) est une LMI en Θ_k et γ ;
- à Θ_k et γ fixés, (2-43) est une LMI en X .

Cette approche n'est pas sans rappeler celle de [VIN 97] : elle est sous-optimale, nécessite d'être initialisée par un correcteur stabilisant, mais est beaucoup plus générale puisqu'elle s'applique sur toute forme standard.

Comme nous le voyons, la synthèse de correcteur d'ordre réduit n'est pas simple à l'aide des LMI car la contrainte de rang n'est pas convexe. Par ailleurs, l'optimisation par LMI nécessite souvent un nombre de variables de décisions important (variant de manière polynomiale avec l'ordre du système) occasionnant des temps de calcul souvent longs, entachés souvent de problèmes numériques.

2.2.3. Synthèse H_∞ par l'optimisation non lisse

2.2.3.1. Rappel sur les méthodes de descente

On considère le problème d'optimisation :

$$\min_{x \in \mathbf{R}^n} f(x), \quad x^* = \arg \min_{x \in \mathbf{R}^n} f(x) \quad (2-30)$$

Où la fonction $f(x)$ est différentiable.

Les méthodes de descente consistent à utiliser un algorithme récursif de déplacement dans l'espace de recherche. A chaque itération, on choisit de se déplacer dans une direction permettant de faire diminuer la fonction $f(x)$:

$$x_{k+1} = x_k + \lambda_k d_k \quad (2-31)$$

Où :

- $d_k \in \mathbf{R}^n$ est la direction de descente,
- $\lambda_k > 0$ est appelé le pas.

Lorsque la fonction $f(x)$ est différentiable, une méthode courante est de choisir le gradient de $f(x)$ comme direction de descente :

$$d_k = -\nabla f(x_k) \quad (2-32)$$

Cette direction correspond à la direction de plus forte diminution de la fonction $f(x)$ dans l'ensemble des directions de descente de f en x_k :

$$\{d \in \mathbf{R}^n : d^T \nabla f(x_k) < 0\} \quad (2-33)$$

Pour choisir le pas, on peut choisir dans la méthode de plus profonde descente :

- λ_k constant, solution simple mais donnant rarement de bon résultats,
- λ_k optimisé à chaque étape :

$$\lambda_k = \arg \min_{\lambda} f(x_k + \lambda d_k) \quad (2-34)$$

Le problème (2-34) est un problème d'optimisation unidimensionnelle à réaliser à chaque étape qui peut être appréhendé par des méthodes de type *Gridding*, méthode de Newton-Raphson, méthode de la sécante, etc...

L'inconvénient de la méthode précédente est que l'on ne tient pas compte de la courbure de la fonction à minimiser, si bien que le piégeage dans un minimum local est très rapide (dans ce cas, le gradient vient à s'annuler et l'optimisation s'arrête).

Ainsi il existe d'autres méthodes de descente où la fonction de coût est approximée par une expression plus simple (c'est-à-dire que l'on sait optimiser), et pour lesquelles on peut se diriger vers le point qui minimise cette approximation. Dans la méthode de Newton, il est d'usage de prendre pour approximation un développement de Taylor à l'ordre 2 de la fonction $f(x)$ au voisinage de x_k :

$$g(x) = f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k) \approx f(x) \quad (2-35)$$

On choisit alors de se déplacer dans la direction qui minimise cette approximation, c'est-à-dire :

$$x_{k+1} = \arg \min_x g(x) \quad (2-36)$$

En annulant le gradient $\nabla g(x)$ de l'approximation (2-35), on arrive à :

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \quad (2-37)$$

On constate que la méthode nécessite l'existence du hessien d'une part, et de son inversibilité d'autre part. Parce que l'inversion du hessien est couteuse en temps de calcul et souvent mal conditionnée, on choisit en général une direction de descente de la forme :

$$x_{k+1} = x_k - M_k \nabla f(x_k) \quad (2-38)$$

Où la matrice M_k est une matrice mise à jour à chaque itération et ayant certaines propriétés du hessien :

$$\begin{aligned}
M_k &> 0 \quad (\text{afin d'avoir } d_k^T \nabla f(x_k) < 0) \\
M_{k+1} (\nabla f(x_{k+1}) - \nabla f(x_k)) &= x_{k+1} - x_k \\
M_{k+1} &= \arg \min_M \|M - M_k\|
\end{aligned} \tag{2-39}$$

La méthode la plus connue est appelée BFGS (Broyden-Fletcher-Goldfarb-Shanno) et consiste à choisir :

$$\begin{aligned}
M_{k+1} &= V_k M_k V_k^T + \alpha_k (d_k^T y_k)^{-1} d_k d_k^T \\
y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \\
V_k &= I - (d_k^T y_k)^{-1} d_k y_k^T \\
M_0 &= I
\end{aligned} \tag{2-40}$$

2.2.3.2. Principe de l'approche

Face à la difficulté que représente la contrainte de rang lors de la résolution par les LMI, l'apparition de nouveaux outils d'optimisation permet de contourner cette dernière en attaquant le problème de la synthèse de correcteur d'ordre réduit de manière frontale.

Le problème consiste à trouver $\Theta_k = \begin{pmatrix} A_K & B_K \\ C_K & D_K \end{pmatrix}$ d'ordre $n_k < n_p$ tel que :

- $T(s)$ soit stable de manière interne,
- $\|T(s)\|_\infty < \gamma$, avec γ minimal.

Avec :

$$T(\Theta_k) = C_{cl}(\Theta_k) (sI_{n_p+n_k} - A_{cl}(\Theta_k))^{-1} B_{cl}(\Theta_k) + D_{cl}(\Theta_k) \tag{2-41}$$

Et :

$$\begin{aligned}
A_{cl}(\Theta_k) &= \bar{A} + \bar{B}_2 \Theta_k \bar{C}_2 \\
B_{cl}(\Theta_k) &= \bar{B}_1 + \bar{B}_2 \Theta_k \bar{D}_{21} \\
C_{cl}(\Theta_k) &= \bar{C}_1 + \bar{D}_{12} \Theta_k \bar{C}_2 \\
D_{cl}(\Theta_k) &= \bar{D}_{11} + \bar{D}_{12} \Theta_k \bar{D}_{21}
\end{aligned} \tag{2-42}$$

Le système bouclé est stable si :

$$\max_i \left(\text{Re} \left(\lambda_i \left(\bar{A} + \bar{B}_2 \Theta_k \bar{C}_2 \right) \right) \right) < 0 \tag{2-43}$$

Où $\text{Re}(\cdot)$ est la partie réelle et λ_i la $i^{\text{ème}}$ valeur propre.

Par ailleurs, par définition de la norme H_∞ :

$$\|T(s)\|_\infty < \gamma \Leftrightarrow \max_{\omega \in [0, +\infty[} (\bar{\sigma}(T(j\omega))) < \gamma \quad (2-44)$$

Où $\bar{\sigma}$ est la plus grande des valeurs singulières.

La synthèse du correcteur se déroule en deux étapes. D'abord l'étape de stabilisation de la boucle fermée qui consiste à résoudre le problème d'optimisation suivant :

$$\min_{\Theta_k} f(\Theta_k), f(\Theta_k) = \max_i \left(\operatorname{Re}(\lambda_i(\bar{A} + \bar{B}_2 \Theta_k \bar{C}_2)) \right) \quad (2-45)$$

La norme H_∞ de la boucle fermée est quant à elle minimisée en résolvant le problème d'optimisation suivant :

$$\min_{\Theta_k} f(\Theta_k), f(\Theta_k) = \max_{\omega \in [0, +\infty[} \bar{\sigma}(T(j\omega)) \quad (2-46)$$

Ainsi les deux étapes sont des problèmes d'optimisation (non nécessairement convexes) de type *minmax* de la forme :

$$\min_x f(x), f(x) = \max_{s \in S} \phi_s(x), x \in \mathbf{R}^n \quad (2-47)$$

Ce type de problème est parfaitement adapté à l'optimisation non lisse [ROC 94] : des algorithmes d'optimisation existent [LEW 12], la seule contrainte étant le caractère lipschitzien de la fonction à minimiser.

Une fonction $f : \mathbf{R}^n \rightarrow \mathbf{R}$ est dite lipschitzienne sur $H \subset \mathbf{R}^n$ s'il existe une constante $L > 0$ telle que, pour tout $y, z \in H$:

$$|f(y) - f(z)| \leq L \|y - z\| \quad (2-48)$$

Le caractère localement lipschitzien d'une fonction en x assure donc que son taux d'accroissement soit limité au voisinage de x . Une fonction localement lipschitzienne en un point n'est pas nécessairement différentiable en ce point (par exemple la fonction $x \rightarrow |x|$ en 0). Par ailleurs toute fonction convexe $f : \mathbf{R}^n \rightarrow \mathbf{R}$ est localement lipschitzienne en tout point de \mathbf{R}^n (et donc f est lipschitzienne sur tout compact de \mathbf{R}^n).

Lorsque la fonction f est lipschitzienne, on généralise la notion de gradient (établie pour les fonctions différentiables) par la notion de sous-différentiel à l'aide de la dérivée directionnelle. Pour tout $w \in \mathbf{R}^n$, la fonction f admet une dérivée directionnelle en x dans la direction w , notée $df(x)(w)$, si la limite suivante existe et est finie :

$$df(x)(w) = \lim_{\substack{t \rightarrow 0 \\ t > 0}} \frac{f(x + tw) - f(x)}{t} \quad (2-49)$$

$df(x)$ s'appelle sous-différentielle de f en x et lorsque f est lipschitzienne :

$$|df(x)(w)| \leq L\|w\| \quad (2-50)$$

A noter que si f est différentiable en x , alors elle admet des dérivées directionnelles dans toutes les directions w , et on a :

$$df(x)(w) = (\nabla f(x))^T w \quad (2-51)$$

On montre que le sous-différentiel dispose des propriétés suivantes :

$$\begin{aligned} df(x)(w_1 + w_2) &\leq df(x)w_1 + df(x)w_2 \\ df(x)(\lambda w) &\leq \lambda df(x)w, \lambda > 0 \end{aligned} \quad (2-52)$$

L'ensemble des sous-gradients de f en x est défini par :

$$\partial f(x) = \left\{ v \in \mathbf{R}^n \mid v \cdot w \leq df(x)(w), \forall w \in \mathbf{R}^n \right\} \quad (2-53)$$

$w \in \mathbf{R}^n$ constitue une direction de descente de f en x si l'une des conditions suivantes est remplie :

- $df(x)(w) < 0$,
- $w^T g < 0, \forall g \in \partial f(x)$.

Ainsi l'optimisation non lisse s'avère adaptée à la minimisation de fonctions non régulières car comparée aux méthodes de descentes usuelles où on emploie le gradient de f en x pour déterminer la direction de descente, on emploiera, dans le cadre de fonction non régulières, un sous-gradient de f en x pouvant être défini comme une combinaison linéaire de dérivées directionnelles de f en x .

Basée sur ces notions et sur l'approche standard pour la commande, une synthèse H_∞ par optimisation non lisse est proposée dans [APK 02].

On pose :

$$f(\Theta_k) = \|T(s)\|_\infty \quad (2-54)$$

On considère la décomposition en valeur singulière⁶ de $T(j\omega)$:

$$T(j\omega) = U\Sigma V^H \quad (2-55)$$

⁶ Dans ce paragraphe, M^H désigne la transposée conjuguée de M .

On pose u et v comme étant respectivement les vecteurs à gauche et à droite correspondant à la plus grande des valeurs singulières issue de la décomposition en valeur singulière de T .

On montre alors que [APK 02] :

$$\left. \begin{aligned} \frac{\partial f}{\partial A_{cl}} &= \phi^* C_{cl}^* u v^* B_{cl}^* \phi^* \\ \frac{\partial f}{\partial B_{cl}} &= \phi^* C_{cl}^* u v^* \\ \frac{\partial f}{\partial C_{cl}} &= u v^* B_{cl}^* \phi^* \\ \frac{\partial f}{\partial D_{cl}} &= u v^* \end{aligned} \right\} := \partial \| \cdot \|_{\infty} (T(\Theta_k)) \quad (2-56)$$

$$\text{Où } \phi = (j\omega I - A_{cl})^{-1}, M = (C_{cl} \phi B_{cl} + D_{cl}).$$

On détermine alors le sous-différentiel de la fonction $f(\Theta_k) = \|T(\Theta_k)\|_{\infty}$ par rapport à Θ_k en utilisant la relation de composition suivante :

$$\partial f(\Theta_k) = T'(\Theta_k)^* (\partial \| \cdot \|_{\infty} (T(\Theta_k))) \quad (2-57)$$

où $\partial \| \cdot \|_{\infty}$ est le sous-différentiel de la norme H_{∞} déterminé précédemment, et $T'(\Theta_k)^*$ est l'adjoint⁷ de $T'(\Theta_k)$ défini par :

$$T'(\Theta_k) \delta \Theta_k := G_{12} \delta \Theta_k G_{21} \quad (2-58)$$

Avec :

$$\begin{aligned} & \begin{pmatrix} T(\Theta_k, s) & G_{12}(\Theta_k, s) \\ G_{21}(\Theta_k, s) & * \end{pmatrix} \\ &= \begin{pmatrix} C_{cl}(\Theta_k) \\ C_2 \end{pmatrix} (sI - A_{cl}(\Theta_k))^{-1} (B_{cl}(\Theta_k) \quad B_2) + \begin{pmatrix} D_{cl}(\Theta_k) & D_{12} \\ D_{21} & * \end{pmatrix} \end{aligned} \quad (2-59)$$

Rappel : Si A , B , X et C sont des matrices de dimensions compatibles, alors on montre que la dérivée de $f(A + BXC)$ par rapport à X est B^*GC^* avec

$$G = \left[\frac{\partial}{\partial x_{ij}} f(X) \right].$$

⁷ On rappelle que a^* est dit adjoint de a si $\langle a(x), y \rangle = \langle x, a^*(y) \rangle$.

Une méthode de descente peut alors employée en guise d'optimisation.

Noter qu'il convient de choisir un point initial lors de l'optimisation. Ce dernier est en général choisi aléatoirement si bien que plusieurs *run* d'optimisation peuvent ne pas donner le même résultat à cause de piègeages dans des optima locaux. Il convient donc d'en lancer plusieurs.

On trouvera dans [GAH 11] une mise en œuvre de la synthèse H_∞ par optimisation non lisse avec l'outil *Hinfstruct* dans Matlab. Signalons aussi la présence d'un autre outil gratuit et performant basé sur les mêmes principes d'optimisation, dénommé *Hifoo* [BUR 06].

2.2.3.3. Correcteurs décentralisés

La recherche des coefficients Θ_k de la représentation d'état du correcteur n'interdit nullement de structurer ce dernier comme bon nous semble, surtout dans le cas MIMO. Ainsi plutôt que de rester sur un correcteur centralisé (c'est-à-dire un correcteur où la dynamique d'état est commune à toutes ses entrées et sorties), il peut être préférable [HIR 12] de rechercher un correcteur décentralisé, c'est-à-dire un correcteur structuré suivant la Figure 2-2, mais au prix d'un temps de calcul plus élevé (puisque le nombre d'inconnues à optimiser augmente).

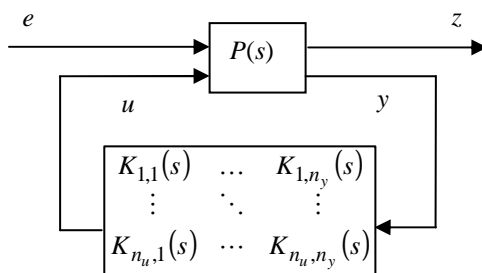


Figure 2-2: Synthèse H_∞ structurée et décentralisée

2.2.3.4. Extension au cadre multi-modèles

Les outils de synthèse H_∞ par optimisation non lisse précédemment cités fournissent un cadre d'emploi pouvant aller bien au-delà de la synthèse H_∞ et permettent notamment de traiter le cas de la synthèse H_∞ structurée multi-modèles. Comme montré plus-haut, une solution à ce problème a été proposée utilisant le formalisme LMI lors du problème de stabilisation sous contrainte d'ordre d'un système modélisé par ses factorisations premières, mais cette approche était limitée à la contrainte d'ordre et de structuration du précorrecteur (et pas du correcteur final). L'optimisation non lisse permet de généraliser l'approche standard de la synthèse H_∞ , à ce cadre.

On considère l'ensemble de m modèles suivant :

$$\mathcal{P} = \{P_i(s), i = 1, \dots, m\} \quad (2-60)$$

Comme sur la Figure 2-3, le problème H_∞ à ordre fixé structurée consiste à déterminer un correcteur $K(s)$ d'ordre n_k qui :

- stabilise chaque boucle fermée $F_l(P_i, K)$ de manière interne ;
- minimise γ tel que :

$$\max_{i=1, \dots, m} \|F_l(P_i, K)\|_\infty < \gamma \quad (2-61)$$

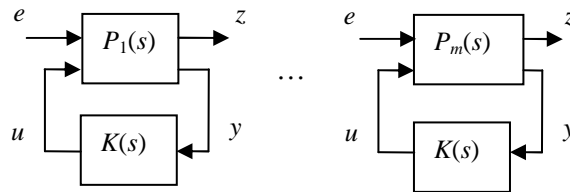


Figure 2-3: La synthèse H_∞ multi-modèles

On trouve un exemple de mise en œuvre de cette approche dans [APK 12]. La mise en œuvre de cette technique est rendue possible en construisant la forme standard de la Figure 2-4 et en imposant au correcteur une structuration particulière.

Ainsi on résout (2-61) en déterminant un correcteur $K'(s)$ d'ordre n_k qui :

- stabilise chaque boucle fermée $F_l(P', K')$ de manière interne ;
- minimise γ tel que :

$$\begin{aligned} \|F_l(P', K')\|_\infty &< \gamma \\ P'(s) &= \text{diag}(P_i(s), i = 1, \dots, m) \\ K'(s) &= \text{diag}(K_i(s), K_i(s) = K(s), i = 1, \dots, m) \end{aligned} \quad (2-62)$$

Noter que la forme standard augmentée ainsi construite a un ordre égal à la somme des ordres des modèles considérés, ce qui peut s'avérer élevée si le nombre de modèles est important, et ainsi ralentir en conséquence le processus d'optimisation du correcteur (nous reparlerons de ce point lors de la mise en œuvre de la synthèse H_∞ par l'optimisation stochastique dans le cadre multi-modèle).

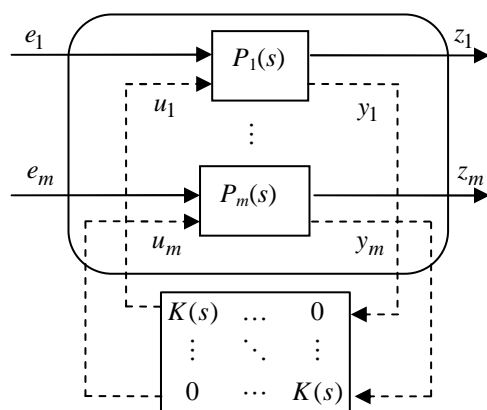


Figure 2-4: La synthèse H_∞ multi-modèles par optimisation non lisse

2.2.4. Formulation de la synthèse H_∞ pour l'optimisation stochastique

2.2.4.1. Principe

L'optimisation stochastique par les métaheuristiques est adaptée à la résolution de problèmes pour lesquels il est difficile de trouver un optimum global ou de bons optimums locaux par des méthodes plus classiques. On rappelle que ce type d'optimisation dispose de trois caractéristiques souvent décisives en optimisation globale :

- l'optimisation par les métaheuristiques ne nécessite pas de connaître le gradient de la fonction à minimiser, la seule contrainte étant de pouvoir évaluer cette dernière qui peut donc avoir une forme quelconque ;
- il n'est pas nécessaire de disposer d'un « bon » point initial, l'initialisation étant réalisée de manière aléatoire dans l'espace de recherche ;
- enfin ce type d'optimisation est stochastique ce qui permet de faire face à l'explosion combinatoire des possibilités et ce qui limite le piégeage dans les optima locaux.

On trouvera dans [DRE 06] une description détaillée des principales métaheuristiques inspirées par des analogies avec la physique, avec l'éthologie, avec la biologie...

Malgré le principal inconvénient qui concerne la pertinence de la solution trouvée évaluable statistiquement par plusieurs exécutions successives (ce qui occasionne un temps de calcul élevé en général), ce type d'optimisation s'avère intéressant pour les problèmes d'optimisation difficile et paraît adapté ici à la synthèse de correcteur d'ordre réduit.

Ainsi nous nous proposons ici de travailler sur une approche de la synthèse H_∞ structurée par l'optimisation stochastique.

En posant :

$$\bar{\lambda}_{cl} = \max_i (\operatorname{Re}(\lambda_i(A_{cl}))) \quad (2-63)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Alors on synthétise un correcteur stabilisant H_∞ -optimal en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_k} f(\Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq 0 \\ & \quad f(\Theta_k) = \bar{\lambda}_{cl} \\ & \text{sinon} \\ & \quad f(\Theta_k) = -\frac{1}{\|T(s)\|_\infty} \end{aligned} \quad (2-64)$$

Dans [ICH 08] on retrouve une idée similaire pour la définition de la fonction de coût envisagée ici. La fonction de coût consiste à réaliser deux étapes : tout d'abord, l'étape de stabilisation de la boucle fermée qui consiste à rendre l'abscisse spectrale $\bar{\lambda}$ strictement négative. Lorsque la boucle fermée devient stable, ses pôles sont alors, selon toute vraisemblance, proches de l'axe imaginaire de telle sorte que la valeur de γ obtenue avec le correcteur correspondant est très grande, et ainsi que $-\gamma^{-1}$ est négatif mais proche de 0. Ainsi il y a une pseudo-continuité dans la fonction de coût entre l'étape de stabilisation et l'étape d'optimisation proprement dite, dont il est bon de profiter ici.

Notons aussi que la fonction de coût ainsi définie ne dépend pas de l'algorithme d'optimisation stochastique (ou même plus généralement d'optimisation par recherche directe). Plus loin nous testons notre approche avec plusieurs métaheuristiques.

On trouve dans la littérature d'autres formulations de la fonction de coût telle que dans [POP 10], où on envisage l'emploi d'un algorithme évolutionnaire pour la synthèse H_∞ à ordre fixé avec la fonction suivante :

$$\begin{aligned} & \min_{\Theta_k} f(\Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq 0 \\ & \quad f(\Theta_k) = \beta + \bar{\lambda}_{cl} \\ & \text{sinon} \\ & \quad f(\Theta_k) = \|T(s)\|_\infty \end{aligned} \quad (2-65)$$

Où β est une pénalité se voulant grande, par exemple $\beta = 10^{10}$.

Pour les raisons citées précédemment, nous resterons sur la formulation (2-64).

2.2.4.2. Amélioration de la rapidité de convergence pour la synthèse des correcteurs

On propose ci-dessous d'améliorer sensiblement la rapidité de convergence du processus d'optimisation.

2.2.4.2.1. Utilisation de la forme tridiagonale

Nous avons posé pour inconnues la représentation d'état du correcteur :

$$\Theta_k = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right] \quad (2-66)$$

Or, pour une fonction de transfert unique, il existe une infinité de représentations d'état possibles. En supposant toutes les matrices de Θ_k pleines, alors le nombre d'inconnues pour un correcteur de type SISO s'établit à $(n_k + 1)^2$.

Une idée pour diminuer le nombre d'inconnues serait de travailler sur une forme compagnon comme la suivante :

$$\begin{aligned} A_k &= \begin{pmatrix} -a_1 & -a_2 & \dots & -a_{k-1} & -a_k \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \quad B_k = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ C_k &= (b_1 \quad b_2 \quad \dots \quad b_{k-1} \quad b_k), \quad D_k = d \end{aligned} \quad (2-67)$$

$$\Leftrightarrow$$

$$K(s) = \frac{b_1 s^{k-1} + b_2 s^{k-2} + \dots + b_{k-1} s + b_k}{s^k + a_1 s^{k-1} + \dots + a_{k-1} s + a_k} + d$$

En effet, dans le cas SISO le nombre d'inconnues s'établit dans ce cas à $2n_k + 1$. Cependant les valeurs des coefficients sont souvent trop différentes en dépit de leur faible nombre, ce qui peut rendre l'optimisation plus délicate.

Une meilleure approche consiste à employer la forme tridiagonale proposée dans [MCK 96]. Par définition, une matrice tridiagonale est une matrice carrée réelle qui contient des éléments non nuls uniquement sur ses diagonales principale, supérieure et inférieure :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & 0 & 0 \\ a_{2,1} & \ddots & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & 0 & a_{1,n-1} & a_{n,n} \end{pmatrix} \quad (2-68)$$

On montre dans [MCK 96] que pour tout système MIMO de représentation d'état $[A, B, C, D]$, il existe une représentation $[A', B', C', D']$ équivalente où A' est sous forme tridiagonale. Une idée consiste donc à chercher la matrice d'état du correcteur sous forme tridiagonale :

$$A_k = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & 0 & 0 \\ a_{2,1} & \ddots & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & a_{n_k-1, n_k} \\ 0 & 0 & 0 & a_{n_k, n_k-1} & a_{n_k, n_k} \end{pmatrix} \quad (2-69)$$

Les matrices B_k, C_k, D_k étant supposées pleines, le nombre d'inconnues pour un correcteur SISO s'établit alors à $5n_k - 1$ qui est du même ordre de grandeur que pour la forme compagnon.

Un cas particulier est celui du correcteur statique où le nombre d'inconnues s'établit à 1 (cela revient à ne chercher que la matrice de transfert direct).

2.2.4.2.2. Transformation de l'espace de recherche

Supposons que tous les paramètres cherchés soient strictement positifs. Dans ce cas, on peut transformer l'espace de recherche initial comme suit :

$$\begin{aligned} x \in [\underline{x}, \bar{x}] &\xrightarrow{x'=\log_{10}(x)} x' \in [\underline{x}', \bar{x}'] \\ x' \in [\underline{x}', \bar{x}'] &\xrightarrow{x=10^{x'}} x \in [\underline{x}, \bar{x}] \end{aligned} \quad (2-70)$$

Cette transformation améliore la sensibilité de l'algorithme : toutes les valeurs de x' sont du même ordre de grandeur ce qui n'est pas le cas avec l'espace initial. Ainsi, il est possible de choisir un intervalle de recherche initial très large, par exemple :

$$x \in [10^{-3}, 10^3] \xrightarrow{x'=\log_{10}(x)} x' \in [-3, 3] \quad (2-71)$$

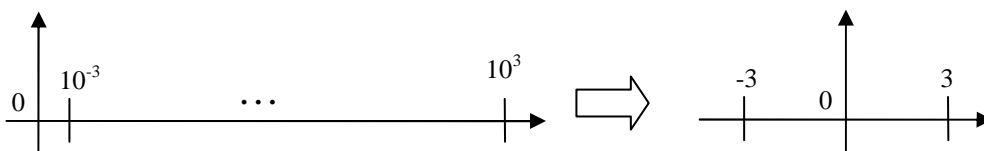


Figure 2-5: Transformation de l'espace de recherche

En réalité tous les paramètres recherchés peuvent être signés. Comme précédemment, on peut chercher à améliorer la sensibilité de l'algorithme par une transformation inspirée de la précédente :

$$\begin{aligned} x \in [\underline{x}, \bar{x}] &\xrightarrow{x' = \text{ash}_{10}(Mx)} x' \in [\underline{x}', \bar{x}'] \\ x' \in [\underline{x}', \bar{x}'] &\xrightarrow{x = \frac{1}{M} \text{sh}_{10}(x')} x \in [\underline{x}, \bar{x}] \end{aligned} \quad (2-72)$$

Avec :

$$\text{ash}_{10}(x) = \log_{10}\left(x + \sqrt{x^2 + 1}\right) ; \quad \text{sh}_{10}(x) = \frac{10^x - 10^{-x}}{2} \quad (2-73)$$

Lorsque $|x|$ est grand, les fonctions $\log_{10}(\)$ et $\text{ash}_{10}(\)$ sont très proches. Elles sont en revanche très différentes lorsque $|x|$ est proche de 0. Pour que les plus petites et les plus grandes valeurs de $|x|$ aient un poids similaire pendant l'optimisation, un coefficient M est employé dans (2-72) tel que $M \gg 1$.

Dans le cadre de cette étude, on choisira (Figure 2-6):

$$\underline{x} = 1000, \quad \bar{x} = -1000, \quad M = 1000 \quad (2-74)$$

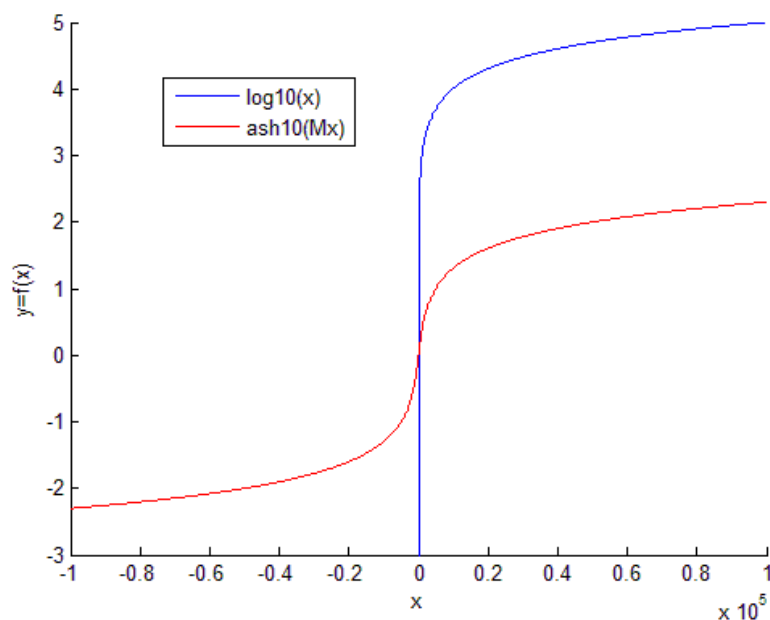


Figure 2-6: Transformation de l'espace de recherche par la fonction $\text{ash}_{10}(Mx)$

Avec $M = 1000$, on retrouve l'allure de $\log_{10}(x)$ au voisinage de $x = 0^+$, et son symétrique en $x = 0^-$ (fonction impaire).

2.2.4.3. Correcteur centralisé et décentralisé

2.2.4.3.1. Correcteur centralisé

Pour un correcteur MIMO centralisé d'ordre n_k à n_y entrées et n_u sorties sous forme tridiagonale (2-75), le nombre d'inconnues se monte à $3n_k - 2 + n_k n_y + n_k n_u + n_u n_y$

$$\left[\begin{array}{ccccc|ccc} a_{1,1} & a_{1,2} & 0 & 0 & 0 & b_{1,1} & \dots & b_{1,n_y} \\ a_{2,1} & \ddots & \ddots & 0 & 0 & b_{2,1} & \vdots & b_{2,n_y} \\ 0 & \ddots & \ddots & \ddots & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & a_{n_k-1,n_k} & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{n_k,n_k-1} & a_{n_k,n_k} & b_{n_k,1} & \dots & b_{n_k,n_y} \\ \hline c_{1,1} & c_{1,2} & \dots & \dots & c_{1,n_k} & d_{1,1} & \dots & d_{1,n_y} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n_u,1} & c_{n_u,1} & \dots & \dots & c_{n_u,n_k} & d_{n_u,1} & \dots & d_{n_u,n_y} \end{array} \right] \quad (2-75)$$

(Un cas particulier est celui du correcteur statique où le nombre d'inconnues s'établit à $n_u n_y$).

Dans ce cas, nous poserons pour vecteur de décision :

$$\begin{aligned} X &= (x_A, x_B, x_C, x_D)^T \\ x_A &= (a_{1,1}, \dots, a_{n_k,n_k}, a_{1,2}, \dots, a_{n_k-1,n_k}, a_{2,1}, \dots, a_{n_k,n_k-1}) \\ x_B &= (b_{1,1}, \dots, b_{n_k,1}, b_{1,2}, \dots, b_{n_k,2}, \dots, b_{1,n_u}, \dots, b_{n_k,n_u}) \\ x_C &= (c_{1,1}, \dots, c_{1,n_k}, c_{2,1}, \dots, c_{2,n_k}, \dots, c_{n_y,1}, \dots, c_{n_y,n_k}) \\ x_D &= (d_{1,1}, \dots, d_{1,n_u}, d_{2,1}, \dots, d_{2,n_u}, \dots, d_{n_y,1}, \dots, d_{n_y,n_u}) \end{aligned} \quad (2-76)$$

2.2.4.3.2. Correcteur décentralisé

Pour un correcteur MIMO décentralisé (2-77), le nombre d'inconnues dépend de l'ordre de chacun des sous-transferts considérés.

$$K(s) = \begin{bmatrix} K_{1,1}(s) & \dots & K_{1,n_y}(s) \\ \vdots & \ddots & \vdots \\ K_{n_u,1}(s) & \dots & K_{n_u,n_y}(s) \end{bmatrix} \quad (2-77)$$

En considérant que chaque sous-transfert est un système SISO (sous forme tridiagonale), et que l'ordre des différents transferts correspond à la matrice (2-78),

alors le nombre d'inconnues s'établit à $\sum_{j=1}^{n_y} \sum_{i=1}^{n_u} (5n_{k_{i,j}} - 1)$. Cette relation n'est valable que si aucun des termes $K_{i,j}(s)$ n'est statique auquel cas il faut remplacer $5n_{k_{i,j}} - 1$ par 1 et par 0 si on impose la nullité de $K_{i,j}(s)$.

$$\partial^\circ K = \begin{bmatrix} n_{k_{1,1}} & \cdots & n_{k_{1,n_y}} \\ \vdots & \ddots & \vdots \\ n_{k_{n_u,1}} & \cdots & n_{k_{n_u,n_y}} \end{bmatrix}, n_{k_{i,j}} \geq 1 \quad (2-78)$$

(Naturellement rien n'empêche d'imposer la nullité d'un ou plusieurs des transferts du correcteur décentralisé).

Chaque sous-terme est représenté sous sa forme tridiagonale, ce qui nous ramène au cas précédent. Dans ce cas, nous poserons pour vecteur de décision (chaque sous terme est supposé être SISO pour lisibilité) :

$$\begin{aligned} X &= \left(X_{k_{1,1}}, \dots, X_{k_{1,n_y}}, X_{k_{2,1}}, \dots, X_{k_{2,n_y}}, \dots, X_{k_{n_u,1}}, \dots, X_{k_{n_u,n_y}} \right)^T \\ X_{k_{i,j}} &= (x_A, x_B, x_C, x_D)_{i,j} \\ \left. \begin{aligned} x_A &= (a_{1,1}, \dots, a_{n,n}, a_{1,2}, \dots, a_{n-1,n}, a_{2,1}, \dots, a_{n,n-1}) \\ x_B &= (b_{1,1}, \dots, b_{n,1}) \\ x_C &= (c_{1,1}, \dots, c_{1,n}) \\ x_D &= d \end{aligned} \right\} n = n_{k_{i,j}} \end{aligned} \quad (2-79)$$

En remarquant que si :

$$H_1(s) := \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array} \right], H_2(s) := \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array} \right] \quad (2-80)$$

Alors :

$$\begin{aligned} \begin{pmatrix} H_1(s) \\ H_2(s) \end{pmatrix} &:= \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline C_1 & 0 & D_1 \\ 0 & C_2 & D_2 \end{array} \right] \\ (H_1(s) \ H_2(s)) &:= \left[\begin{array}{cc|cc} A_1 & 0 & B_1 & 0 \\ 0 & A_2 & 0 & B_2 \\ \hline C_1 & C_2 & D_1 & D_2 \end{array} \right] \end{aligned} \quad (2-81)$$

Alors il est aisé de reconstruire dans l'espace d'état le correcteur $K(s)$, connaissant la représentation d'état des sous-transferts $K_{i,j}(s)$ par l'algorithme (2-82).

pour $j=1$ à n_y colonnes de $K(s)$
 pour $i=1$ à n_u lignes de $K(s)$
 si $i=1$,
 $A_H = A_{k_{1,1}}, B_H = B_{k_{1,1}}$
 $C_H = C_{k_{1,1}}, D_H = D_{k_{1,1}}$
 sinon
 $A_H \leftarrow \begin{pmatrix} A_H & 0 \\ 0 & A_{k_{i,j}} \end{pmatrix}, B_H \leftarrow \begin{pmatrix} B_H & 0 \\ 0 & B_{k_{i,j}} \end{pmatrix}$
 $C_H \leftarrow \begin{pmatrix} C_H & C_{k_{i,j}} \end{pmatrix}, D_H \leftarrow \begin{pmatrix} D_H & D_{k_{i,j}} \end{pmatrix}$
 fin si
 fin i

construction de la
 $j^{\text{ème}}$ colonne de $K(s)$

si $j=1$,
 $A_G = A_H, B_G = B_H$
 $C_G = C_H, D_G = D_H$
 sinon
 $A_G \leftarrow \begin{pmatrix} A_G & 0 \\ 0 & A_H \end{pmatrix}, B_G \leftarrow (B_G \ B_H)^T$
 $C_G \leftarrow \begin{pmatrix} C_G & 0 \\ 0 & C_H \end{pmatrix}, D_G \leftarrow \begin{pmatrix} D_G & 0 \\ 0 & D_H \end{pmatrix}$
 fin si
 fin j

(2-82)

$$K(s) := \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array} \right]$$

2.2.4.3.3. Correcteur décentralisé et cascadié

Au lieu d'implémenter le correcteur de boucle décentralisé en feedback, il est courant de vouloir l'implémenter sous forme cascade. Cela consiste à implémenter la correction comme sur la Figure 2-7.

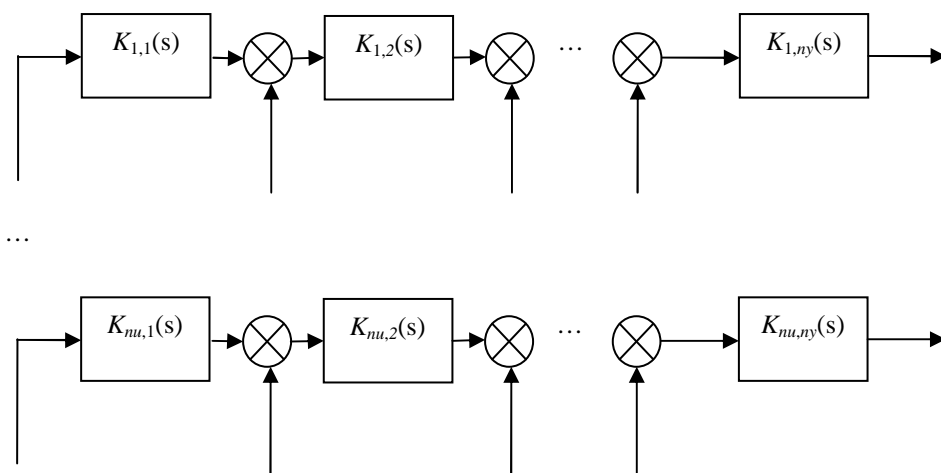


Figure 2-7: Implémentation en cascade

En utilisant le fait que :

$$H_2(s)(I + H_1(s)) := \left[\begin{array}{cc|cc} A_2 & B_2 C_1 & B_2 D_1 & B_2 \\ 0 & A_1 & B_1 & 0 \\ \hline C_2 & D_2 C_1 & D_2 D_1 & D_2 \end{array} \right] \quad (2-83)$$

Alors il est aisé de reconstruire par récurrence le correcteur $K(s)$ en cascade à partir des $K_{i,j}(s)$ qui le composent, par l'algorithme (2-84).

$$\begin{array}{l}
 \text{pour } j=1 \text{ à } n_y \text{ colonnes de } K(s) \\
 \quad \text{pour } i=n_u \text{ lignes à 1 de } K(s) \\
 \quad \quad \text{si } i=n_u, \\
 \quad \quad \quad A_H = A_{k_{i,j}}, B_H = B_{k_{i,j}} \\
 \quad \quad \quad C_H = C_{k_{i,j}}, D_H = D_{k_{i,j}} \\
 \quad \quad \text{sinon} \\
 \quad \quad \quad A_H \leftarrow \begin{pmatrix} A_H & B_H C_{k_{i,j}} \\ 0 & A_{k_{i,j}} \end{pmatrix}, B_H \leftarrow \begin{pmatrix} B_H D_{k_{i,j}} & B_H \\ B_{k_{i,j}} & 0 \end{pmatrix} \\
 \quad \quad \quad C_H \leftarrow \begin{pmatrix} C_H & D_H C_{k_{i,j}} \end{pmatrix}, D_H \leftarrow \begin{pmatrix} D_H D_{k_{i,j}} & D_H \end{pmatrix} \\
 \quad \quad \text{fin si} \\
 \quad \text{fin } i \\
 \\
 \text{si } j=1, \\
 \quad A_G = A_H, B_G = B_H \\
 \quad C_G = C_H, D_G = D_H \\
 \text{sinon} \\
 \quad A_G \leftarrow \begin{pmatrix} A_G & 0 \\ 0 & A_H \end{pmatrix}, B_G \leftarrow (B_G \ B_H)^T \\
 \quad C_G \leftarrow \begin{pmatrix} C_G & 0 \\ 0 & C_H \end{pmatrix}, D_G \leftarrow \begin{pmatrix} D_G & 0 \\ 0 & D_H \end{pmatrix} \\
 \text{fin si} \\
 \text{fin } j \\
 \\
 K(s) := \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array} \right]
 \end{array} \tag{2-84}$$

2.2.4.4. Evaluation de la fonction de coût

Nous cherchons à résoudre le problème d'optimisation suivant :

$$\begin{array}{l}
 \min f(X) \\
 X \in [x; \bar{x}]
 \end{array} \tag{2-85}$$

Où la variable X est définie dans le paragraphe 2.2.4.3 et où la fonction $f(X)$ consiste à réaliser les étapes suivantes :

- construire $\left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right]$ à partir de X et du paragraphe 2.2.4.3 ;
- construire la matrice A_{cl} de la boucle fermée suivante :

$$A_{cl} = \begin{bmatrix} A + B_2 M_1 D_k C_2 & B_2 M_1 C_k \\ B_k M_2 C_2 & A_k + B_k M_2 D_{22} C_k \end{bmatrix} \quad (2-86)$$

$$M_1 = (I - D_k D_{22})^{-1}$$

$$M_2 = (I - D_{22} D_k)^{-1}$$

– évaluer l'abscisse spectrale de la boucle fermée :

$$\bar{\lambda} = \max_i (\operatorname{Re}(A_{cl})) \quad (2-87)$$

– si $\bar{\lambda} \geq 0$ alors $f(X) = \bar{\lambda}$,

– sinon :

- construire la boucle fermée $F_l(P, K)$ suivante :

$$T(s) = F_l(P(s), K(s))$$

$$= \left[\begin{array}{cc|c} A + B_2 M_1 D_k C_2 & B_2 M_1 C_k & B_1 + B_2 M_1 D_k D_{21} \\ B_k M_2 C_2 & A_k + B_k M_2 D_{22} C_k & B_k M_2 D_{21} \\ \hline C_1 + D_{12} M_1 D_k C_2 & D_{12} M_1 C_k & D_{11} + D_{12} M_1 D_k D_{21} \end{array} \right] \quad (2-88)$$

$$M_1 = (I - D_k D_{22})^{-1}$$

$$M_2 = (I - D_{22} D_k)^{-1}$$

- calculer $\gamma = \|F_l(P, K)\|_\infty$;
- évaluer $f(X) = -\frac{1}{\gamma}$.

2.2.4.5. Benchmarking avec *COMPl_eib*

2.2.4.5.1. Introduction

La librairie *COMPl_eib* [LEI 03] est composée d'une collection d'exemples de systèmes (linéaires et invariants) pouvant être employés pour tester et comparer des algorithmes de synthèse de correcteurs basés sur différentes techniques d'optimisation telles que les *LMI*, *BMI*, l'optimisation non lisse,... Aussi décidons-nous d'employer cette librairie pour tester la synthèse H_∞ par l'optimisation stochastique formulée précédemment. Les exemples de *COMPl_eib* vont du purement académique au plus réaliste industriellement et sont répertoriés en sous-groupes :

- Modèles d'avions (AC),
- Modèles d'hélicoptère (HE),
- Modèles de moteur d'avion (JE) et de réacteurs (REA),
- Système décentralisés interconnectés (DIS),

- Modèle de type poutre d'Euler Bernouilli (EB),
- Modèle 2D de transfert de chaleur (HF2D),
- Problème de contrôle d'ordre réduit (ROC),
- Problèmes académiques (NN),
- Problèmes variés : modèle d'éoliennes (WEC),...

On emploie ici la version 1.1 de *COMPl_eib*.

Le nombre total de cas tests qui compose ce *benchmarking* est de 234 : ils sont identiques à ceux que l'on peut trouver dans [APK 10]. On rappelle en annexe C la liste des cas testés ainsi que le nombre d'itérations employées.

2.2.4.5.2. Comparaison de métaheuristiques

2.2.4.5.2.1 Mesure de performances

Les algorithmes d'optimisation testés étant stochastiques, une évaluation statistique des résultats est nécessaire pour chaque test.

Il existe de nombreuses manières de mesurer la performance d'algorithmes mais ce qui nous intéresse ici d'un point de vue pratique, c'est le comportement moyen des résultats de synthèse H_∞ (valeur minimale pour γ et répétabilité associée) sur l'ensemble des cas test de *COMPl_eib* avec les algorithmes (et leur réglage) présentés en première partie.

Ainsi la logique de mesure de performance est la suivante.

Pour l'ensemble des 234 cas test, chaque test fait l'objet de 10 *runs* successifs : nous relevons alors la valeur minimale de γ obtenue (γ_{min}), la valeur moyenne de γ (γ_{mean}), son écart-type (γ_{std}) et finalement le temps de calcul (t_{cpu}) mis pour la réalisation de ces 10 *runs*.

$$\left. \begin{aligned} \gamma_{min} &= \min_{i=1, \dots, n \text{ runs}} \gamma_n \\ \gamma_{mean} &= \frac{1}{n} \sum_{i=1}^{n \text{ runs}} \gamma_i \\ \gamma_{std} &= \sqrt{\frac{1}{n} \sum_{i=1}^{n \text{ runs}} (\gamma_i - \gamma_{mean})^2} \end{aligned} \right\} j^{\text{ème}} \text{ cas test} \quad (2-89)$$

Ainsi pour chaque test (par exemple AC1, AC2,...), chaque algorithme peut être classé par un rang vis-à-vis des grandeurs (2-89).

Lorsque tous les tests sont réalisés, on établit alors un rang « moyen » pour chaque algorithme vis-à-vis des grandeurs (2-89) à partir des rangs obtenus pour chaque test (Figure 2-8).

Nous considérons que l'algorithme disposant du rang moyen le plus bas est finalement le plus performant vis-à-vis de la grandeur considérée de (2-89) pour le problème de la synthèse H_{∞} .

La mesure du temps de calcul est ici intéressante car :

- Certains algorithmes disposent d'une phase de recherche locale associée à une phase de recherche globale (ABC, CS) ; ainsi lors d'une itération, on peut évaluer la fonction de coût un nombre de fois plus important que pour des approches plus directes (PSO, QPSO, DE).
- La fonction de coût (2-64) se décompose en deux parties : la stabilisation et la performance. Or ces deux phases ne disposent pas du même temps d'exécution (la seconde prend plus de temps).

Ainsi à nombres d'itérations égaux, les temps de calcul employés par chaque algorithme pour évaluer chaque cas test peuvent être très disparates.

Pour chaque test, le nombre d'itérations choisi (empiriquement) est identique pour tous les algorithmes testés : il est donné en annexe C pour l'ensemble des tests.

La totalité du test a été réalisée sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

2.2.4.5.2.2 Performance globale

On présente ici les performances obtenues sur la globalité du test.

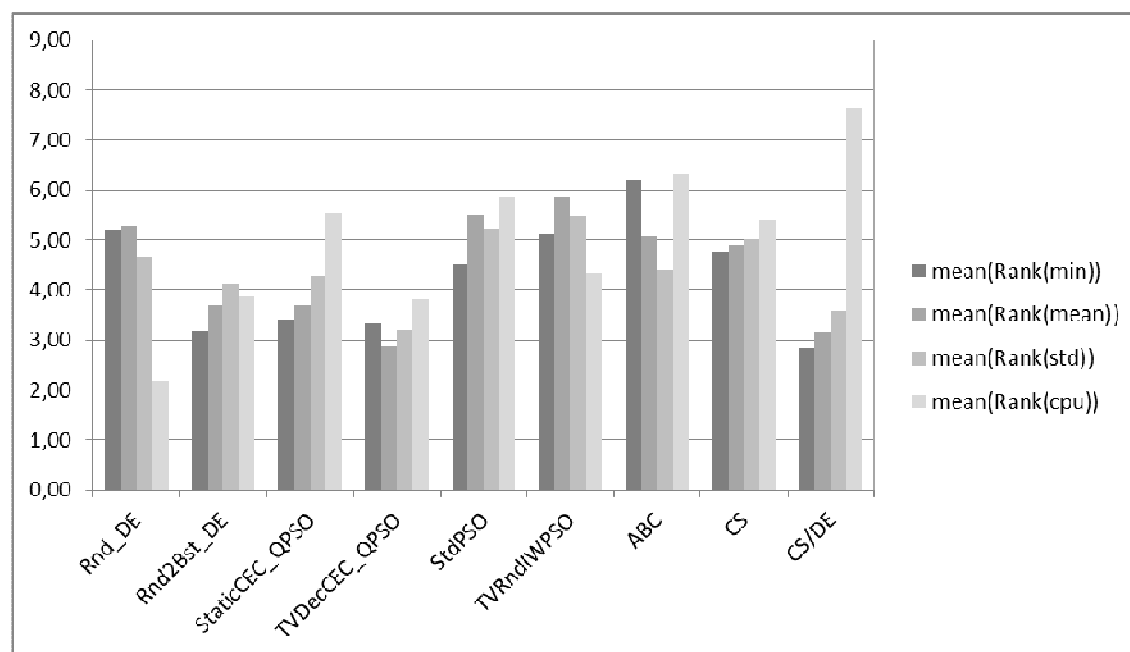


Figure 2-8: Rangs moyens des différents algorithmes sur la globalité du test (234 tests)

2.2.4.5.2.3 Discussion

On constate que le meilleur algorithme est *CS/DE* toutefois au prix de temps de calcul prohibitifs, ce qui est dû à la structure même de l'algorithme : une phase d'exploration locale et une phase d'exploration globale donnant lieu à un nombre conséquent d'évaluations de la fonction de coût. On préférera *Rnd2Bst_DE* qui est le second en termes de valeur minimale atteinte et du même ordre de grandeur pour les moyennes et écarts-type : l'algorithme est plutôt bien classé en termes de temps de calcul.

Les algorithmes à essais particuliers quantiques de type QPSO donnent des résultats intéressants mais moins bons pour la valeur minimale atteinte.

Les autres algorithmes sont moins compétitifs pour le problème de la synthèse H_∞ à ordre fixé vis-à-vis des critères (2-89).

2.2.4.5.3. Comparaison avec l'optimisation non-lisse

On retrouve dans [APK 10] un comparatif similaire au précédent entre *Hifoo* et *Hinfstruct*. Aussi décidons-nous de comparer *Rnd2Bst_DE* avec les outils d'optimisation non lisse précités sur un échantillon exhaustif de cas tests. La comparaison porte sur la valeur minimale atteinte et le temps de calcul (n'ayant pas accès à la moyenne et l'écart-type de résultats issues *Hifoo* et *Hinfstruct*). Comme pour l'optimisation stochastique, 10 runs sont lancés pour chaque test avec les outils d'optimisation non lisse.

Test	∂° (P,K)	HinfStruct		Hifoo		Evolutionary	
		γ	cpu (min)	γ	cpu (min)	γ	cpu (min)
AC1	(5,0)	0,00	0,16	0,00	1,99	0,00	2,40
AC1	(5,2)	0,00	0,22	0,00	3,23	0,00	1,90
AC2	(5,0)	0,11	0,04	0,11	0,34	0,11	1,03
AC2	(5,2)	0,11	0,02	0,11	0,30	0,11	1,70
AC3	(5,0)	3,42	0,05	3,66	0,48	3,57	3,20
AC3	(5,2)	2,98	0,10	3,12	1,54	3,51	12,92
AC4	(4,0)	0,94	0,04	0,94	0,20	0,95	0,18
AC4	(4,2)	0,56	0,05	0,56	0,87	0,56	8,52
AC5	(4,0)	665	0,03	669	0,05	2366	0,04
AC5	(4,2)	665	0,05	665	0,70	682	6,82
AC6	(7,0)	4,11	0,04	4,11	1,35	4,11	3,64
AC6	(7,4)	3,46	0,24	3,55	7,32	3,83	26,79
AC7	(9,0)	0,07	0,02	0,07	0,06	0,07	0,26
AC7	(9,5)	0,04	0,18	0,04	4,33	0,04	23,15
AC8	(9,0)	2,01	0,03	2,01	1,08	2,02	1,25
AC8	(9,5)	1,62	0,13	1,62	6,61	1,66	24,25
AC9	(10,0)	1,00	0,12	1,00	2,57	1,01	12,04

AC9	(10,5)	1,00	0,24	1,01	5,20	1,19	24,86
AC9	(10,8)	1,00	0,33	1,05	3,31	1,34	50,75
AC10	(55,0)	13,24	0,34	Inf	0,58	13,6	2,46
AC10	(55,3)	7,92	2,41	Inf	4,09	14	134
AC10	(55,10)	7,90	4,57	20,6	44,20	16,0	374
AC10	(55,20)	7,69	5,81	Inf	17,03	19,2	163,3
AC12	(4,2)	0,00	0,18	0,29	3,19	0,32	9,16
AC13	(28,0)	163	0,20	163	15,03	164	26,99
AC13	(28,7)	156	0,46	156	28,96	156	28,28
AC14	(40,0)	102	0,16	102	5,19	102	17,63
AC14	(40,2)	100	0,19	100	3,02	100	8,40
AC15	(4,0)	15,2	0,03	15,2	1,00	15,2	1,58
AC15	(4,2)	14,9	0,05	14,9	1,20	15,2	4,18
AC16	(4,0)	14,9	0,03	14,9	0,39	14,8	1,54
AC16	(4,2)	14,9	0,04	14,9	0,74	14,8	2,08
AC17	(4,0)	6,61	0,01	6,61	0,01	6,61	0,67
AC17	(4,2)	6,61	0,01	6,61	0,02	6,61	1,99
AC18	(10,0)	10,70	0,06	12,7	0,75	11,7	1,03
AC18	(10,2)	7,12	0,11	7,20	1,73	7,66	8,85
HE1	(4,0)	0,15	2,64	0,15	3,98	0,15	0,29
HE1	(4,2)	0,08	4,31	0,08	95,30	0,12	10,06
HE2	(4,0)	3,91	1,90	4,00	8,88	4,40	0,84
HE2	(4,2)	2,42	4,45	2,43	76,44	2,83	33,25
HE3	(8,0)	0,81	4,10	0,81	146,96	0,82	40,95
HE3	(8,4)	0,80	9,11	0,81	270,37	0,83	55,74
HE4	(8,0)	22,8	5,58	22,8	122,48	23,9	45,28
HE4	(8,4)	22,8	5,26	22,8	209,73	27,9	49,09
HE5	(8,0)	8,90	1,80	8,90	88,43	8,86	4,20
HE5	(8,2)	2,16	7,90	2,17	195,25	2,27	46,30
HE6	(20,0)	192,4	10,4	192,4	294,61	193,9	109,3
HE6	(20,2)	15,5	48,6	16,4	724,23	85,3	139,8
HE7	(20,0)	192,4	18,7	192,4	251,34	200,6	113,7
HE7	(20,8)	2,62	79,6	10,1	594,84	28,7	109,1
JE1	(30,0)	10,13	0,52	23,49	7,24	23,32	19,04
JE1	(30,2)	4,10	1,09	19,65	16,26	5,41	142,43
JE1	(30,8)	3,92	1,84	111,4	1,50	5,62	118,74
JE2	(21,0)	183,4	0,32	183,4	14,02	182	14,19
JE2	(21,2)	72,86	0,81	97,91	20,29	139,6	82,53
JE2	(21,8)	58,58	1,38	80,47	20,88	134,1	57,58
JE3	(24,0)	5,10	0,15	5,10	7,99	5,26	5,74
JE3	(24,2)	2,90	0,47	2,90	11,61	3,41	32,97
JE3	(24,8)	2,89	0,43	3,20	7,86	3,25	64,21

REA1	(4,0)	0,87	0,03	0,89	0,07	0,87	1,43
REA1	(4,2)	0,86	0,02	0,86	0,15	0,87	1,88
REA2	(4,0)	1,15	0,03	1,15	1,35	1,16	0,62
REA2	(4,2)	1,13	0,02	1,13	0,16	1,13	1,77
REA3	(12,0)	74,25	0,02	74,25	0,07	74,25	1,85
REA3	(12,2)	74,25	0,02	74,25	0,11	74,25	4,65
REA3	(12,5)	74,25	0,03	74,25	0,06	74,25	4,44
DIS1	(8,0)	4,16	0,06	4,16	0,86	4,19	5,15
DIS1	(8,4)	4,16	0,07	4,16	1,82	4,18	29,64
DIS2	(3,0)	1,06	0,02	1,03	0,23	1,14	1,29
DIS2	(3,2)	0,95	0,04	0,95	0,71	1,03	18,46
DIS3	(6,0)	1,06	0,08	1,06	1,27	1,07	11,10
DIS3	(6,2)	1,05	0,08	1,05	1,69	1,06	24,65
DIS4	(6,0)	0,74	0,04	0,74	0,81	0,74	5,32
DIS4	(6,2)	0,73	0,05	0,74	1,27	0,74	11,76
DIS5	(4,0)	1035	0,04	1035	0,26	1119	0,43
DIS5	(4,2)	667	0,08	668	1,73	745	8,86
EB1	(10,0)	3,12	0,04	3,12	0,04	3,10	0,10
EB1	(10,4)	3,10	0,04	3,10	0,15	3,09	2,79
EB2	(10,0)	2,02	0,03	2,02	0,01	2,02	0,09
EB2	(10,4)	1,78	0,06	1,78	1,47	1,81	2,76
EB3	(10,0)	2,06	0,03	2,06	0,02	2,05	0,08
EB3	(10,2)	1,87	0,04	1,87	0,65	1,87	3,19
NN1	(3,0)	Inf	0,05	14,89	0,05	14,76	0,11
NN1	(3,2)	13,13	0,06	13,47	2,19	13,99	1,62
NN2	(2,0)	2,22	0,02	2,22	0,01	2,21	0,07
NN2	(2,2)	1,76	0,03	1,76	0,29	1,76	1,73
NN3	(4,0)	Inf	0,02	Inf	0,00	Inf	0,00
NN3	(4,2)	20,5	0,08	20,5	0,57	Inf	0,20
NN4	(4,0)	1,36	0,04	1,36	0,49	1,40	0,98
NN4	(4,2)	1,29	0,05	1,29	0,62	1,34	1,87
NN5	(7,0)	266,5	0,02	266,5	0,11	267	0,12
NN5	(7,4)	238,5	0,09	238,8	1,80	241,6	22,20
NN6	(9,0)	5602	0,04	5602	2,00	5640	1,03
NN6	(9,2)	171,2	0,32	221,5	6,89	220,6	11,20
NN7	(9,0)	74,1	0,03	74,1	0,96	78,1	0,91
NN7	(9,5)	37,4	0,42	39,7	6,77	41,9	29,43
NN8	(3,0)	2,89	0,03	2,89	0,27	2,94	1,13
NN8	(3,3)	2,36	0,03	2,36	0,28	2,42	1,30
NN9	(5,0)	28,7	0,05	28,9	0,73	28,7	0,98
NN9	(5,3)	13,7	0,13	14,1	3,07	21,8	17,41
NN11	(16,0)	0,09	0,06	0,10	0,04	0,09	3,26

NN11	(16,2)	0,02	0,28	0,05	1,87	0,06	13,14
NN12	(6,0)	17,3	0,06	Inf	0,19	21,7	0,10
NN12	(6,2)	7,43	0,20	16,2	2,08	16,1	16,07
NN13	(6,0)	14,1	0,05	14,1	0,83	18,9	0,73
NN13	(6,2)	10,5	0,08	11	2,07	12,9	8,52
NN14	(6,0)	17,5	0,04	17,5	0,69	18,3	0,60
NN14	(6,2)	9,47	0,12	9,54	2,22	10,5	15,76
NN15	(3,0)	0,10	0,02	0,10	0,01	0,10	1,10
NN15	(3,2)	0,10	0,03	0,10	0,13	0,10	1,98
NN16	(8,0)	0,96	0,07	0,96	2,26	0,96	2,24
NN16	(8,2)	0,96	0,08	0,96	5,77	0,96	12,88
NN17	(3,0)	11,2	0,01	11,2	0,13	11,2	0,37
NN17	(3,2)	2,76	0,04	5,04	0,87	5,16	1,36
HF2D10	(5,0)	8,02E+04	0,06	8,00E+04	2,08	8,11E+04	1,55
HF2D10	(5,2)	7,96E+04	0,06	7,96E+04	1,85	7,98E+04	4,90
HF2D11	(5,0)	7,73E+04	0,02	7,72E+04	1,01	7,72E+04	2,31
HF2D11	(5,2)	7,65E+04	0,04	7,65E+04	2,30	7,66E+04	4,86
HF2D12	(5,0)	1,04E+06	0,01	1,04E+06	0,13	1,04E+06	2,26
HF2D12	(5,2)	1,04E+06	0,02	1,04E+06	0,12	1,04E+06	5,03
HF2D13	(5,0)	1,02E+05	0,01	1,02E+05	0,07	1,02E+05	2,41
HF2D13	(5,2)	1,02E+05	0,02	1,02E+05	0,10	1,02E+05	5,03
ROC1	(9,1)	1,24	0,07	1,24	1,63	1,24	3,64
ROC1	(9,5)	1,14	0,17	1,14	4,79	1,29	14,45
ROC2	(10,1)	0,05	0,15	0,05	5,09	0,61	3,62
ROC2	(10,5)	0,04	0,33	0,04	5,45	0,15	11,32
ROC3	(11,1)	228,2	0,57	4281	0,11	Inf	7,04
ROC3	(11,3)	185,8	0,73	5730	2,74	Inf	8,83
ROC4	(9,1)	302,2	0,04	302	1,69	302,7	2,60
ROC4	(9,3)	155,3	0,24	266	4,19	304,3	6,87
ROC5	(7,1)	0,00	0,34	0,00	0,78	0,01	3,40
ROC5	(7,3)	0,00	0,44	0,00	2,58	0,01	6,65
WEC1	(10,0)	4,05	0,07	4,05	1,82	6,07	6,07
WEC1	(10,2)	3,64	0,05	3,64	1,85	3,64	15,42
WEC1	(10,4)	3,64	0,04	3,64	1,65	3,64	37,51
WEC1	(10,9)	3,64	0,08	3,64	2,80	3,64	67,85
WEC2	(10,0)	4,25	0,08	4,25	2,28	5,98	6,16
WEC2	(10,2)	3,60	0,04	3,60	1,99	3,60	15,42
WEC2	(10,4)	3,60	0,06	3,60	2,96	3,60	31,96
TG1	(10,0)	12,85	0,04	12,85	1,00	12,86	1,71
TG1	(10,2)	3,47	0,09	3,47	4,16	5,46	5,31

TG1	(10,4)	3,47	0,09	3,47	6,28	3,47	8,98
AGS	(12,0)	8,17	0,03	8,17	0,70	8,17	2,75
AGS	(12,2)	8,17	0,04	8,17	1,07	8,17	4,59
AGS	(12,4)	8,17	0,07	8,17	1,50	8,17	3,35
TMD	(6,0)	2,23	0,07	2,53	1,12	2,70	1,74
PAS	(5,0)	0,00	0,09	40,1	0,02	1e6	0,13
PAS	(5,2)	0,00	0,18	0,01	0,26	0,13	3,69
IH	(21,8)	1,01	2,25	1,81	5,89	8,59	325,5
IH	(21,15)	1,03	9,63	1,81	5,70	14,24	237,8

Tableau 2.1 Comparatif optimisation non-lisse / évolutionnaire

On constate que :

- *Hinfstruct* est l'outil le plus rapide et c'était déjà la conclusion de l'article [APK 10] ; l'optimisation évolutionnaire se caractérise par des temps de calcul comparables à ceux de *Hifoo* ; noter que ce n'est pas là l'objectif premier de l'approche évolutionnaire car ce qui nous intéresse ici c'est la flexibilité de l'approche comme nous allons le voir plus bas.
- L'optimisation évolutionnaire et l'optimisation non lisse donnent des résultats comparables sur la valeur de γ atteinte. Sur quelques tests, l'optimisation évolutionnaire donne même de meilleurs résultats (en vert sur le tableau précédent) et parvient même à trouver une solution là où l'optimisation non lisse ne converge pas (NN1 (3,0)). En ce sens, cela peut constituer une alternative intéressante, dont la solution pourrait initialiser l'approche non-lisse.

2.2.4.6. Prise en compte de contraintes supplémentaires

Cette approche de la synthèse structurée (beaucoup plus simple que l'optimisation non lisse puisqu'il s'agit uniquement d'évaluer la fonction de coût et non son gradient) permet d'étendre simplement le cadre de la synthèse. Les possibilités sont multiples et nombreuses comme on le montre ci-après.

2.2.4.6.1. Contrainte sur la stabilité du correcteur

On peut chercher à résoudre le problème suivant : trouver un correcteur $K(s)$ de représentation d'état $\Theta_k = \begin{pmatrix} A_K & B_K \\ C_K & D_K \end{pmatrix}$, et d'ordre $n_k < n_p$ tel que :

- $T(s)$ soit stable de manière interne ;
- $\|T(s)\|_\infty$ soit minimale ;
- le correcteur $K(s)$ soit stable.

En posant :

$$\bar{\lambda}_k = \max_i (\operatorname{Re}(\lambda_i(A_k))) \quad (2-90)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Alors on synthétise le correcteur en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_k} f(\Theta_k) \\ & \text{si } \max(\bar{\lambda}_{cl}, \bar{\lambda}_k) \geq 0 \\ & \quad f(\Theta_k) = \max(\bar{\lambda}_{cl}, \bar{\lambda}_k) \\ & \text{sinon} \\ & \quad f(\Theta_k) = -\frac{1}{\|T(s)\|_\infty} \end{aligned} \quad (2-91)$$

2.2.4.6.2. Contrainte sur la position des pôles en boucle fermée

On peut chercher à résoudre le problème suivant : trouver un correcteur $K(s)$ de représentation d'état $\Theta_k = \begin{pmatrix} A_K & B_K \\ C_K & D_K \end{pmatrix}$, et d'ordre $n_k < n_p$ tel que :

- $T(s)$ soit stable de manière interne ;
- $\|T(s)\|_\infty$ soit minimal ;
- les pôles de $T(s)$ soient tous à partie réelle inférieure à un nombre strictement négatif α .

Alors on synthétise le correcteur H_∞ -optimal en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_k} f(\Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq \alpha \\ & \quad f(\Theta_k) = \bar{\lambda}_{cl} - \alpha \\ & \text{sinon} \\ & \quad f(\Theta_k) = -\frac{1}{\|T(s)\|_\infty} \end{aligned} \quad (2-92)$$

2.2.4.7. Extension à d'autres problèmes complexes

2.2.4.7.1. Synthèse H_∞ structurée multi-modèles

2.2.4.7.1.1 Principe

Comme nous le proposons dans [FEY 14a] rien n'empêche de généraliser la synthèse H_∞ structurée présentée précédemment au cadre multi-modèle. Pour cela, on considère l'ensemble de m modèles suivant :

$$\mathcal{P} = \{P_i(s), i = 1, \dots, m\} \quad (2-93)$$

Comme sur la Figure 2-9, le problème H_∞ structuré multi-modèles consiste à déterminer un correcteur $K(s)$ d'ordre n_k qui :

- stabilise chaque boucle fermée $F_l(P_i, K)$ manière interne ;
- minimise γ tel que :

$$\max_{i=1, \dots, m} \|F_l(P_i, K)\|_\infty < \gamma \quad (2-94)$$

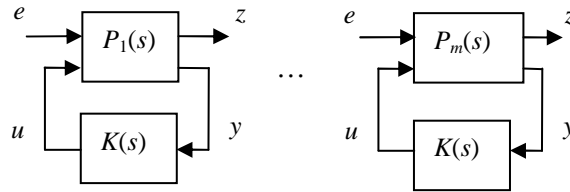


Figure 2-9: La synthèse H_∞ multi-modèles

La $i^{\text{ème}}$ forme standard dispose de la représentation d'état suivante :

$$P_i(s) := \left[\begin{array}{c|cc} A_i & B_{1i} & B_{2i} \\ \hline C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{array} \right] \quad (2-95)$$

La représentation d'état du $i^{\text{ème}}$ système bouclé par $K(s)$ s'écrit :

$$\begin{aligned} T_i(s) &= F_l(P_i(s), K(s)) \\ &= \left[\begin{array}{cc|c} A_i + B_{2i}M_{1i}D_kC_{2i} & B_{2i}M_{1i}C_k & B_{1i} + B_{2i}M_{1i}D_kD_{21i} \\ B_kM_{2i}C_{2i} & A_k + B_kM_{2i}D_{22i}C_k & B_kM_{2i}D_{21i} \\ \hline C_{1i} + D_{12i}M_{1i}D_kC_{2i} & D_{12i}M_{1i}C_k & D_{11i} + D_{12i}M_{1i}D_kD_{21i} \end{array} \right] \quad (2-96) \\ M_{1i} &= (I - D_kD_{22i})^{-1} \\ M_{2i} &= (I - D_{22i}D_k)^{-1} \end{aligned}$$

On définit l'abscisse spectrale de la boule de modèles « bouclés par $K(s)$ » :

$$\bar{\lambda}_{cl} = \max_{j=1,\dots,m} \max_i \left(\operatorname{Re}(\lambda_i(A_{clj})) \right) \quad (2-97)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Alors on synthétise un correcteur stabilisant H_∞ -optimal dans le cadre multi-modèle en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_k} f(\Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq 0 \\ & \quad f(\Theta_k) = \bar{\lambda}_{cl} \\ & \text{sinon} \\ & \quad f(\Theta_k) = -\frac{1}{\gamma}, \quad \gamma = \max_{j=1,\dots,m} \|T_j(s)\|_\infty \end{aligned} \quad (2-98)$$

L'adaptation de l'algorithme du paragraphe 2.2.4.4 est alors immédiate.

Cette méthode de synthèse de correcteur amène quelques observations :

- On retrouve strictement l'algorithme du paragraphe 2.2.4.4 en prenant $m=1$,
- aucune hypothèse n'a été faite quant aux ordres des $P_i(s)$ qui peuvent être différents les uns des autres,
- notre approche n'empêche pas chaque $P_i(s)$ de disposer de canaux exogènes z_i différents. De même pour les sorties e_i ,
- comme pour la synthèse « mono-modèle », rien n'empêche de rajouter les contraintes sur la stabilité du correcteur, ainsi que sur la position des pôles de l'ensemble des boucles fermées.

2.2.4.7.1.2 Exemple

A titre d'exemple, considérons le système de la Figure 2-10. Il s'agit d'un système à mode flexible connecté à un moteur à courant continu.

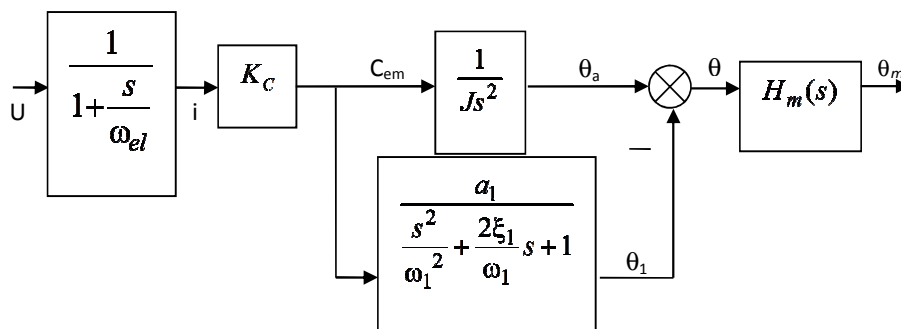


Figure 2-10: Système à mode flexible motorisé

Les valeurs des paramètres, sujets à des incertitudes, sont données ci-après.

$$\begin{aligned}
 J &= 0.0014 \text{ kg.m}^2, K_c = 0.06 \text{ Nm.A}^{-1}, \\
 a_1 &= -7.10^{-4} \pm 7\%, \xi_1 = 0.029 \pm 70\%, \\
 \omega_1 &= 1\,335 \text{ s}^{-1} \pm 10\%, \omega_{el} = 1\,257 \text{ s}^{-1} \pm 20\%
 \end{aligned} \tag{2-99}$$

La position angulaire θ est mesurée par l'intermédiaire d'un capteur de fonction de transfert :

$$H_m(s) = \frac{1}{\frac{s^2}{1382^2} + \frac{s}{1382} + 1} \tag{2-100}$$

Le modèle nominal qui correspond aux valeurs nominales des paramètres est noté $H_0(s)$.

On se propose de calculer un correcteur assurant la spécification suivante :

- bande passante supérieure à 40 Hz ;
- marge de gain supérieure à 5 dB ;
- marge de phase supérieure à 40° ;
- pas d'erreur statique pour une perturbation en échelon en entrée de modèle ;
- stabiliser le système vis-à-vis de ses incertitudes paramétriques.

On décide de synthétiser le correcteur en utilisant l'approche loop-shaping de la synthèse H_∞ . On rappelle que l'approche consiste à :

- 1) modéliser la boucle ouverte sous la forme d'un loop-shape $H_p(s)$ en fonction des spécifications, en employant des pondérations en entrée et en sortie du système :

$$H_p(s) = W_o(s)H_0(s)W_i(s) \tag{2-101}$$

En posant :

$$H_0(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right], \quad W_o(s) := \left[\begin{array}{c|c} A_o & B_o \\ \hline C_o & D_o \end{array} \right], \quad W_i(s) := \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array} \right] \tag{2-102}$$

On montre facilement que :

$$H_p(s) := \left[\begin{array}{c|c} A_p & B_p \\ \hline C_p & D_p \end{array} \right] = \left[\begin{array}{ccc|c} A & BC_i & 0 & BD_i \\ 0 & A_i & 0 & B_i \\ \hline B_o C & B_o DC_i & A_o & B_o DD_i \\ \hline D_o C & D_o DC_i & C_o & D_o DD_i \end{array} \right] \tag{2-103}$$

- 2) déterminer le précorrecteur $K_p(s)$ stabilisant $H_p(s)$ en résolvant le problème H_∞ 4-blocs⁸ de la Figure 2-11 et de l'équation (2-104)

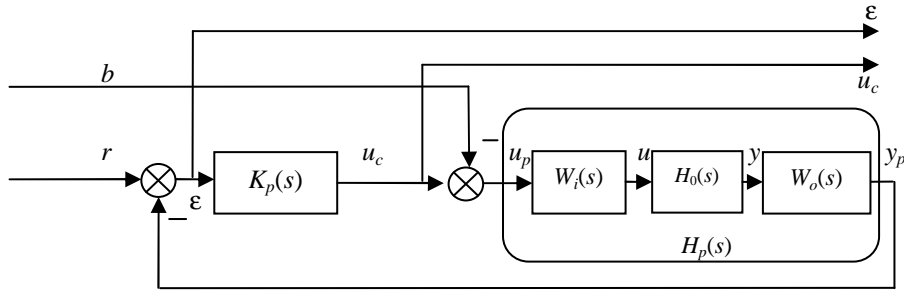


Figure 2-11: Synthèse H_∞ par loop-shaping

$$K_p(s) = \arg \min \gamma, \left\| \begin{pmatrix} S_p & S_p H_p \\ K_p S_p & K_p S_p H_p \end{pmatrix} \right\|_\infty < \gamma, \quad S_p = (I + K_p H_p)^{-1} \quad (2-104)$$

La forme standard du problème considéré s'écrit :

$$\begin{pmatrix} z \\ \varepsilon \end{pmatrix} = P(s) \begin{pmatrix} e \\ u_c \end{pmatrix}, \quad e = (r, b)^T, \quad z = (\varepsilon, u_c)^T \quad (2-105)$$

On montre facilement que :

$$P(s) := \begin{bmatrix} A_p & 0 & -B_p & B_p \\ -C_p & I & D_p & -D_p \\ 0 & 0 & 0 & I \\ -C_p & I & D_p & -D_p \end{bmatrix} \quad (2-106)$$

- 3) calculer le correcteur final $K(s)$ à partir des pondérations et du précorrecteur

$$K(s) = W_i(s) K_p(s) W_o(s) \quad (2-107)$$

Choix des pondérations

Le réglage des pondérations se fait par les règles de l'automatique fréquentielle classique. Ainsi au regard de la spécification, le loop-shape doit :

- disposer d'une composante intégrale afin d'annuler l'erreur statique en réponse à un échelon de perturbation,
- disposer d'un réseau avance de phase afin d'assurer les marges de stabilité,

⁸ On peut aussi résoudre (ce qui est équivalent) le problème de stabilisation d'un système modélisé par ses factorisations premières

- disposer d'une bande passante d'au moins 40 Hz,
- disposer d'un roll-off suffisant pour stabiliser le système soumis aux incertitudes paramétriques : ici le roll-off est choisi pour satisfaire le théorème du petit gain [ZHO 96] vis-à-vis d'une incertitude multiplicative $\Delta(s)$ introduite conformément à la Figure 2-12 pour modéliser les incertitudes hautes fréquences dues aux incertitudes sur les paramètres.

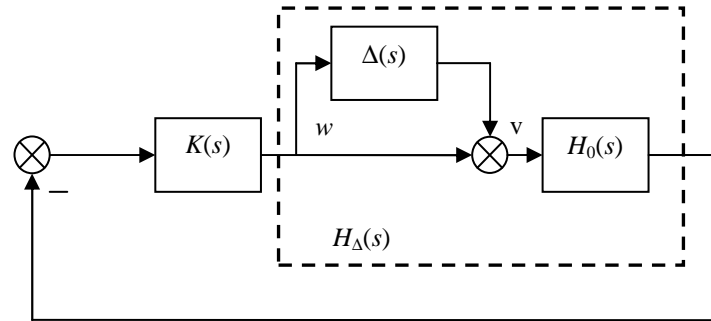


Figure 2-12: Incertitude multiplicative

En effet d'après la Figure 2-12 :

$$w(s) = T(s)v(s), \quad v(s) = \Delta(s)w(s) \quad (2-108)$$

Ainsi en vertu du théorème du petit gain, le système bouclé est stable si et seulement si (2-109) est satisfaite.

$$\forall \omega \quad |T(j\omega)| < \left| \frac{1}{\Delta(j\omega)} \right| \quad (2-109)$$

Ainsi aux fréquences où $|\Delta(j\omega)|$ est grand, la boucle ouverte sera de faible gain pour assurer la stabilité, et dans ce cas $|T(j\omega)| \approx |H_{BO}(j\omega)|$ ce qui permet donc de spécifier le *roll-off* nécessaire à conférer à la boucle ouverte pour satisfaire à l'objectif de stabilité :

$$|H_{BO}(j\omega)| < \left| \frac{1}{\Delta(j\omega)} \right| \quad (2-110)$$

L'objectif de robustesse en stabilité peut donc être traduit directement sur le *loop-shape* nominal $H_p(s) = W_o(s)H_0(s)W_i(s)$ par :

$$|H_p(j\omega)| < \left| \frac{1}{\Delta(j\omega)} \right| \quad (2-111)$$

En prenant les valeurs minimale, nominale et maximale des 4 paramètres incertains⁹, on crée une boule de $3^4 = 81$ modèles $H_{\Delta}(s)$ centrée sur $H_0(s)$, dont les réponses fréquentielles sont données sur la Figure 2-13 :

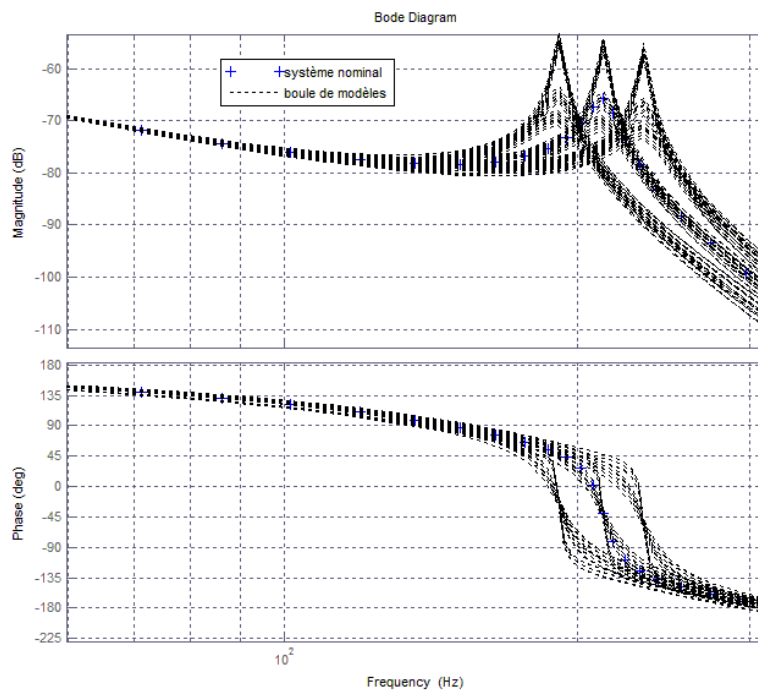


Figure 2-13: Réponse fréquentielle des différents modèles

Or d'après la Figure 2-12 :

$$H_{\Delta}(s) = (1 + \Delta(s))H_0(s) \Leftrightarrow \Delta(s) = \frac{H_{\Delta}(s) - H_0(s)}{H_0(s)} \quad (2-112)$$

L'ensemble des réponses fréquentielles de $\Delta(s)$ correspondant est donné sur la Figure 2-14 :

$$^9 \begin{cases} a_1 = \{-7,49 \cdot 10^{-4}; -7 \cdot 10^{-4}; -6,51 \cdot 10^{-4}\} \\ \xi_1 = \{0,0087; 0,029; 0,0493\} \\ \omega_1 = \{1201,5; 1335; 1468,5\} \\ \omega_{el} = \{1005,6; 1257; 1508,4\} \end{cases}$$

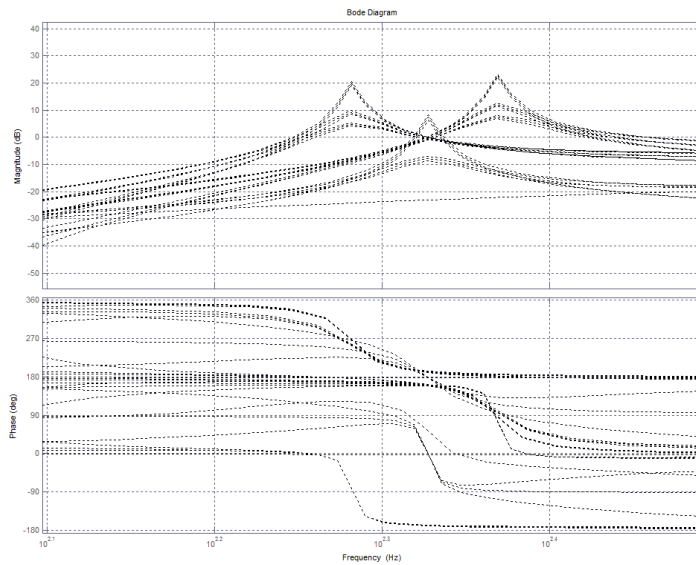


Figure 2-14: Réponses fréquentielles des incertitudes correspondantes

On se propose de choisir le réglage de pondérations suivant :

$$W_o(s) = \frac{128.5(s + 81.24)}{s} \frac{\left(1 + \frac{s}{5.03}\right)}{\left(1 + \frac{s}{1257}\right)}$$

$$W_i(s) = \left(\frac{\frac{s^2}{942^2} + \frac{0.25s}{942} + 1}{\frac{s^2}{942^2} + \frac{1.8s}{942} + 1} \right) \left(\frac{\frac{s^2}{2200^2} + \frac{0.01s}{2200} + 1}{\frac{s^2}{2200^2} + \frac{2.2s}{2200} + 1} \right) \frac{1}{\left(\frac{s}{942} + 1\right)} \quad (2-113)$$

On vérifie sur la Figure 2-15 que les spécifications en termes de stabilité et de bande passante sont satisfaites.

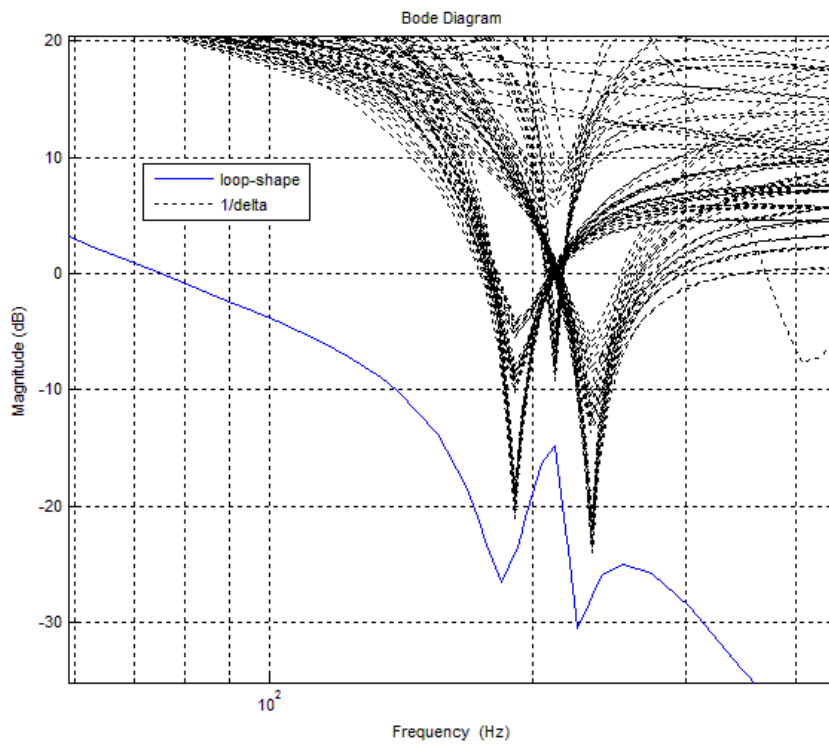


Figure 2-15: Loop-shape et vérification de la condition (2-111)

On retrouvera cet exemple davantage détaillé dans [FEY 13c].

Limites de la synthèse H_{∞} d'ordre plein sur le modèle nominal

On procède à une synthèse d'ordre plein et on trouve $\gamma = 3.30$. On vérifie sur la Figure 2-16 que les marges de stabilité sont satisfaisantes pour le modèle nominal.

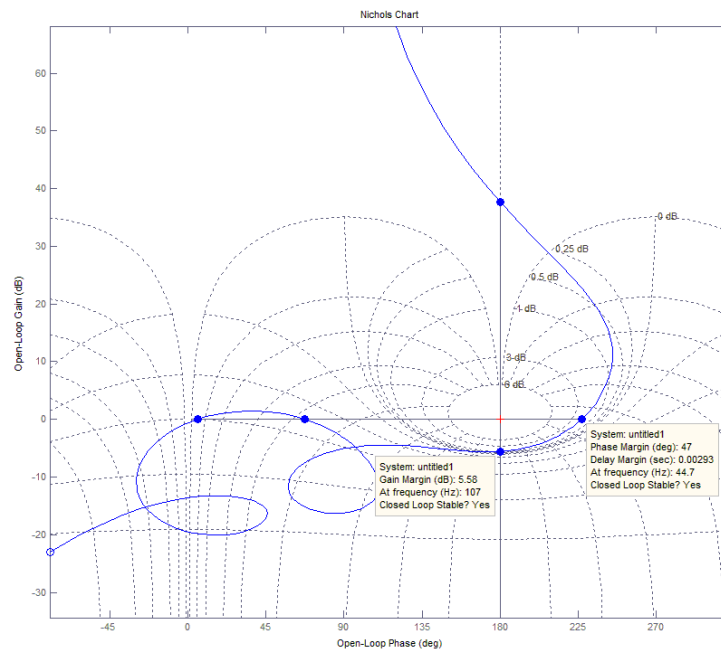


Figure 2-16: Boucle ouverte nominale

Cependant cette synthèse souffre de deux inconvénients majeurs :

- tout d’abord, on trace l’ensemble des réponses fréquentielles en boucle ouverte relatif à la boule de modèles sur la Figure 2-17. On constate que bien que toute la boule soit stabilisée, il est des systèmes pour lesquels les marges de stabilité s’avèrent très réduites : la valeur de γ « pire cas » peut atteindre 10,5.
- enfin, le précorrecteur est d’ordre 15 et le correcteur final d’ordre 23 ce qui peut paraître important au regard de la faible robustesse en marges de stabilité du correcteur calculé.

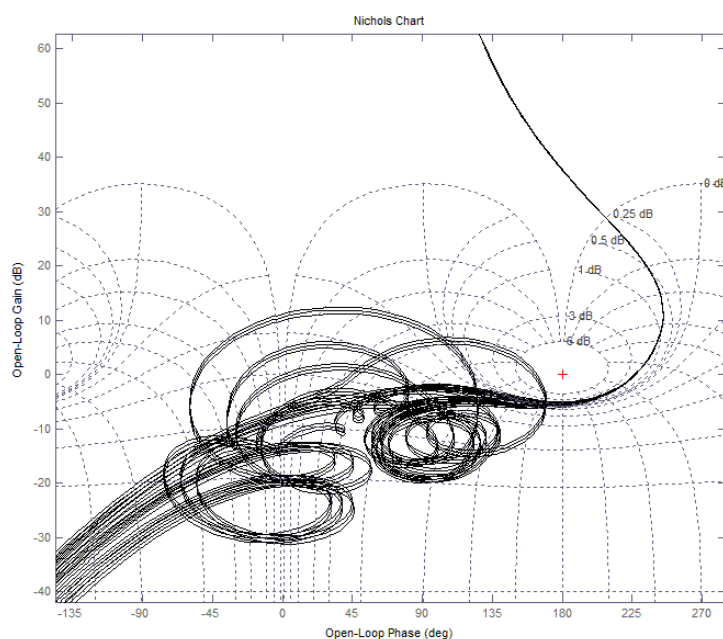


Figure 2-17: Boucles ouvertes sur l'ensemble de la boule de modèles

Synthèse H_∞ à ordre fixé

Dans un premier temps, on décide de synthétiser un précorrecteur d'ordre 5 par l'optimisation stochastique puis de comparer avec l'optimisation non lisse, toujours sur le modèle nominal.

Parce que DE est un algorithme stochastique, on procède à 10 *runs* successifs. Le nombre maximum d'itérations pour chaque *run* est de 250.

Pour l'approche non lisse, nous utilisons *Hinfstruct*. Les résultats fournis par *Hinfstruct* dépendent du point d'initialisation : ainsi on procède aussi à 10 *runs* en positionnant l'option *randomstream* à 10. De plus comme pour l'approche stochastique, le correcteur est recherché sous forme tridiagonale. Les autres options de *Hinfstruct* sont positionnées par défaut.

Les calculs sont réalisés avec Matlab 2011b sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

Une évaluation statistique des résultats est nécessaire pour chaque test. Ainsi comme en (2-89), nous relevons la valeur minimale de γ obtenue (γ_{\min}), la valeur moyenne de γ (γ_{mean}), son écart-type (γ_{std}) et finalement le temps de calcul moyen ($t_{\text{cpu/run}}$) mis pour la réalisation d'un *run*.

Les résultats sont présentés sur le Tableau 2.2.

On constate que les résultats obtenus avec les deux approches en terme de γ sont proches mais le temps de calcul à l'issue de l'approche évolutionnaire est 4 fois plus important qu'à l'issue de *Hinfstruct*.

	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$
Approche stochastique	3,92	4,65	0,89	8 s
Approche non lisse	3,56	4,14	1,93	2 s

Tableau 2.2 Synthèse à ordre fixé – Approches évolutionnaire et non lisse

Synthèse H_∞ à ordre fixé multi-modèle

Dans un second temps on synthétise un correcteur d'ordre 5 par l'approche multi-modèle par l'approche évolutionnaire puis non lisse. Les résultats sont présentés sur le Tableau 2.3.

	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$
Approche stochastique	4,26	5,13	1,14	11 min
Approche non lisse	3,95	4,39	1,85	300 min

Tableau 2.3 Synthèse à ordre fixé multi-modèle – Approches évolutionnaire et non lisse

On constate cette fois-ci que les résultats obtenus avec les deux approches en terme de γ sont proches mais le temps de calcul à l'issue de l'approche évolutionnaire est 5 fois plus petit qu'à l'issue de *Hinfstruct*. Aussi pouvons-nous dire que notre approche de la synthèse multi-modèle constitue une alternative intéressante aux outils actuels modernes, car beaucoup moins gourmande en temps de calcul, et ce d'autant plus lorsque le nombre de modèles est important. Avec *Hinfstruct*, on cherche un correcteur qui stabilise une forme standard regroupant les m sous-modèles [APK 12], alors qu'avec l'optimisation stochastique on stabilise indépendamment les m formes standard avec un correcteur. Là encore, l'intérêt de l'approche évolutionnaire réside dans la flexibilité dont elle fait preuve et dans sa facilité de mise en œuvre.

On trace l'ensemble des réponses fréquentielles en boucle ouverte sur toute la boule de modèles sur la Figure 2-18. On constate que les marges de stabilité « pire cas » sont meilleures. Comme le montre le Tableau 2.3, la valeur de γ « pire cas » est désormais raisonnable (4,26).

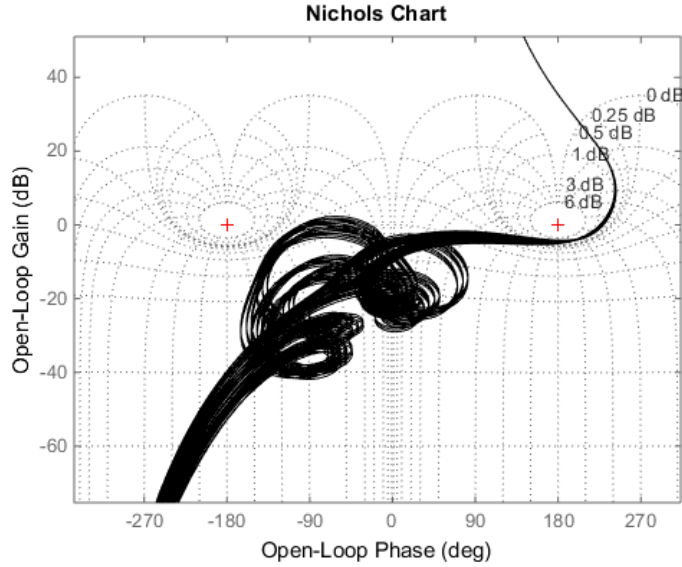


Figure 2-18: Boucles ouvertes sur l'ensemble de la boule de modèle après la synthèse multi-modèles à ordre fixé évolutionnaire

2.2.4.7.2. Méthodologie générale pour la synthèse de correcteurs structurés stabilisant

2.2.4.7.2.1 Principe général

On peut généraliser la méthode de synthèse d'un correcteur stabilisant structuré par un problème d'optimisation dont la fonction de coût à minimiser a la forme générale suivante :

$$\begin{aligned}
 & \min_{\Theta_k} f(\Theta_k) \\
 & \text{si } \bar{\lambda}_{cl} \geq 0 \\
 & \quad f(\Theta_k) = \bar{\lambda}_{cl} \\
 & \text{sinon} \\
 & \quad f(\Theta_k) = -\frac{1}{g(\Theta_k)}
 \end{aligned} \tag{2-114}$$

Où $g(\Theta_k)$ est une fonction positive de Θ_k décrivant les spécifications demandées. L'intérêt de cette formulation est sa grande flexibilité (adaptation facile à une multitude de problèmes complexes, extension au cadre multi-modèles, ...). On donne ci-dessous quelques exemples d'extension supplémentaires.

2.2.4.7.2.2 Exemple 1 : synthèse H_∞ structurée sous contrainte quelconque

Il suffit pour cela de choisir :

$$g(\Theta_k) = \|T_\infty(s)\|_\infty + f_c(\Theta_k) \tag{2-115}$$

Où la fonction $f_c(\Theta_k)$ traduit la contrainte considérée (sous la forme d'une fonction barrière par exemple). C'est cette formulation que nous avons adoptée pour le réglage de correcteurs structurés et robustes vis-à-vis d'exigences de haut niveau (voir partie suivante du document).

2.2.4.7.2.3 Exemple 2 : synthèse mixte H_2/H_∞ structurée

Considérant la Figure 2-19, il existe plusieurs formulations possibles au problème de la synthèse mixte H_2/H_∞ . On pose :

$$z_\infty(s) = T_\infty(s)e_\infty(s), \quad z_2(s) = T_2(s)e_2(s) \quad (2-116)$$

On cherche ainsi à déterminer un correcteur $K(s)$ d'ordre $n_k < n_p$ qui :

- stabilise la boucle fermée $F_l(P,K)$ de manière interne ;
- avec l'un des objectifs suivants :
 - $\alpha \|T_\infty(s)\|_\infty + \beta \|T_2(s)\|_2 < \gamma$, avec γ minimal et $(\alpha > 0, \beta > 0)$ donnés. Dans ce cas :

$$g(\Theta_k) = \alpha \|T_\infty(s)\|_\infty + \beta \|T_2(s)\|_2 \quad (2-117)$$

- $\|T_\infty(s)\|_\infty < \gamma_\infty$ sous contrainte que $\|T_2(s)\|_2 < \gamma_2$, avec γ_∞ minimal et γ_2 donné. Dans ce cas :

$$g(\Theta_k) = \|T_\infty(s)\|_\infty + f_c(\Theta_k), \quad f_c(\Theta_k) = e^{M(\|T_2(s)\|_2 - \gamma_2)} \quad (2-118)$$

Où $M \gg 1$ puisque f_c représente alors une pénalité barrière.

- $\|T_2(s)\|_2 < \gamma_2$ sous contrainte que $\|T_\infty(s)\|_\infty < \gamma_\infty$, avec γ_2 minimal et γ_∞ donné. Dans ce cas :

$$g(\Theta_k) = \|T_2(s)\|_2 + f_c(\Theta_k), \quad f_c(\Theta_k) = e^{M(\|T_\infty(s)\|_\infty - \gamma_\infty)} \quad (2-119)$$

Où $M \gg 1$ puisque f_c représente alors une pénalité barrière.

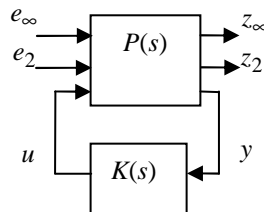


Figure 2-19: La synthèse mixte H_2/H_∞

Notons que comme pour la synthèse « mono-modèle » du §2.2.4, rien n'empêche de rajouter les contraintes sur la stabilité du correcteur, ainsi que sur la position des pôles de l'ensemble des boucles fermées.

De même rien n'empêche l'extension de cette technique au cadre multi-modèle.

2.2.4.8. Conclusion

Dans cette partie, une méthode de synthèse de correcteur structuré originale a été mise en œuvre. Si le temps de calcul de la synthèse de correcteur par l'optimisation stochastique est plus important que pour d'autres techniques pour des problèmes « simples » (on évalue plus de fois la fonction de coût), l'approche tire son intérêt de la complexité des critères employés qui peuvent être quelconques et que d'autres techniques ne peuvent pas traiter facilement. De même rien n'empêche de rajouter des contraintes exotiques sur tel ou tel transfert par des pénalités. La liste de ces critères n'est en rien exhaustive : rien n'empêche de rajouter des critères temporels tels que des bornes sur les réponses impulsionnelles, indicielles, ... et de déterminer ainsi un correcteur structuré en optimisant un critère de forme quelconque (voir partie 3).

2.3. μ -synthèse

2.3.1. Le problème de la robustesse en performance

On considère le schéma de commande de la Figure 2-20 où $P(s)$ est un système linéaire et invariant, $K(s)$ est un correcteur linéaire et invariant à déterminer et où $\Delta(s)$ représente les incertitudes sur le système. $(z_\Delta, e_\Delta) \in \mathbf{R}^{n_\Delta \times n_\Delta}$ et $(z, e) \in \mathbf{R}^{n_z \times n_e}$ sont respectivement les canaux liés aux incertitudes et les canaux liés à la performance. $y \in \mathbf{R}^{n_y}$ représente le vecteur de mesures nécessaires au correcteur pour élaborer le vecteur de commandes $u \in \mathbf{R}^{n_u}$.

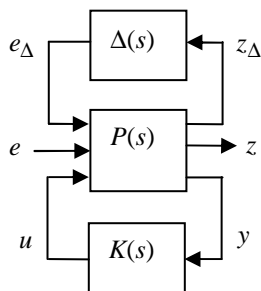


Figure 2-20: Schéma de commande avec bloc d'incertitudes sous forme LFT

On note :

$$T(s) = F_l(P(s), K(s)) = \begin{pmatrix} T_{z_{\Delta}e_{\Delta}}(s) & T_{z_{\Delta}e}(s) \\ T_{z_e\Delta}(s) & T_{z_e}(s) \end{pmatrix} \quad (2-120)$$

Nous supposons sans perte de généralité que $\Delta(s)$ est normalisée, c'est-à-dire que $\|\Delta(s)\|_{\infty} < 1$.

D'après le théorème du petit gain [ZHO 96], le système $T(s) \in \mathbf{RH}_{\infty}$ est stable de manière interne pour toute incertitude non structurée $\Delta(s) \in \mathbf{RH}_{\infty}$ telle que $\|\Delta(s)\|_{\infty} < 1$ si et seulement si :

$$\|T_{z_{\Delta}e_{\Delta}}(s)\|_{\infty} \leq 1 \quad (2-121)$$

Ainsi le théorème du petit gain donne une condition nécessaire et suffisante pour la robustesse en stabilité dans le cadre non structuré. Cependant il est connu que l'analyse de robustesse avec le théorème du petit gain dans ce cadre peut amener à des résultats très conservatifs en dépit de la facilité de mise en œuvre. Des résultats bien plus pertinents peuvent être obtenus en prenant en compte directement la structure diagonale par bloc obtenue en (2-122) lorsque l'on met le système et son incertitude sous forme LFT¹⁰ :

$$\begin{aligned} \Delta(s) &= \text{diag}\{\Delta_1(s), \dots, \Delta_q(s), \delta_1 I_{r_1}, \dots, \delta_r I_{r_r}, \varepsilon_1 I_{c_1}, \dots, \varepsilon_c I_{c_c}\} \\ \Delta_i(s) &\in \mathbf{RH}_{\infty}^{n_i \times n_i}, \delta_j \in \mathbf{R}, \varepsilon_k \in \mathbf{C} \end{aligned} \quad (2-122)$$

Et vérifiant les conditions de normalisation :

$$\left(\|\Delta_i(s)\|_{\infty} < 1 ; \delta_j \in]-1; 1[; |\varepsilon_k| < 1 \right) \Leftrightarrow \left(\|\Delta(s)\|_{\infty} < 1 \right) \quad (2-123)$$

Noter que d'après (2-122) :

$$n_{\Delta} = \sum_{i=1}^q n_i + \sum_{i=1}^r r_i + \sum_{i=1}^c c_i \quad (2-124)$$

Ainsi, on définit l'ensemble $\underline{\Delta}$ des matrices complexes ayant la même structure que $\Delta(s)$:

$$\underline{\Delta} := \begin{cases} \Delta = \text{diag}\{\Delta_1, \dots, \Delta_q, \delta_1 I_{r_1}, \dots, \delta_r I_{r_r}, \varepsilon_1 I_{c_1}, \dots, \varepsilon_c I_{c_c}\} \in \mathbf{C}^{n_{\Delta} \times n_{\Delta}} \\ \Delta_i \in \mathbf{C}^{n_i \times n_i}, \delta_j \in \mathbf{R}, \varepsilon_k \in \mathbf{C} \end{cases} \quad (2-125)$$

L'hypothèse de structuration sous forme diagonale par bloc de la matrice d'incertitude n'est en rien restrictive car lors de la mise sous forme LFT de la Figure

¹⁰ Transformation Fractionnaire Linéaire

2-20 des incertitudes d'un système, on peut toujours se ramener à une structure comme (2-125).

Pour prendre en compte la structure particulière $\underline{\Delta}$, on définit la valeur singulière structurée d'une matrice P relative à la structure $\underline{\Delta}$ [ZHO 96] :

$$\begin{aligned} \mu_{\underline{\Delta}}(P) &:= \left(\inf_{\Delta \in \underline{\Delta}} \left(\bar{\sigma}(\Delta) : \det(I - \Delta P) = 0 \right) \right)^{-1} \\ \mu_{\underline{\Delta}}(P) &:= 0 \text{ si } \forall \Delta \in \underline{\Delta} \det(I - \Delta P) \neq 0 \end{aligned} \quad (2-126)$$

$\mu_{\underline{\Delta}}(P)^{-1}$ est donc la plus petite norme d'une matrice $\Delta \in \underline{\Delta}$ pour laquelle le système linéaire de la Figure 2-21 en z et e devient singulier.

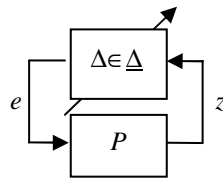


Figure 2-21: Interprétation de la valeur singulière structurée

$$\begin{cases} e = \Delta z \\ z = Pe \end{cases} \Leftrightarrow \begin{pmatrix} I & -\Delta \\ -P & I \end{pmatrix} \begin{pmatrix} e \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{cases} (I - \Delta P)e = 0 \\ z = Pe \end{cases} \quad (2-127)$$

Ainsi le théorème du petit gain peut-être généralisé [ZHO 96]. Le système $T(s) \in \mathbf{RH}_{\infty}$ est stable de manière interne pour toute incertitude structurée $\Delta(s) \in \underline{\Delta}$ si et seulement si :

$$\forall \omega, \mu_{\underline{\Delta}}(T_{z_{\Delta} w_{\Delta}}(j\omega)) \leq 1 \quad (2-128)$$

Réaliser une analyse de robustesse dans le cadre structuré avec le théorème du petit gain généralisé est appelé μ -analyse.

Il est possible d'étendre l'analyse de robustesse en stabilité à une analyse de robustesse en performance. Ainsi en supposant que la performance est atteinte lorsque $\|T_{ze}(s)\|_{\infty} \leq 1$, le but est d'assurer cette condition pour toute matrice $\Delta(s) \in \underline{\Delta}$: cela définit le problème de robustesse en performance. D'après le théorème du petit gain, c'est le cas si et seulement si le système de la

Figure 2-22 est stable pour tout $\Delta_f(s)$ telle que $\|\Delta_f(s)\|_{\infty} < 1$, où $\Delta_f(s)$ est une incertitude fictive non structurée introduite entre le canal de performance z et les entrées exogènes e .

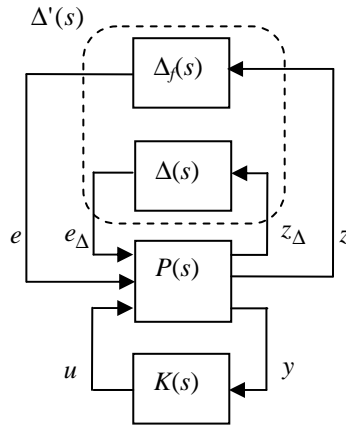


Figure 2-22: Schéma de commande pour l'analyse de robustesse en performance

$\Delta_f(s)$ et $\Delta(s)$ peuvent être rassemblées dans une matrice d'incertitudes augmentée $\Delta'(s)$:

$$\Delta'(s) = \text{diag}(\Delta_f(s), \Delta(s)) \quad (2-129)$$

Comme $\Delta'(s)$ est structurée, rien n'empêche d'utiliser le théorème du petit gain généralisé. Ainsi le système de la

Figure 2-22 est stable de manière interne pour tout $\Delta'(s) \in \underline{\Delta}'$ si et seulement si :

$$\forall \omega, \mu_{\underline{\Delta}'}(T(j\omega)) \leq 1 \quad (2-130)$$

Avec :

$$\underline{\Delta}' := \left\{ \Delta' = \text{diag}(\Delta, \Delta_f), \Delta \in \underline{\Delta}, \Delta_f \in \mathbf{C}^{n_z \times n_w} \right\} \quad (2-131)$$

Ce système est stable pour tout $\Delta'(s) \in \underline{\Delta}'$ telle que :

$$\begin{aligned} & \|\Delta'(s)\|_{\infty} < 1 \\ & \Leftrightarrow \\ & \begin{cases} \|\Delta(s)\|_{\infty} < 1, \forall \Delta \in \underline{\Delta} \\ \|\Delta_f(s)\|_{\infty} < 1 \Rightarrow \|T_{ze}(s)\|_{\infty} < 1 \end{cases} \\ & \Leftrightarrow \\ & \|T_{ze}(s)\|_{\infty} < 1, \forall \Delta(s) \in \underline{\Delta} \end{aligned} \quad (2-132)$$

Le problème de l'analyse de robustesse en performance peut donc se réduire à un problème de robustesse en stabilité augmenté :

$$\|T_{ze}\|_{\infty} < 1, \forall \Delta(s) \in \underline{\Delta} \Leftrightarrow \forall \omega, \mu_{\underline{\Delta}'}(F_l(P(j\omega), K(j\omega))) < 1 \quad (2-133)$$

L'analyse structurée par $\mu_{\underline{\Delta}}(P)$ fournit un cadre moins conservatif que le cadre non-structuré. Cependant l'usage de la valeur singulière structurée peut devenir vite complexe car $\mu_{\underline{\Delta}}(P)$ est très difficile à calculer (en pratique sauf dans le cas d'exemples très simples, le calcul de $\mu_{\underline{\Delta}}(P)$ est un problème à complexité exponentielle). On en cherche donc un majorant. On montre que [YOU 90] :

$$\mu_{\underline{\Delta}}(P) \leq \min_{D \in \underline{D}} \bar{\sigma}(DPD^{-1}) \quad (2-134)$$

Où $D \in \underline{D}$ est une matrice qui commute avec Δ (c'est-à-dire que $\Delta D = D\Delta$) :

$$\underline{D} := \left\{ \begin{array}{l} D = \text{diag}(d_1 I_{k_1}, \dots, d_q I_{k_q}, D_1, \dots, D_r, D_1', \dots, D_c') \\ d_i \in \mathbf{R} ; d_i > 0 \\ D_i \in \mathbf{C}^{r_i \times r_i} ; D_i = D_i^* > 0 ; D_i' \in \mathbf{C}^{c_i \times c_i} ; D_i' = D_i'^* > 0 \end{array} \right\} \quad (2-135)$$

On peut calculer cette borne supérieure de $\mu_{\underline{\Delta}}(P)$ en résolvant le problème d'optimisation suivant :

$$\mu_{\underline{\Delta}}(P) \leq \alpha_{opt} \text{ avec } \alpha_{opt} = \min_{\alpha, D \in \underline{D}} \alpha : \begin{cases} \alpha > 0 \\ P^* DP - \alpha^2 D < 0 \end{cases} \quad (2-136)$$

Qui est un problème d'optimisation quasi-convexe :

- A α fixé, la contrainte est une inégalité matricielle linéaire en D ,
- A D fixée, il s'agit d'un problème de calcul de valeurs propres.

2.3.2. Principe de la μ -synthèse

L'émergence des techniques H_∞ (voir première partie) a permis de prendre en compte la contrainte de robustesse en stabilité pour le calcul d'un correcteur stabilisant $K(s)$. A la vue du paragraphe précédent, un problème bien plus ambitieux pourrait être le calcul d'un correcteur stabilisant $K(s)$ satisfaisant $\|T_{ze}(s)\|_\infty < \gamma$ pour tout $\Delta(s) \in \mathbf{RH}_\infty$ telle que $\|\Delta(s)\|_\infty < \gamma^{-1}$.

En supposant que $\Delta(s) \in \underline{\Delta}$, alors le problème revient à calculer un correcteur stabilisant $K(s)$ satisfaisant $\|F_l(P, K)\|_\infty < 1$ et tel que $\|T_{ze}(s)\|_\infty < 1, \Delta(s) \in \underline{\Delta}$, c'est-à-dire associer le problème de la synthèse H_∞ standard avec le problème de la μ -analyse telle que présentée dans les paragraphes précédents : c'est le problème de μ -synthèse [YOU 93].

En introduisant une incertitude fictive sur les canaux de performance telle que sur la

Figure 2-22, le problème précédent peut être reformulé en calcul d'un correcteur stabilisant $K(s)$ et satisfaisant :

$$\forall \omega, \mu_{\underline{\Delta}}(F_l(P(j\omega), K(j\omega))) \leq 1 \quad (2-137)$$

Malheureusement, il n'existe pas de solution exacte au problème de μ -synthèse telle qu'elle existe pour le problème H_∞ standard [DOY 89]. Cependant, on a vu que :

$$\forall \omega, \mu_{\underline{\Delta}}(F_l(P(j\omega), K(j\omega))) \leq \bar{\sigma}(D_\omega F_l(P(j\omega), K(j\omega)) D_\omega^{-1}) \quad (2-138)$$

Où D_ω est n'importe quelle matrice inversible qui commute avec la structure $\underline{\Delta}'$ (ie $\underline{\Delta}' D_\omega = D_\omega \underline{\Delta}'$).

Ainsi pouvons-nous approcher le problème de μ -synthèse par le problème suivant : trouver un correcteur stabilisant $K(s)$ et une matrice de transfert $D(s)$ tels que :

- $D(s), D^{-1}(s) \in \mathbf{RH}_\infty$,
- $D(s)\underline{\Delta}'(s) = \underline{\Delta}'(s)D(s), \forall \underline{\Delta}'(s) \in \underline{\Delta}'$
- $\|D(s)F_l(P, K)D(s)^{-1}\|_\infty < 1$

A noter que $D(s)$ peut être choisie sans perte de généralité sous la forme (2-139) où chaque $D_m(s)$ commute avec le bloc correspondant dans $\underline{\Delta}(s)$.

$$D(s) = \begin{pmatrix} \text{diag}(D_m(s), m = 1, \dots, n_{\underline{\Delta}}) & 0 \\ 0 & D_{ze}(s) \end{pmatrix} \quad (2-139)$$

A noter de plus que le scaling $D_{ze}(s)$ peut être choisi égal à la matrice identité lorsque $n_z = n_e$. Afin de simplifier les développements suivants, nous garderons cette dernière hypothèse.

2.3.3. μ -synthèse d'ordre plein

2.3.3.1. Approches itératives

L'approche la plus courante est la D - K itération qui consiste à chercher alternativement D et K [ZHO 96].

- A $D(s)$ fixée, on calcule un correcteur $K(s)$ en résolvant le problème H_∞ standard de la Figure 2-23
- A $K(s)$ fixée, on détermine la matrice $D(s)$ à l'issue d'une μ -analyse en :
 - Calculant une borne supérieure de la valeur singulière structurée $\mu_{\underline{\Delta}}(P)$ pour un ensemble choisi de fréquence ω en résolvant (2-136), ainsi que l'ensemble des scalings D_ω correspondant,
 - Interpolant l'ensemble des scalings D_ω par un transfert $D(s)$ stable à inverse stable.

La procédure s'initialise en choisissant l'identité pour $D(s)$.

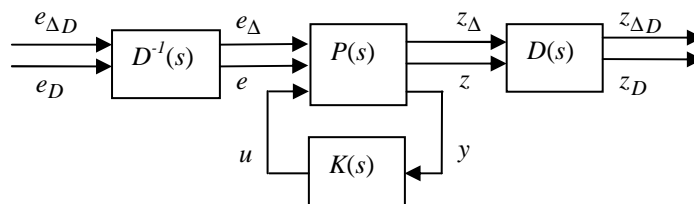


Figure 2-23: D-K itération : Calcul de $K(s)$ à $D(s)$ fixée

Cette procédure amène les remarques suivantes :

- on s'arrête lorsque l'indicateur γ n'évolue plus d'une itération à l'autre ou lorsque les scalings n'évoluent plus d'une itération à l'autre ;
- on note que l'ordre du correcteur calculé à chaque itération grandit de plus en plus puisque l'ordre de la forme standard augmentée des scalings $D(s)$ croît au fur et à mesure des itérations ;
- la phase d'interpolation est numériquement très sensible : nombreuses possibilités dues au choix du maillage fréquentiel (ce dernier doit être suffisamment fin pour prendre en compte toutes les dynamiques du système), choix de la méthode d'interpolation et surtout choix de l'ordre des scalings.

Malgré tout cette approche donne de très bons résultats dans de nombreuses applications [DAL 08].

En réalité cette approche dispose d'un certain conservatisme car elle suppose toutes les incertitudes complexes, ce qui n'est pas le cas lorsqu'il s'agit d'incertitudes paramétriques réelles. Il peut être intéressant de tenir compte de cette distinction. Ainsi d'autres approches plus ou moins complexes à mettre en œuvre existent.

Citons par exemple [YOU 93] où on montre qu'une borne supérieure de $\mu_{\Delta}(P)$ est :

$$\forall \omega, \mu_{\Delta}(F_l(P(j\omega), K(j\omega))) < \bar{\sigma} \left((D_{\omega} F_l(P(j\omega), K(j\omega)) D_{\omega}^{-1} - G_{\beta\omega}) G_{\alpha\omega} \right) \leq \beta_{\omega}$$

$$G_{\beta\omega} = j\beta_{\omega} G_{\omega}, \quad G_{\alpha\omega} = (I + G_{\omega}^2)^{-1/2} \quad (2-140)$$

On calcule alors la borne maximale en résolvant le problème d'optimisation suivant :

$$\mu_{\Delta}(P) \leq \beta_{opt} \quad \text{avec} \quad \beta_{opt} = \min_{\beta, D \in \underline{D}, G \in \underline{G}} \beta : \begin{cases} \beta \geq 0 \\ P^* DP + j(GP - P^* G) - \beta^2 D \leq 0 \end{cases} \quad (2-141)$$

Où :

$$\underline{G} := \left\{ \begin{array}{l} G = \text{diag}(0_{k_1}, \dots, 0_{k_q}, G_1, \dots, G_r, 0_{c_1}, \dots, 0_{c_c}) \\ G_i \in \mathbf{C}^{r_i \times r_i} ; G_i = G_i^* \end{array} \right\} \quad (2-142)$$

G contient des matrices dont les éléments non nuls correspondent aux blocs réels de Δ .

On peut alors employer une procédure similaire à la précédente, appelé D - G , K itération [YOU 93]. Cette approche est cependant très lourde à mettre en œuvre et très sensible numériquement. Ainsi pouvons-nous nous orienter vers la μ - K itération qui permet d'éviter l'interpolation de la matrice G [TOF 95].

D'autres approches permettant l'éviter la phase d'interpolation des scaling existent mais elles sont lourdes à mettre en œuvre et numériquement sensibles. Citons par exemple [TSA 02] et [LIN 05].

2.3.3.2. Reformulation pour l'optimisation stochastique

2.3.3.2.1. Introduction

Comme on le voit le problème de la μ -synthèse reste ouvert. La plupart des approches sont itératives et souffrent de deux inconvénients majeurs :

- Tout d'abord ces approches disposent d'une phase d'interpolation de scalings très sensible numériquement rendant la méthode considérée sous-optimale de par les nombreuses possibilités s'offrant à l'ingénieur : choix de la méthode d'interpolation, choix du maillage fréquentielle, choix de l'ordre des scalings...
- Par ailleurs ces méthodes, parce qu'itératives, produisent des correcteurs dont l'ordre croît au fur et à mesure des itérations ce qui peut s'avérer numériquement vite problématique.

Ainsi nous proposons dans [FEY 14b] de retraiter le problème de la μ -synthèse par l'optimisation stochastique.

Dans la littérature, on trouve quelques tentatives intéressantes principalement basées sur une approche directe du problème de μ -synthèse. Ainsi par exemple, dans [LAR 12], on calcule avec PSO un correcteur structuré de type PID en attaquant le problème de manière frontale :

$$\min_{K(s)} \mu_{\underline{\Delta}}(T(s)) \quad (2-143)$$

$$\text{Sous contraintes } \mu_{\underline{\Delta}}(T_{z_{\Delta} e_{\Delta}}(s)) < 1$$

Cette approche semble efficace car « directe » mais n'a été testée que pour le calcul d'un correcteur de type PID (4 paramètres à déterminer). Par ailleurs, la fonction de coût envisagée consiste à évaluer directement la valeur singulière structurée : bien que la phase d'interpolation des scalings n'existe plus, la problématique du choix du maillage reste entière et le calcul de la valeur singulière reste couteux en temps dans le cas d'un maillage fin (nécessaire à l'obtention d'un bon résultat).

2.3.3.2.2. Principe de l'approche

Nous proposons ici de résoudre directement le problème d'optimisation global suivant :

$$\min_{X \in [x; x]} f(X) \Leftrightarrow \min_{D(s), D(s)^{-1} \in \mathbf{RH}_\infty} \gamma \quad (2-144)$$

Avec :

$$\gamma = \min_{K(s)} \left\| D(s) F_l(P, K) D(s)^{-1} \right\|_\infty \quad (2-145)$$

Ce problème n'est pas sans rappeler celui de la synthèse H_∞ structurée (§ 2.2.4.1) et va donc suivre le même schéma.

En posant pour inconnue :

$$\Theta_D = \begin{pmatrix} A_D & B_D \\ C_D & D_D \end{pmatrix} \quad (2-146)$$

Et :

$$\bar{\lambda}_D = \max \left[\max_i (\operatorname{Re}(\lambda_i(A_D))), \max_i (\operatorname{Re}(\lambda_i(A_D - B_D D_D^{-1} C_D))) \right] \quad (2-147)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre,

Alors on trouve les scalings et le correcteur d'ordre plein μ -optimaux en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_D} f(\Theta_D) \\ & \text{si } \bar{\lambda}_D \geq 0 \\ & f(\Theta_k) = \bar{\lambda}_D \end{aligned} \quad (2-148)$$

sinon

$$f(\Theta_k) = -\frac{1}{\gamma}, \quad \gamma = \min_{K(s)} \left\| D(s) F_l(P, K) D(s)^{-1} \right\|_\infty$$

2.3.3.2.3. Mise en œuvre

Le scaling $D(s)$ dispose de la structure (2-139) pour laquelle on supposera pour simplifier que où $n_z = n_e$:

$$D(s) = \begin{pmatrix} \text{diag}(D_m(s), m=1, \dots, n_\Delta) & 0 \\ 0 & I_{n_e} \end{pmatrix} \quad (2-149)$$

Aussi l'inconnue X représente les coefficients des représentations d'état des $n_\Delta = q+r+c$ éléments $D_m(s)$ constituant $D(s)$.

Chaque terme $D_m(s)$ dispose de la représentation d'état suivante :

$$D_m(s) = \left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & D_m \end{array} \right] \quad (2-150)$$

Comme pour l'approche stochastique de la synthèse H_∞ , on accélère grandement le temps de convergence de l'algorithme en cherchant chaque $D_m(s)$ sous forme tridiagonale et en transformant l'espace de recherche (voir § 2.2.4.2). En supposant que chaque $D_m(s)$ cherchée soit SISO et d'ordre n_m , nous posons pour vecteur de décision :

$$\begin{aligned} X &= (X_{D_1}, \dots, X_{D_m}, \dots, X_{D_{n_\Delta}})^T \\ X_{D_m} &= (x_A, x_B, x_C, x_D)_m \\ \left. \begin{aligned} x_A &= (a_{1,1}, \dots, a_{n,n}, a_{1,2}, \dots, a_{n-1,n}, a_{2,1}, \dots, a_{n,n-1}) \\ x_B &= (b_{1,1}, \dots, b_{n,1}) \\ x_C &= (c_{1,1}, \dots, c_{1,n}) \\ x_D &= d \end{aligned} \right\} n = n_m \end{aligned} \quad (2-151)$$

A noter que la structure (2-149) est une structure décentralisée particulière (voir § 2.2.4.3.2) :

$$\begin{aligned} D(s) &= \left[\begin{array}{cc|cc} \text{diag}(A_m, m=1, \dots, n_\Delta) & 0 & \text{diag}(B_m, m=1, \dots, n_\Delta) & 0 \\ 0 & -I_{n_e} & 0 & 0_{n_e} \\ \hline \text{diag}(C_m, m=1, \dots, n_\Delta) & 0 & \text{diag}(D_m, m=1, \dots, n_\Delta) & 0 \\ 0 & 0_{n_e} & 0 & I_{n_e} \end{array} \right] \\ &= \left[\begin{array}{c|c} A_D & B_D \\ \hline C_D & D_D \end{array} \right] \end{aligned} \quad (2-152)$$

$$\begin{aligned} D(s)^{-1} &= \left[\begin{array}{c|c} A_D - B_D D_D^{-1} C_D & -B_D D_D^{-1} \\ \hline D_D^{-1} C_D & D_D^{-1} \end{array} \right] \\ &= \left[\begin{array}{c|c} \underline{A}_D & \underline{B}_D \\ \hline \underline{C}_D & \underline{D}_D \end{array} \right] \end{aligned}$$

Remarquons la présence d'un terme $-I_{n_e}$ dans A_D servant à assurer que la partie de $D(s)$ relative au canal de performance soit stable.

La contrainte de stabilité des scalings s'écrit :

$$\begin{aligned} D(s) \in \mathbf{RH}_\infty &\Leftrightarrow \max_i (\operatorname{Re}(\lambda_i(A_D))) < 0 \\ D(s)^{-1} \in \mathbf{RH}_\infty &\Leftrightarrow \max_i (\operatorname{Re}(\lambda_i(A_D - B_D D_D^{-1} C_D))) < 0 \end{aligned} \quad (2-153)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Enfin, en supposant que le $P(s)$ de la Figure 2-23 dispose de la représentation d'état suivante :

$$\begin{bmatrix} \dot{x} \\ z_\Delta \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ e_\Delta \\ e \\ u \end{bmatrix} \quad (2-154)$$

Alors, on montre facilement que la représentation d'état de la forme standard augmentée des scalings $D(s)P(s)D(s)^{-1}$ de la Figure 2-23 s'écrit :

$$\begin{bmatrix} \dot{x} \\ z_{D_\Delta} \\ z_D \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 \underline{C}_D & 0 & B_1 \underline{D}_D & B_2 \\ 0 & \underline{A}_D & 0 & \underline{B}_D & 0 \\ \underline{B}_D C_1 & B_D D_{11} \underline{C}_D & A_D & B_D D_{11} \underline{D}_D & B_D D_{12} \\ \underline{D}_D C_1 & D_D D_{11} \underline{C}_D & C_D & D_D D_{11} \underline{D}_D & D_D D_{12} \\ C_2 & D_{21} \underline{C}_D & 0 & D_{21} \underline{D}_D & D_{22} \end{bmatrix} \begin{bmatrix} x \\ e_{D_\Delta} \\ e_D \\ u \end{bmatrix} \quad (2-155)$$

2.3.3.2.4. Evaluation de la fonction de coût

Nous cherchons à résoudre le problème d'optimisation suivant :

$$\begin{aligned} \min f(X) \\ X \in [x; \bar{x}] \end{aligned} \quad (2-156)$$

Où la variable X est définie en (2-151) et où la fonction $f(X)$ consiste à réaliser les étapes suivantes :

- construire les réalisations $\begin{bmatrix} A_m & B_m \\ C_m & D_m \end{bmatrix}, m = 1, \dots, n_\Delta$ à partir de X ;
- construire les réalisations $\begin{bmatrix} A_D & B_D \\ C_D & D_D \end{bmatrix}$ et $\begin{bmatrix} \underline{A}_D & \underline{B}_D \\ \underline{C}_D & \underline{D}_D \end{bmatrix}$ avec (2-152) ;
- calculer l'abscisse spectrale suivante :

$$\begin{aligned}
 \lambda_u &= \max_i (\operatorname{Re}(\lambda_i(A_D))) \\
 \lambda_d &= \max_i (\operatorname{Re}(\lambda_i(\underline{A}_D))) \\
 \bar{\lambda} &= \max(\lambda_u, \lambda_d)
 \end{aligned} \tag{2-157}$$

- Si $\bar{\lambda} \geq 0$ alors $f(X) = \bar{\lambda}$,
- Sinon :
 - construire la représentation d'état de $D(s)P(s)D(s)^{-1}$ à l'aide de (2-155) ;
 - résoudre le problème H_∞ standard (2-145), puis retenir la valeur de γ obtenue ;
 - évaluer $f(X) = -\frac{1}{\gamma}$.

On remarque que cette fonction coût suit le même principe que pour la synthèse H_∞ structurée (§2.2.4.4)

A l'issue de l'optimisation, le correcteur μ -optimal est directement obtenu en résolvant le problème H_∞ standard (2-145) avec les meilleurs scalings obtenus.

Cas de scalings statiques

On note par ailleurs que la première étape de l'optimisation consiste en la stabilisation des scalings et de leurs inverses. Dans le cas de scalings statiques, cette étape n'est plus nécessaire et l'évaluation de la fonction de coût $f(X)$ est dès lors beaucoup plus simple ; elle consiste à réaliser les étapes suivantes :

- construire chaque D_m , et chaque D_m^{-1} puis construire les matrices de transfert

$$D = \begin{pmatrix} \operatorname{diag}(D_m, m=1, \dots, n_\Delta) & 0 \\ 0 & I_{n_e} \end{pmatrix} \text{ et } D^{-1} = \begin{pmatrix} \operatorname{diag}(D_m^{-1}, m=1, \dots, n_\Delta) & 0 \\ 0 & I_{n_e} \end{pmatrix};$$

- construire la représentation d'état de $DP(s)D^{-1}$:

$$\begin{bmatrix} \dot{x} \\ z_{D\Delta} \\ z_D \\ y \end{bmatrix} = \left[\begin{array}{c|c|c} A & B_1 D^{-1} & B_2 \\ \hline DC_1 & DD_{11} D^{-1} & DD_{12} \\ \hline C_2 & D_{21} D^{-1} & D_{22} \end{array} \right] \begin{bmatrix} x \\ e_{D\Delta} \\ e_D \\ u \end{bmatrix} \tag{2-158}$$

- résoudre le problème H_∞ standard (2-145) puis retenir la valeur de γ obtenue ;
- évaluer $f(X) = -\frac{1}{\gamma}$;

2.3.3.2.5. Exemple de mise en œuvre

2.3.3.2.5.1 Rappel de spécification

On reprend l'exemple du paragraphe 2.2.4.7.1.2 que nous avons employé lors de la synthèse H_∞ multi-modèle par approche stochastique.

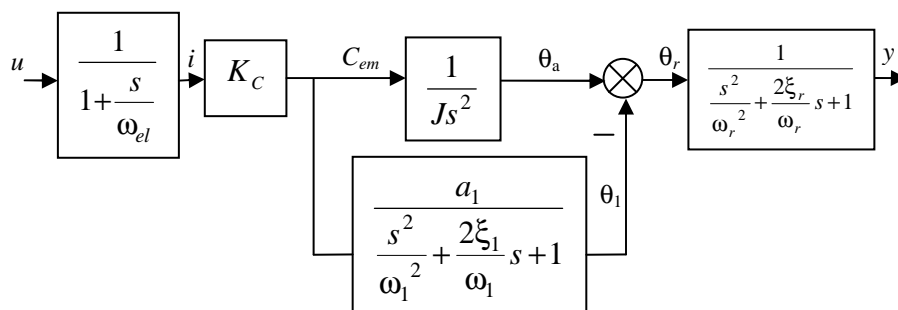


Figure 2-24: Système à mode souple

On a :

$$H_0(s) = \frac{K_c}{\left(1 + \frac{s}{\omega_{el}}\right)} \left(\frac{1}{Js^2} - \frac{a_1}{\frac{s^2}{\omega_1^2} + \frac{2\xi_1}{\omega_1}s + 1} \right) \left(\frac{1}{\frac{s^2}{\omega_r^2} + \frac{2\xi_r}{\omega_r}s + 1} \right) \quad (2-159)$$

Les valeurs des paramètres, sujets à des incertitudes, sont rappelées ci-après.

$$\begin{aligned} J &= 0.0014 \text{ kg.m}^2, \quad K_c = 0.06 \text{ Nm.A}^{-1}, \\ \omega_r &= 1382 \text{ rad.s}^{-1}, \quad \xi_r = 0.5 \\ a_1 &= -7.10^{-4} \pm 7\%, \quad \xi_1 = 0.029 \pm 70\%, \\ \omega_1 &= 1335 \text{ s}^{-1} \pm 10\%, \quad \omega_{el} = 1257 \text{ s}^{-1} \pm 20\% \end{aligned} \quad (2-160)$$

Ces incertitudes nous avaient permis (en considérant les valeurs minimales, maximales et médianes) de définir une boule de 81 modèles puis de procéder à une synthèse multi-modèles efficace. Nous choisissons ici de considérer le système avec ses incertitudes paramétriques pour procéder à une μ -synthèse.

On rappelle la spécification suivante :

- bande passante supérieure à 40 Hz ;
- marge de gain supérieure à 5 dB ;
- marge de phase supérieure à 40° ;
- pas d'erreur statique pour une perturbation en échelon en entrée de modèle ;
- stabiliser le système vis-à-vis de ses incertitudes paramétriques.

Nous avons vu au paragraphe 2.2.4.7.1.2 que cette spécification pouvait être satisfaite par *loop-shaping* ((2-162), Figure 2-25) avec le jeu de pondérations suivant :

$$W_o(s) = \frac{128.5(s + 81.24)}{s} \frac{\left(1 + \frac{s}{5.03}\right)}{\left(1 + \frac{s}{1257}\right)}$$

$$W_i(s) = \left(\frac{\frac{s^2}{942^2} + \frac{0.25s}{942} + 1}{\frac{s^2}{942^2} + \frac{1.8s}{942} + 1} \right) \left(\frac{\frac{s^2}{2200^2} + \frac{0.01s}{2200} + 1}{\frac{s^2}{2200^2} + \frac{2.2s}{2200} + 1} \right) \frac{1}{\left(\frac{s}{942} + 1\right)}$$
(2-161)

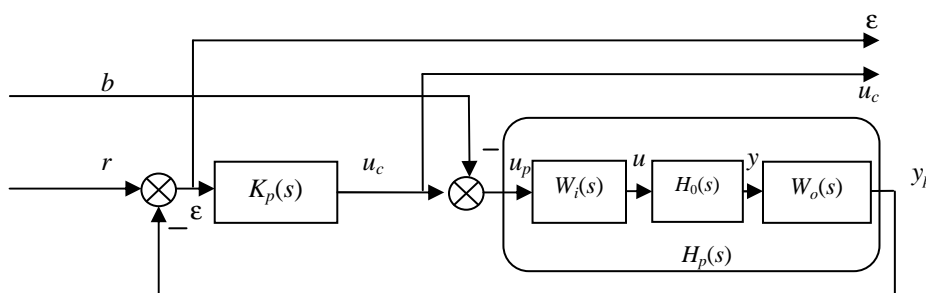


Figure 2-25: Synthèse H_∞ par loop-shaping

$$K_p(s) = \arg \min \gamma, \left\| \begin{array}{cc} S_p & S_p H_p \\ K_p S_p & K_p S_p H_p \end{array} \right\|_\infty < \gamma, \quad S_p = (I + K_p H_p)^{-1}$$
(2-162)

La synthèse nominale conduit à $\gamma = 3.30$ et la valeur γ « pire cas » peut atteindre $\gamma_{wc} = 10.5$. On décide maintenant de réduire la variation de γ vis-à-vis des incertitudes paramétrique par une μ -synthèse et nous posons pour la suite : $\gamma_{obj} = 3.30$; l'objectif est donc de synthétiser un correcteur tel que γ reste autour de γ_{obj} quelles que soient les incertitudes considérées.

2.3.3.2.5.2 Mise en forme pour la μ -synthèse

2.3.3.2.5.2.1 Mise sous forme LFT du système incertain

La première étape consiste à mettre le système (2-159) et ses incertitudes (2-160) sous la forme LFT de la Figure 2-26.

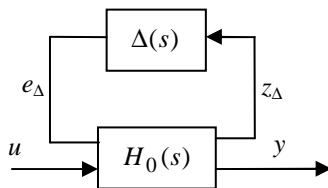


Figure 2-26: forme LFT

Cet exemple est un cas particulier d'un système à n modes souples incertains dont une mise sous forme LFT est proposée en Annexe A.

En écrivant les paramètres incertains sous la forme :

- ω_{el} de sorte que $\omega_{el} \in [\omega_{el_{\min}}; \omega_{el_{\max}}]$ et :

$$\omega_{el} = \omega_{el0} + \delta\omega_{el} \omega_{el1}, \quad |\delta\omega_{el}| \leq 1, \quad \omega_{el1} = \frac{\omega_{el_{\max}} - \omega_{el_{\min}}}{2} \quad (2-163)$$

- ξ_1 de sorte que $\xi_1 \in [\xi_{1_{\min}}; \xi_{1_{\max}}]$ et :

$$\xi_1 = \xi_{10} + \delta\xi_1 \xi_{11}, \quad |\delta\xi_1| \leq 1, \quad \xi_{11} = \frac{\xi_{1_{\max}} - \xi_{1_{\min}}}{2} \quad (2-164)$$

- a_1 de sorte que $a_1 \in [a_{1_{\min}}; a_{1_{\max}}]$ et :

$$a_1 = a_{10} + \delta a_1 a_{11}, \quad |\delta a_1| \leq 1, \quad a_{11} = \frac{a_{1_{\max}} - a_{1_{\min}}}{2} \quad (2-165)$$

- ω_1 de sorte que $\omega_1 \in [\omega_{1_{\min}}; \omega_{1_{\max}}]$ et :

$$\omega_1 = \omega_{10} + \delta\omega_1 \omega_{11}, \quad |\delta\omega_1| \leq 1, \quad \omega_{11} = \frac{\omega_{1_{\max}} - \omega_{1_{\min}}}{2} \quad (2-166)$$

Une forme LFT dans l'espace d'état du système (2-159)-(2-160) est (voir Annexe A) :

$$\begin{aligned}
 \frac{dX}{dt} &= AX + Bu + Me_{\Delta} \\
 y &= CX + Du \\
 z_{\Delta} &= NX \\
 e_{\Delta} &= \Delta z_{\Delta}, \quad \|\Delta\|_{\infty} \leq 1
 \end{aligned} \tag{2-167}$$

Avec :

$$\begin{aligned}
 X &= (i \quad \theta_a \quad \dot{\theta}_a \quad \theta_1 \quad \dot{\theta}_1 \quad \theta_r \quad \dot{\theta}_r \quad \chi)^T \\
 e_{\Delta} &= (e_{\Delta e l_i} \quad e_{\Delta e l_u} \quad e_{\Delta \omega_{1\theta}} \quad e_{\Delta \omega_{1C}} \quad e_{\Delta \xi_1} \quad e_{\Delta a_1})^T \\
 z_{\Delta} &= (z_{\Delta e l_i} \quad z_{\Delta e l_u} \quad z_{\Delta \omega_{1\theta}} \quad z_{\Delta \omega_{1C}} \quad z_{\Delta \xi_1} \quad z_{\Delta a_1})^T
 \end{aligned} \tag{2-168}$$

Et :

$$\left. \begin{aligned}
 A &= \begin{pmatrix} -\omega_{el0} & 0 & 0 & 0 & 0 & 0 & 0 & \omega_{el0} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ J^{-1}K_c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \omega_{10}^2 a_{10} K_c & 0 & 0 & -\omega_{10}^2 & -2\xi_{10}\omega_{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \omega_r^2 & 0 & \omega_r^2 & 0 & -\omega_r^2 & -2\xi_r\omega_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\tau_f^{-1} \end{pmatrix} \\
 B &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau_f^{-1} \end{pmatrix}, \quad M = \begin{pmatrix} \sqrt{\omega_{el1}} & \sqrt{\omega_{el1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2\omega_{11}} & \sqrt{a_{10}\omega_{10}2\omega_{11}} & \sqrt{2\xi_{11}\omega_{10}} & \sqrt{\omega_{10}a_{11}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 N &= \begin{pmatrix} -\sqrt{\omega_{el1}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{\omega_{el1}} \\ 0 & 0 & 0 & -\omega_{10}\sqrt{2\xi_{11}} & -\xi_{10}\sqrt{2\omega_{11}} & 0 & 0 & 0 \\ K_c\sqrt{a_{10}\omega_{10}2\omega_{11}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{2\xi_{11}\omega_{10}} & 0 & 0 & 0 \\ K_c\sqrt{\omega_{10}a_{11}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 C &= (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0), \quad D = 0
 \end{aligned} \right\} \tag{2-169}$$

La matrice d'incertitudes s'écrit :

$$\Delta = \text{diag}(\delta\omega_{el}, \delta\omega_{el}, \delta\omega_1, \delta\omega_1, \delta\xi_1, \delta a_1), \|\Delta\|_\infty < 1 \quad (2-170)$$

Noter que le système nominal est retrouvé avec $\Delta=0_{6 \times 6}$:

$$H_0(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (2-171)$$

Comme dans l'annexe A, noter que la variable d'état χ est introduite par la mise en place d'une constante de temps fictive (très haute fréquence $\tau_f \ll 1$) en entrée de modèle afin de rendre z_Δ strictement propre en la commande (Figure 2-27).

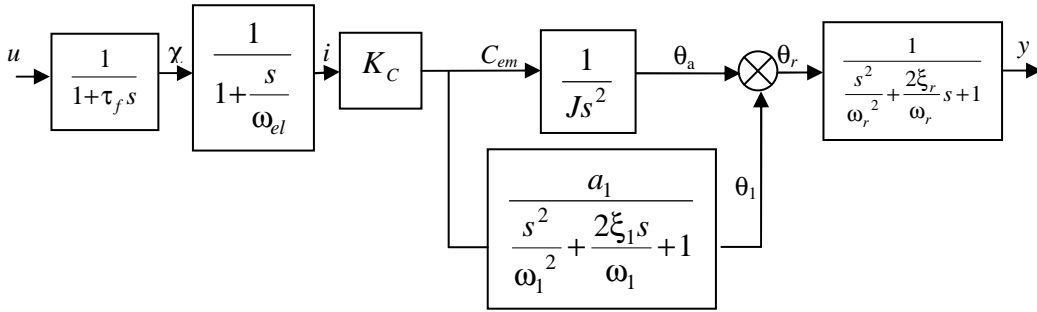


Figure 2-27: Système à mode souple avec constante de temps fictive en entrée

2.3.3.2.5.2.2 Mise en forme pour la μ -analyse

La seconde étape consiste à établir un schéma de μ -analyse de robustesse en performance par l'ajout d'une incertitude complexe fictive Δ_f entre les canaux de performances de la Figure 2-25, c'est-à-dire entre les vecteurs $(b,r)^T$ et $(\varepsilon, u_c)^T$, ce qui permet de construire le bloc d'incertitude augmentée $\Delta' = \text{diag}(\Delta_f, \Delta)$.

$$\Delta_f = \begin{pmatrix} \delta_{er} & \delta_{ur} \\ \delta_{eb} & \delta_{ub} \end{pmatrix} \quad (2-172)$$

L'incertitude fictive Δ_f est normalisée par l'introduction de coefficients de normalisation $(\alpha_r, \alpha_e, \alpha_b, \alpha_u)$ comme sur la Figure 2-28. La robustesse en performance est atteinte pour tout Δ telle que $\|\Delta\|_\infty < 1$ si le système de la Figure 2-25 reste stable pour tout Δ et Δ_f telles que $\|\Delta\|_\infty < 1$ et $\|\Delta_f\|_\infty < 1$, c'est-à-dire, d'après le théorème du petit gain si :

$$\left\| \begin{array}{cc} \alpha_e S_p \alpha_r & \alpha_b S_p H_p \alpha_e \\ \alpha_r K_p S_p \alpha_u & \alpha_b K_p S_p H_p \alpha_u \end{array} \right\|_\infty < 1 \quad (2-173)$$

Or le loop-shape est tel que :

$$\left\| \begin{array}{cc} S_p & S_p H_p \\ K_p S_p & K_p S_p H_p \end{array} \right\|_{\infty} < \gamma_{obj} \quad (2-174)$$

Il suffit donc de choisir :

$$\alpha_{\varepsilon} = \alpha_u = \alpha_b = \alpha_r = \gamma_{obj}^{-1/2} \quad (2-175)$$

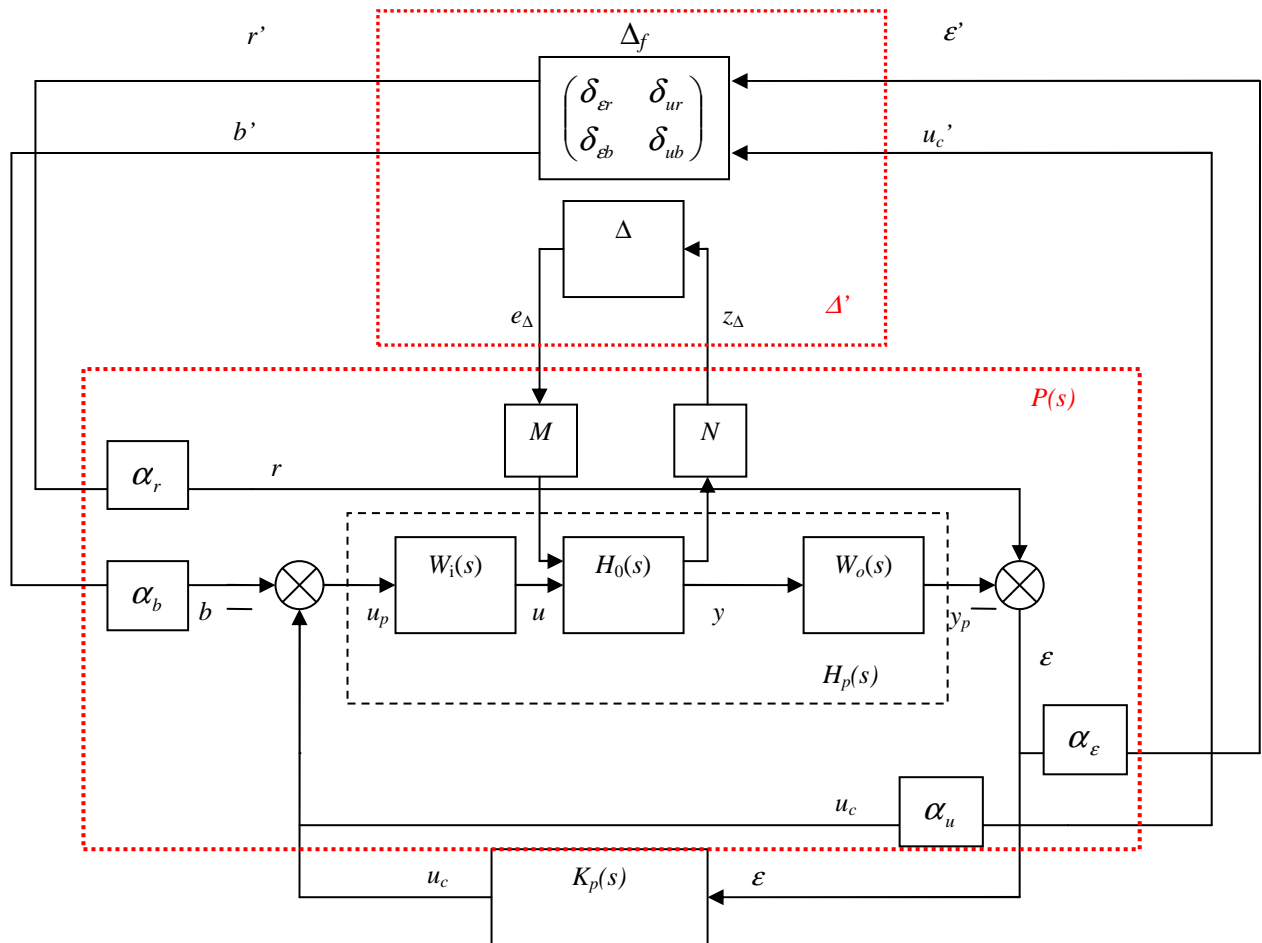


Figure 2-28: Schéma d'analyse de robustesse en performance

2.3.3.2.5.2.3 Mise en forme pour la μ -synthèse

La troisième étape consiste en la construction du schéma de commande pour la μ -synthèse de la Figure 2-29 par l'ajout des scalings $D(s)$ et $D(s)^{-1}$ (Figure 2-23).

Les scalings auront la structure suivante :

$$D(s) = \begin{pmatrix} \text{diag}(D_m(s), m = 1, \dots, 6) & 0 \\ 0 & I_{2 \times 2} \end{pmatrix} \tag{2-176}$$

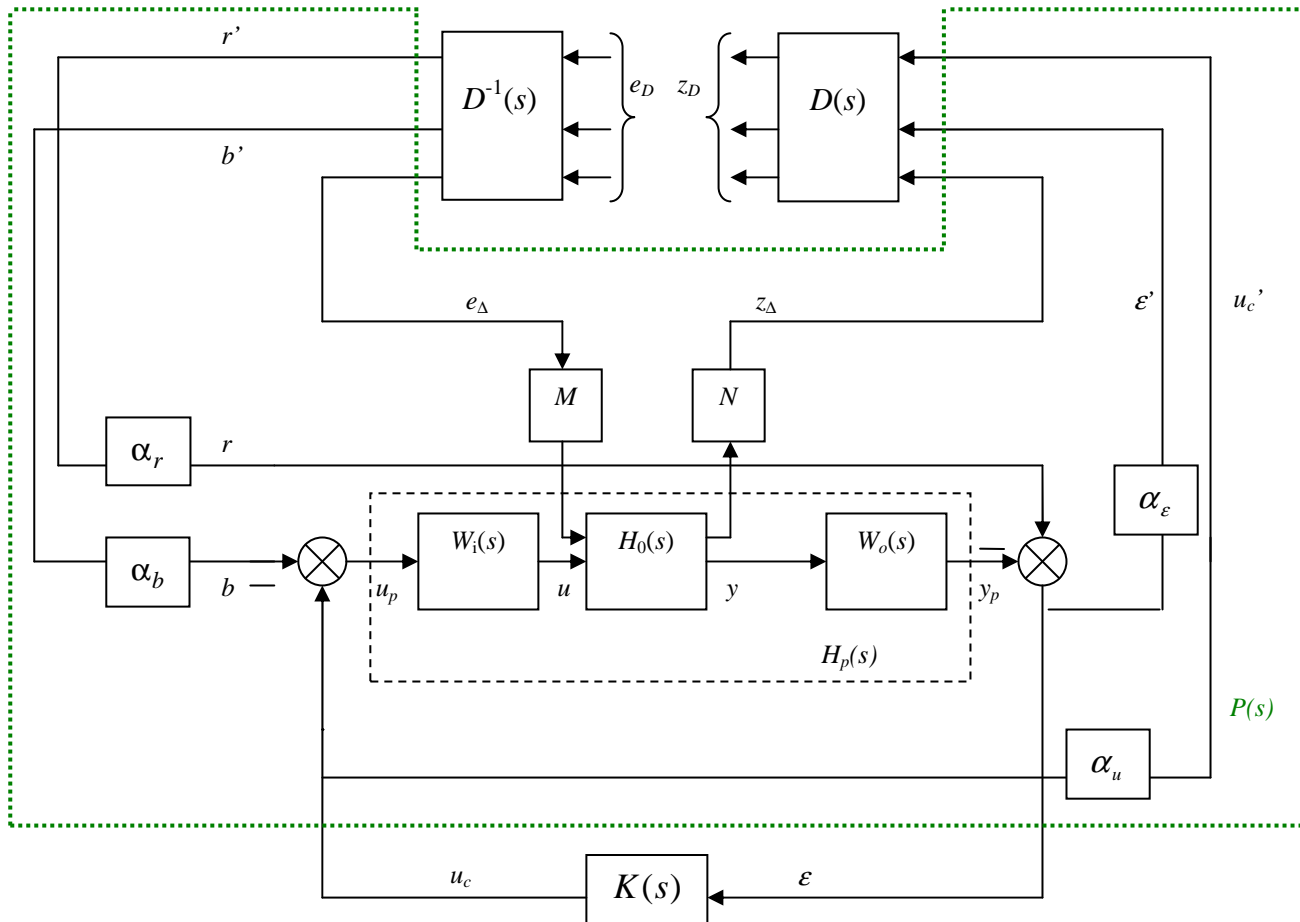


Figure 2-29: Schéma de commande pour la μ -synthèse

On montre facilement à partir de (2-167) et de (2-103) que le loop-shape mis sous forme LFT s'écrit :

$$\begin{aligned} \frac{dX_p}{dt} &= A_p X_p + B_p u_p + M_p e_\Delta \\ y_p &= C_p X_p + D_p u \\ z_\Delta &= N_p X_p \\ e_\Delta &= \Delta z_\Delta, \quad \|\Delta\|_\infty \leq 1 \end{aligned} \tag{2-177}$$

Avec :

$$A_p = \begin{bmatrix} A & BC_i & 0 \\ 0 & A_i & 0 \\ B_o C & B_o DC_i & A_o \end{bmatrix}, B_p = \begin{bmatrix} BD_i \\ B_i \\ B_o DD_i \end{bmatrix}, M_p = \begin{bmatrix} M \\ 0 \\ 0 \end{bmatrix} \quad (2-178)$$

$$C_p = [D_o C \quad D_o DC_i \quad C_o], D_p = D_o DD_i, N_p = [N \quad 0 \quad 0]$$

On a alors d'après la Figure 2-29:

$$\begin{bmatrix} \dot{X} \\ z_\Delta \\ \varepsilon \\ u_c \\ \varepsilon \end{bmatrix} = \left[\begin{array}{c|ccc|c} A_p & M_p & B_p & 0 & -B_p \\ \hline N_p & 0 & 0 & 0 & 0 \\ \hline -C_p & 0 & 0 & I & -D_p \\ \hline 0 & 0 & 0 & 0 & I \\ \hline -C_p & 0 & 0 & I & -D_p \end{array} \right] \begin{bmatrix} X \\ e_\Delta \\ b \\ r \\ u_c \end{bmatrix} \quad (2-179)$$

Par ailleurs :

$$\begin{aligned} r &= \alpha_r r' \\ b &= \alpha_b b' \\ \varepsilon &= \alpha_\varepsilon \varepsilon' \\ u_c &= \alpha_u u_c' \end{aligned} \quad (2-180)$$

On obtient finalement :

$$\begin{bmatrix} \dot{X}_p \\ z_\Delta \\ \varepsilon' \\ u_c' \\ \varepsilon' \end{bmatrix} = \left[\begin{array}{c|ccc|c} A_p & M_p & \alpha_b B_p & 0 & -B_p \\ \hline N_p & 0 & 0 & 0 & 0 \\ \hline -C_p \alpha_\varepsilon & 0 & 0 & I \alpha_r & -D_p \alpha_\varepsilon \\ \hline 0 & 0 & 0 & 0 & \alpha_u \\ \hline -\alpha_\varepsilon C_p & 0 & 0 & I \alpha_\varepsilon & -D_p \alpha_\varepsilon \end{array} \right] \begin{bmatrix} X_p \\ e_\Delta \\ b' \\ r' \\ u_c \end{bmatrix} \quad (2-181)$$

Qui est bien de la forme (2-154) avec :

$$\begin{aligned} e &= (b', r')^T, z = (\varepsilon', u_c')^T \\ y &= \varepsilon', u = u_c \end{aligned} \quad (2-182)$$

Le rajout des scalings conduit alors à la forme (2-155).

2.3.3.2.5.3 Résultats d'optimisation

On teste l'approche stochastique de la μ -synthèse pour différents ordres de scaling¹¹ $D(s)$: 0, 1 et 2. Pour chaque ordre de scaling, les différentes métaheuristiques

¹¹ Structurellement, chaque sous-bloc $D_m(s)$ constituant $D(s)$ est supposé disposer du même ordre que $D(s)$.

sont testées. Comme en (2-89), une évaluation statistique des résultats est nécessaire. Ainsi nous relevons la valeur minimale de γ obtenue (γ_{min}), la valeur moyenne de γ (γ_{mean}), son écart-type (γ_{std}). A titre informatif, on relève également le temps de calcul moyen par *run* ($t_{cpu/run}$) mis pour la réalisation des 10 *runs* constituant chaque test.

Les résultats sont présentés sur les Tableau 2.4, Tableau 2.5 et Tableau 2.6.

$\partial^{\circ}D = 0, IT / run = 100$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$ (min)
<i>StaticCEC_QPSO</i>	3,3302	3,3370	0,0078	18,7
<i>TVDdecCEC_QPSO</i>	3,3302	3,3486	0,0164	20,25
<i>Rnd_DE</i>	3,3536	3,5450	0,1044	19,30
<i>Rnd2Bst_DE</i>	3,3302	3,3894	0,0805	18,42
<i>Std_PSO</i>	3,3380	3,3888	0,0613	19,26
<i>TVRndIW_PSO</i>	3,3614	3,5411	0,1366	36,12
<i>ABC</i>	3,3302	3,3770	0,0301	39,47
<i>CS</i>	3,3302	3,4771	0,1787	41,9
<i>CS/DE</i>	3,3302	3,3962	0,0777	42,26

Tableau 2.4 Comparatif d'algorithmes pour la μ -synthèse d'ordre plein - $\partial^{\circ}D = 0$

$\partial^{\circ}D = 1, IT / run = 200$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$
<i>StaticCEC_QPSO</i>	1,2424	3,1497	0,6714	49,60 / 88,1
<i>TVDdecCEC_QPSO</i>	3,2650	3,4033	0,1371	49,97
<i>Rnd_DE</i>	4,0098	140,0991	381,7546	38,76
<i>Rnd2Bst_DE</i>	3,2286	3,3202	0,0391	48,35
<i>Std_PSO</i>	3,1964	295,3305	880,3437	52,99
<i>TVRndIW_PSO</i>	3,3681	11,3777	21,6602	47,02
<i>ABC</i>	4,3830	21,8867	49,7157	32,79
<i>CS</i>	3,2158	3,5808	0,2688	78,33
<i>CS/DE</i>	3,2484	3,4387	0,1524	82,83

Tableau 2.5 Comparatif d'algorithmes pour la μ -synthèse d'ordre plein - $\partial^{\circ}D = 1$

$\partial^{\circ}D = 2, IT / run = 300$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$
<i>StaticCEC_QPSO</i>	1,4449	3,0991	0,5883	238
<i>TVDdecCEC_QPSO</i>	3,1060	3,2951	0,2001	198
<i>Rnd_DE</i>	5,6416	206,2636	372,5146	62
<i>Rnd2Bst_DE</i>	3,2679	230,1133	647,4193	111,5
<i>Std_PSO</i>	3,2052	28,3716	46,6732	192,8
<i>TVRndIW_PSO</i>	3,4708	58,3784	89,8340	188
<i>ABC</i>				
<i>CS</i>	3,2616	4,7234	1,9278	325,2
<i>CS/DE</i>	3,3217	15,8320	36,5823	176,2

Tableau 2.6 Comparatif d'algorithmes pour la μ -synthèse d'ordre plein - $\partial^{\circ}D = 2$

Pour le problème de la μ -synthèse à ordre plein avec optimisation des scalings fréquentiels, les algorithmes à essaim particulière sont les plus performants. Plus particulièrement *StaticCEC_QPSO* fournit les résultats les plus compétitifs pour un temps de calcul par contre assez élevé.

2.3.4. μ -synthèse structurée

Comme on l'a vu dans les paragraphes précédents, la μ -synthèse conduit à des correcteurs d'ordre élevé. Il s'en suit en général une post-réduction mais se pose alors le problème de la robustesse du correcteur obtenu. Ainsi tout comme pour la synthèse H_∞ on peut chercher à contraindre *a priori* l'ordre du correcteur synthétisé.

Nous abordons là un domaine de la recherche encore ouvert.

2.3.4.1. μ -synthèse par l'optimisation non lisse

Hinfstruct permet de réaliser une μ -synthèse structurée [APK 11]. Remarquons tout d'abord que la condition de stabilité des scalings et de leur inverse n'est une nécessité que pour la résolution du problème (2-145) par les méthodes de calculs de correcteurs « classiques » (basée sur Riccati par exemple). A la base il suffit juste que (2-138) soit satisfaite. Ainsi on peut chercher à résoudre le problème suivant ; trouver un correcteur stabilisant $K(s)$ d'ordre n_k et une matrice de transfert $D(s)$ tels que :

$$\begin{aligned} - & D(s)\Delta'(s) = \Delta'(s)D(s), \forall \Delta'(s) \in \underline{\Delta}' \\ - & \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_\infty < 1 \end{aligned}$$

En posant :

$$\begin{aligned} \tilde{z}_D &= (z_{\Delta D} \quad z_D)^T \\ \tilde{e}_D &= (e_{\Delta D} \quad e_D)^T \end{aligned} \tag{2-183}$$

Alors on constate les relations d'équivalence de la Figure 2-30. Les problèmes suivants sont dès lors équivalents :

$$\begin{aligned} \min_{D(s), K(s)} & \left\| D(s)F_l(P(s), K(s))D(s)^{-1} \right\|_\infty \\ & \text{sous contraintes que} \\ & F_l(P, K) \in \mathbf{RH}_\infty \\ & \Leftrightarrow \\ \min_{K_c(s)} & \left\| F_l(P_c(s), K_c(s)) \right\|_\infty \\ & \text{sous contraintes que} \\ & F_l(P_c, K_c) \in \mathbf{RH}_\infty \end{aligned} \tag{2-184}$$

Avec (en supposant toujours que $n_z = n_e$) :

$$\begin{aligned}
D(s) &= \begin{pmatrix} \text{diag}(D_m(s), m = 1, \dots, n_\Delta) & 0 \\ 0 & I_{n_z} \end{pmatrix} \\
\underline{D}(s) &= D(s) - I \\
K_c(s) &= \begin{pmatrix} K(s) & & \\ & \underline{D}(s) & \\ & & \underline{D}(s) \end{pmatrix} \\
P_c(s) &:= L_c P(s) R_c + Q_c
\end{aligned} \tag{2-185}$$

$$:= \begin{bmatrix} I_{n_\Delta+n_z} & 0 \\ 0 & I_{n_y} \\ 0 & 0 \\ I_{n_\Delta+n_z} & 0 \end{bmatrix} P(s) \left[\begin{array}{c|cc} I_{n_\Delta+n_z} & 0 & -I_{n_\Delta+n_z} \\ \hline 0 & I_{n_u} & 0 \end{array} \right] + \left[\begin{array}{c|cc} 0 & 0 & 0 & I \\ \hline 0 & 0 & 0 & 0 \\ I & 0 & -I & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right]$$

On peut alors déterminer par ce biais le correcteur et les scalings simultanément par cette structuration décentralisée particulière.

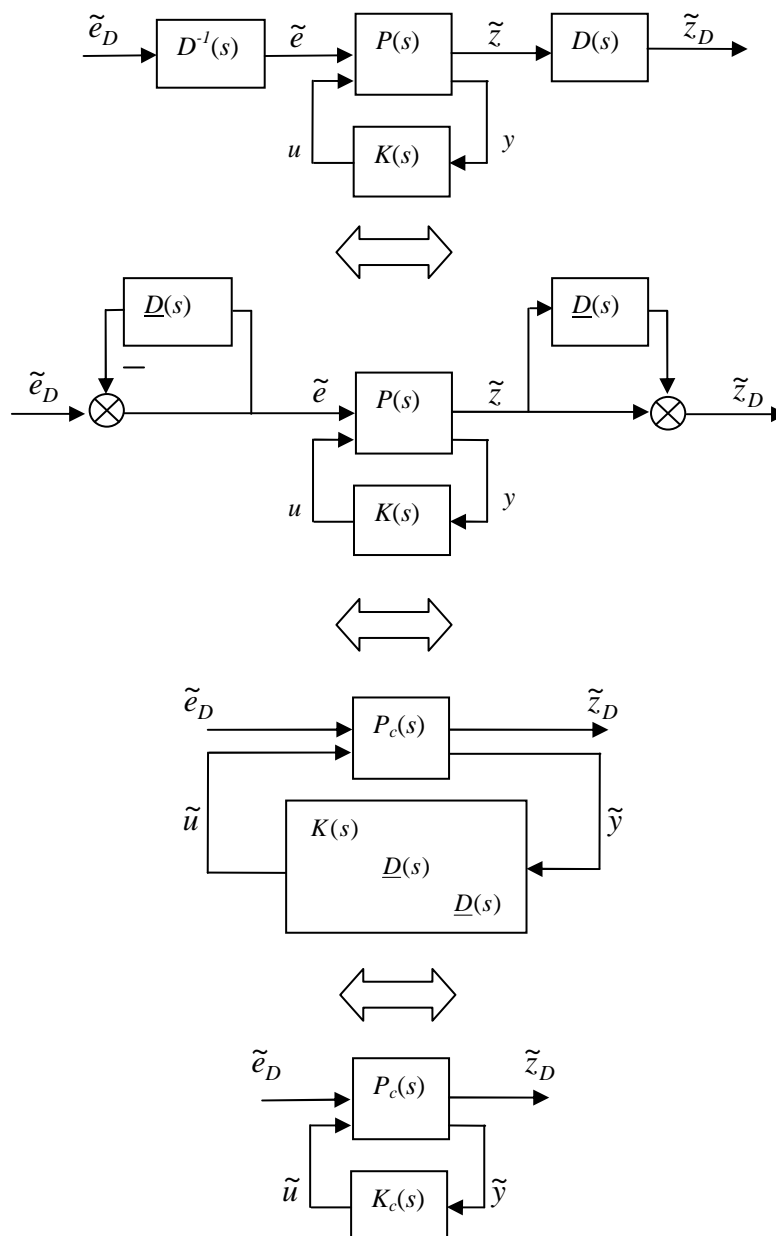


Figure 2-30: μ -synthèse par l'optimisation non lisse

2.3.4.2. Reformulation par l'optimisation stochastique

En observant les reformulations de la synthèse H_∞ structurée par l'optimisation stochastique (§2.2.4) et de la μ -synthèse (d'ordre plein) avec scaling dynamiques (§2.3.3.2), la reformulation de la μ -synthèse structurée par l'optimisation stochastique devient quasi-immédiate puisqu'elle consiste en la réunion de ces 2 problèmes. C'est l'approche que nous proposons dans [FEY 14c] comme une alternative à la μ -synthèse par l'optimisation non lisse précédente, l'intérêt étant la souplesse de l'approche et la possibilité de prendre en compte de nombreuses contraintes exotiques supplémentaires.

2.3.4.2.1. Principe de l'approche

On cherche à résoudre le problème suivant ; trouver un correcteur stabilisant $K(s)$ d'ordre n_k et une matrice de transfert $D(s)$ tels que :

- $D(s)\Delta'(s) = \Delta'(s)D(s), \forall \Delta'(s) \in \underline{\Delta}'$
- $\left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty} < 1$

Ce problème n'est ni plus ni moins que la réunion des problèmes (2-64) et (2-148), à la différence que comme vue pour l'approche non lisse, la stabilité des scalings n'est pas nécessaire.

En supposant $D(s)$ convenablement structurée, nous proposons ainsi ici de résoudre directement le problème d'optimisation globale suivant :

$$\begin{aligned} \min_{X \in [x; x]} f(X) \Leftrightarrow \min_{D(s), K(s)} \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty} \\ \text{sous contraintes que} \\ F_l(P, K) \in \mathbf{RH}_{\infty} \end{aligned} \quad (2-186)$$

En posant pour inconnue :

$$\Theta_D = \begin{pmatrix} A_D & B_D \\ C_D & D_D \end{pmatrix}, \Theta_k = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix} \quad (2-187)$$

Et :

$$\bar{\lambda}_{cl} = \max_i (\operatorname{Re}(\lambda_i(A_{cl}))) \quad (2-188)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Alors on trouve les scalings et le correcteur structuré μ -optimum en considérant le problème d'optimisation suivant :

$$\begin{aligned} \min_{\Theta_D, \Theta_k} f(\Theta_D, \Theta_k) \\ \text{si } \bar{\lambda}_{cl} \geq 0 \\ f(\Theta_D, \Theta_k) = \bar{\lambda}_{cl} \\ \text{sinon} \\ f(\Theta_D, \Theta_k) = -\frac{1}{\gamma}, \gamma = \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty} \end{aligned} \quad (2-189)$$

Notons que $\left\| \cdot \right\|_{\infty}$ désigne ici la norme L_{∞} [ZHO 96, p 114] évaluable pour un système instable contrairement à la norme H_{∞} qui nécessite la stabilité du système.

2.3.4.2.2. Mise en œuvre

Le scaling $D(s)$ a exactement la même structure que pour le §2.3.3.2.3.

Le correcteur $K(s)$ dispose de la même structure que pour le §2.2.4.3 Il peut être centralisé ou décentralisé.

Comme pour l'approche stochastique de la synthèse H_∞ , on accélère grandement le temps de convergence de l'algorithme en cherchant les scalings et le correcteur sous forme tridiagonale et en transformant l'espace de recherche (voir § 2.2.4.2).

Le vecteur de décision sera de la forme :

$$X = \begin{pmatrix} X_D \\ X_k \end{pmatrix} \quad (2-190)$$

Où X_D est défini en (2-151) et X_k défini en (2-76) pour un correcteur centralisé et en (2-79) pour un correcteur décentralisé.

En utilisant le fait que :

$$\begin{aligned} H_1 &= \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array} \right], \quad H_2 = \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array} \right] \\ H_1 H_2 &= \left[\begin{array}{cc|c} A_1 & B_1 C_2 & B_1 D_2 \\ 0 & A_2 & B_2 \\ \hline C_1 & D_1 C_2 & D_1 D_2 \end{array} \right] = \left[\begin{array}{cc|c} A_2 & 0 & B_2 \\ B_1 C_2 & A_1 & B_1 D_2 \\ \hline D_1 C_2 & C_1 & D_1 D_2 \end{array} \right] \end{aligned} \quad (2-191)$$

Et en posant :

$$\begin{aligned} F_l(P(s), K(s)) &= \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array} \right] \\ D(s) &= \left[\begin{array}{c|c} A_D & B_D \\ \hline C_D & D_D \end{array} \right] \\ D(s)^{-1} &= \left[\begin{array}{c|c} A_D - B_D D_D^{-1} C_D & -B_D D_D^{-1} \\ \hline D_D^{-1} C_D & D_D^{-1} \end{array} \right] = \left[\begin{array}{c|c} \underline{A}_D & \underline{B}_D \\ \hline \underline{C}_D & \underline{D}_D \end{array} \right] \end{aligned} \quad (2-192)$$

Alors, on montre facilement que la représentation d'état de la forme standard augmentée des scalings $D(s)P(s)D(s)^{-1}$ de la Figure 2-23 s'écrit :

$$\left[\begin{array}{ccc|c} A_D & B_D C_{cl} & B_D D_{cl} C_D & B_D D_{cl} D_D \\ 0 & A_{cl} & B_{cl} C_D & B_{cl} D_D \\ 0 & 0 & A_D & B_D \\ \hline C_D & D_D C_{cl} & D_D D_{cl} C_D & D_D D_{cl} D_D \end{array} \right] \quad (2-193)$$

2.3.4.2.3. Evaluation de la fonction de coût

Nous cherchons à résoudre le problème d'optimisation suivant :

$$\min_{X \in [\underline{x}; \bar{x}]} f(X) \quad (2-194)$$

Où la variable X est définie en (2-151) et où la fonction $f(X)$ consiste à réaliser les étapes suivantes :

- construire $\left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right]$ à partir de X et du paragraphe 2.2.4.3 ;
- construire la matrice A_{cl} de la boucle fermée à l'aide de (2-86) ;
- évaluer l'abscisse spectrale de la boucle fermée :

$$\bar{\lambda} = \max_i (\operatorname{Re}(A_{cl})) \quad (2-195)$$

- Si $\bar{\lambda} \geq 0$ alors $f(X) = \bar{\lambda}$,
- Sinon :
 - construire la boucle fermée $F_l(P, K)$ à l'aide de (2-88) ;
 - construire les réalisations $\left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & D_m \end{array} \right], m=1, \dots, n_\Delta$ à partir de X et de (2-149) à (2-152) ;
 - construire les réalisations $\left[\begin{array}{c|c} A_D & B_D \\ \hline C_D & D_D \end{array} \right]$ et $\left[\begin{array}{c|c} \underline{A}_D & \underline{B}_D \\ \hline \underline{C}_D & \underline{D}_D \end{array} \right]$ avec (2-152) ;
 - construire la réalisation de $D(s)F_l(P(s), K(s))D(s)^{-1}$ à l'aide de (2-193) ;
 - évaluer $f(X) = -\frac{1}{\gamma}$, $\gamma = \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_\infty$.

2.3.4.2.4. Prise en compte de contraintes supplémentaires

Comme pour la synthèse H_∞ structurée formulé au §2.2.4.6, un intérêt de cette approche est sa souplesse et la prise en compte possible de contraintes complexes.

2.3.4.2.4.1 Contrainte sur la stabilité du correcteur

On peut chercher à résoudre le problème suivant ; trouver un correcteur stabilisant $K(s)$ d'ordre n_k et une matrice de transfert $D(s)$ tels que :

$$- D(s)\Delta'(s) = \Delta'(s)D(s), \forall \Delta'(s) \in \underline{\Delta}'$$

- $\|D(s)F_l(P, K)D(s)^{-1}\|_\infty < 1$,
- $K(s)$ soit stable.

En posant :

$$\bar{\lambda}_k = \max_i (\operatorname{Re}(\lambda_i(A_k))) \quad (2-196)$$

Où $\lambda_i(\cdot)$ est la $i^{\text{ème}}$ valeur propre.

Alors on synthétise le correcteur μ -optimal en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_D, \Theta_k} f(\Theta_D, \Theta_k) \\ \text{si } & \max(\bar{\lambda}_{cl}, \bar{\lambda}_k) \geq 0 \\ & f(\Theta_D, \Theta_k) = \max(\bar{\lambda}_{cl}, \bar{\lambda}_k) \\ \text{sinon} & \\ & f(\Theta_D, \Theta_k) = -\frac{1}{\gamma}, \quad \gamma = \|D(s)F_l(P, K)D(s)^{-1}\| \end{aligned} \quad (2-197)$$

2.3.4.2.4.2 Contrainte sur la position des pôles en boucle fermée

On peut chercher à résoudre le problème suivant ; trouver un correcteur stabilisant $K(s)$ d'ordre n_k et une matrice de transfert $D(s)$ tels que :

- $D(s)\Delta'(s) = \Delta'(s)D(s), \forall \Delta'(s) \in \underline{\Delta}'$
- $\|D(s)F_l(P, K)D(s)^{-1}\|_\infty < 1$,
- Les pôles de la boucle fermée $F_l(P, K)$ soient tous à partie réelle inférieure à α (négatif).

Alors on synthétise le correcteur μ -optimal en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_D, \Theta_k} f(\Theta_D, \Theta_k) \\ \text{si } & \bar{\lambda}_{cl} \geq \alpha \\ & f(\Theta_D, \Theta_k) = \bar{\lambda}_{cl} - \alpha \\ \text{sinon} & \\ & f(\Theta_D, \Theta_k) = -\frac{1}{\gamma}, \quad \gamma = \|D(s)F_l(P, K)D(s)^{-1}\| \end{aligned} \quad (2-198)$$

2.3.4.2.4.3 Contrainte sur la norme d'un transfert particulier

On peut chercher à résoudre le problème suivant ; trouver un correcteur stabilisant $K(s)$ d'ordre n_k et une matrice de transfert $D(s)$ tels que :

- $D(s)\Delta'(s) = \Delta'(s)D(s), \forall \Delta'(s) \in \underline{\Delta}'$
- $\left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty} < 1,$
- $\|M(s)\|_{2 \text{ ou } \infty} < \gamma_0.$

Où $M(s)$ est un transfert bouclé quelconque.

Alors on synthétise le correcteur μ -optimal en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_D, \Theta_k} f(\Theta_D, \Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq \alpha \\ & f(\Theta_D, \Theta_k) = \bar{\lambda}_{cl} \\ & \text{sinon} \end{aligned} \tag{2-199}$$

$$f(\Theta_D, \Theta_k) = - \frac{1}{\gamma + e^{\alpha(\|M(s)\|_{2 \text{ ou } \infty} - \gamma_0)}}, \gamma = \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty}, \alpha \gg 1$$

2.3.4.2.5. Exemple

On reprend à l'identique l'exemple §2.3.3.2.5. On rappelle que nous avons cherché lors de la synthèse multi-modèles du §2.2.4.7.1.2 un correcteur K_p d'ordre 5. Afin de comparer les performances de ces 2 techniques structurées, nous procédons ici de même avec des scalings d'ordre 0,1 et 2.

Les calculs sont réalisés avec Matlab 2011b sur un COREi7 à 2.3 GHz avec 8 Gb Ram.

Une évaluation statistique des résultats est nécessaire. Ainsi comme en (2-89), nous relevons la valeur minimale de γ obtenue (γ_{min}), la valeur moyenne de γ (γ_{mean}), son écart-type (γ_{std}). A titre informatif, on relève aussi le temps de calcul moyen par *run* ($t_{cpu/run}$) mis pour la réalisation des 10 *runs* constituant chaque test.

Les résultats sont présentés sur les Tableau 2.7, Tableau 2.8 et Tableau 2.9.

Au regard du Tableau 2.7 et du Tableau 2.8, on constate que comme pour la synthèse H_{∞} structurée, **Rnd2Bst_DE** est le plus performant (parmi les algorithmes testés) pour le problème de la μ -synthèse structurée.

Comme le montre le Tableau 2.7, les scalings statiques ne disposent pas assez d'ordre pour disposer d'une valeur de γ convenable et les performances obtenues à

l'ordre 1 sont bien meilleures sur le Tableau 2.8 ; Sur le Tableau 2.9, on augmente l'ordre des scaling à 2. On constate qu'on est alors plus performant vis-à-vis de la valeur de γ obtenue mais au prix d'un temps de calcul conséquent.

$\partial^\circ D = 0, IT / run = 5000$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$ (min)
<i>StaticCEC_QPSO</i>	3,67	3,92	0,15	11,79
<i>TVDecCEC_QPSO</i>	3,89	4,12	0,26	5,52
<i>Rnd_DE</i>	3,64	3,96	0,29	8,95
<i>Rnd2Bst_DE</i>	3,55	3,88	0,22	9,93
<i>Std_PSO</i>	4,71	790600	2482600	13,35
<i>TVRndIW_PSO</i>	4,05	563540	1782100	11,23
<i>ABC</i>				
<i>CS</i>	3,99	28138	87043	3,55
<i>CS/DE</i>	3,57	3,92	0,17	19,10

Tableau 2.7 Comparatif d'algorithmes pour la μ -synthèse structurée -
 $\partial^\circ D = 0, \partial^\circ K_p = 5$

$\partial^\circ D = 1, IT / run = 20000$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$ (min)
<i>StaticCEC_QPSO</i>	3,41	3,74	0,28	94,31
<i>TVDecCEC_QPSO</i>	3,41	3,65	0,17	46,23
<i>Rnd_DE</i>	3,22	3,38	0,16	72,14
<i>Rnd2Bst_DE</i>	2,94	3,14	0,15	95,34
<i>Std_PSO</i>	3,15	4,06	1,48	96,45
<i>TVRndIW_PSO</i>	3,20	7,38	12,31	85,22
<i>ABC</i>	3,75	4,35	0,41	76,49
<i>CS</i>	4,15	4,51	0,21	20,49
<i>CS/DE</i>	3,16	3,31	0,15	163,02

Tableau 2.8 Comparatif d'algorithmes pour la μ -synthèse structurée -
 $\partial^\circ D = 1, \partial^\circ K_p = 5$

$\partial^\circ D = 2, IT / run = 40000$	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$ (min)
<i>Rnd2Bst_DE</i>	2,82	2,99	0,12	318,88

Tableau 2.9 Comparatif d'algorithmes pour la μ -synthèse structurée -
 $\partial^\circ D = 2, \partial^\circ K_p = 5$

En guise de comparaison, nous procédons aux mêmes tests avec l'optimisation non lisse (*HinfStruct*). Comme pour les tests précédents, on procède à 10 *runs* successifs et nous comparons aux résultats obtenus avec *Rnd2Bst_DE*. Les scalings et le correcteur sont aussi cherchés sous forme tridiagonale. Les réglages de *HinfStruct* sont considérés par défaut.

	γ_{min}	γ_{mean}	γ_{std}	$t_{cpu/run}$ (min)
$\partial^\circ D = 0$	3,28	3,67	0,14	0,14
$\partial^\circ D = 1$	3,54	4,10	0,35	0,17
$\partial^\circ D = 2$	3,63	4,21	0,32	0,24

Tableau 2.10 Approche non-lisse pour la μ -synthèse à ordre fixé - $\partial^\circ K_p = 5$

Si on compare les deux approches (évolutionnaire et non-lisse), l'approche évolutionnaire fournit de meilleurs résultats mais en usant de temps de calcul beaucoup plus longs. L'approche évolutionnaire peut constituer une alternative intéressante à l'approche non-lisse de par sa flexibilité et sa facilité de mise en œuvre ; en effet, rien n'empêche la mise en œuvre de contraintes exotiques (stabilité du correcteur, ...) comme nous l'avons fait pour la synthèse H_∞ structurée au §2.2.4.6. Enfin l'approche évolutionnaire peut fournir un point d'initialisation à l'approche non-lisse.

Partie 3

Réglage optimal des correcteurs H_∞ structurés vis-à-vis d'exigences de haut-niveau.

3.1. Introduction

3.1.1. Problématique

On s'intéresse dans cette partie à un autre inconvénient majeur inhérent aux méthodes de synthèse robuste avancées modernes : le réglage des pondérations requiert un certain degré de savoir-faire de la part de l'ingénieur. En effet, partant d'une spécification haut niveau souvent complexe portant sur des grandeurs quelconques, une première étape consiste souvent à simplifier cette dernière (procédure de linéarisation, approximations, ...) afin d'en déduire une spécification fréquentielle sous forme de gabarit de l'asservissement. En fonction de l'expérience plus ou moins grande de l'ingénieur, il s'en suit un nombre plus ou moins important d'itérations de synthèse car le correcteur synthétisé sur cette spécification fréquentielle doit satisfaire aussi, et c'est le plus important, la spécification haut niveau ; ces itérations peuvent donc occasionner des surcoûts de développement importants. Ainsi dans un souci de développer des asservissements robustes et performants à coût maîtrisé, on se propose dans cette partie de travailler sur une méthode de réglage optimal des correcteurs H_∞ directement vis-à-vis des spécifications industrielles complexes haut niveau en évitant la phase de déclinaison fréquentielle intermédiaire coûteuse en temps de développement.

Une des missions principales de l'automaticien est de déterminer une loi de commande bouclant un système complexe car potentiellement non linéaire afin de satisfaire une spécification complexe car potentiellement non linéaire portant sur des grandeurs quelconques Z_1, Z_2, \dots en réponse à des entrées quelconques, perturbations, fonctionnalités, etc.... Cette spécification rassemble très souvent un nombre fini m d'exigences du type (3-1).

$$Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \quad (3-1)$$

En général, cette spécification industrielle « haut niveau » porte sur un système potentiellement de grande série fonctionnant sur une plage environnementale plus ou moins large (environnement vibratoire, thermique, etc...), avec des conditions de mise en œuvre plus ou moins variées ; par la loi de commande adéquate, l'automaticien doit donc stabiliser le système avec un certain degré de robustesse sous contraintes de satisfaire les exigences haut-niveau (3-1). Cette mission peut être résumée sur la Figure 3-1.

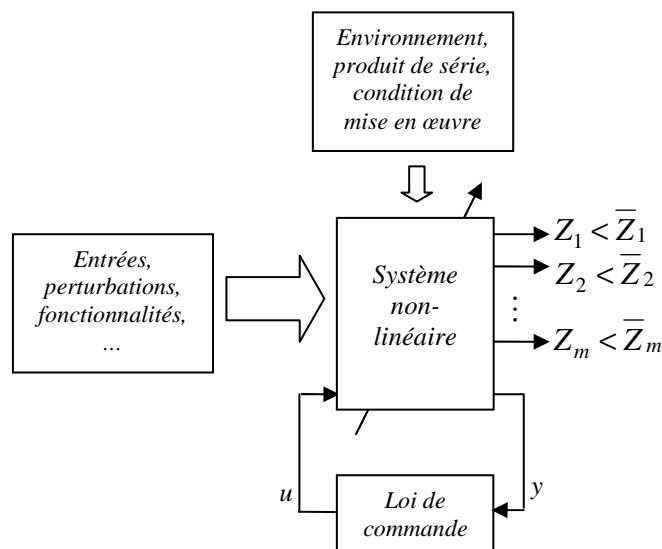


Figure 3-1: Spécifications système haut niveau

La complexité de ce problème empêche la plupart du temps l'emploi des outils linéaires connus pour le calcul du correcteur ; aussi procède-t-on en premier lieu à une phase de simplification de la spécification (linéarisations, etc...) portant sur un système lui aussi linéarisé¹² dont les incertitudes sont vues comme des perturbations internes. Cette spécification « simplifiée » prend alors la forme de gabarits fréquentiels portant sur différents transferts de boucle entre des entrées exogènes e et des sorties à surveiller z . Sur la forme standard de la Figure 3-2, le vecteur e représente les entrées exogènes (consignes, perturbations,...), z représente les signaux à surveiller (erreurs de suivi, commandes,...), y représente les mesures à disposition du correcteur pour le calcul de la commande u .

¹² Il est à noter que l'écart entre le système vrai et son linéarisé peut aussi être pris en compte dans le bloc d'incertitudes portant sur le système

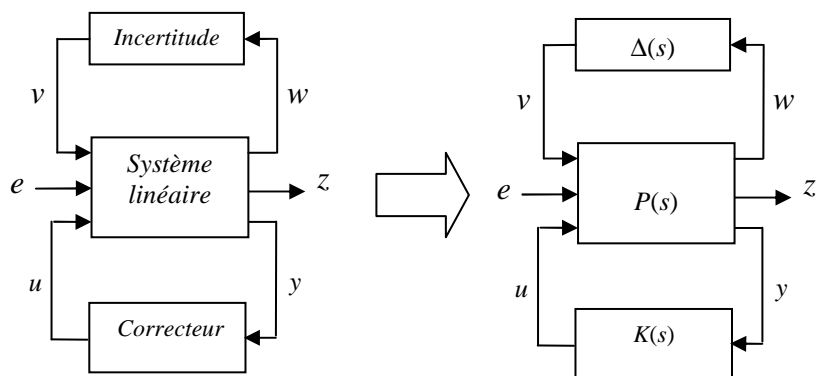


Figure 3-2: Forme standard pour la commande

Dans les années 1990, l'automatique fait un grand bond en avant avec l'apparition des techniques de synthèse de correcteurs basées sur la norme H_∞ :

- d'une part, la prise en compte d'un cahier des charges fréquentiel devient possible par l'emploi de pondérations fréquentielles $W_o(s)$ et $W_i(s)$ respectivement sur les entrées exogènes et les signaux à surveiller (Figure 3-3) puis par la minimisation de la norme H_∞ du transfert entre ces mêmes signaux à l'aide d'un correcteur stabilisant solution du problème (3-2) dont la résolution dans l'espace d'état a été explicitement formulée dans [DOY 89], et issue d'une dichotomie sur la résolution d'équations de Riccati (γ -itération).

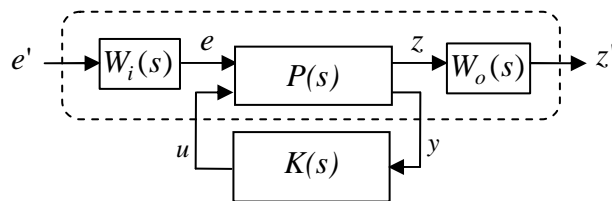


Figure 3-3: Approche standard

$$K(s) = \arg \min \gamma, \quad \|W_o(s)F_l(P, K)W_i(s)\|_\infty < \gamma \quad (3-2)$$

- d'autre part, le théorème du petit gain donne une condition nécessaire et suffisante assurant la stabilité du bouclage obtenu pour toute incertitude $\Delta(s)$ telle que $\|\Delta(s)\|_\infty < \gamma^{-1}$. Ce dernier est stable si et seulement si $\|T_{vw}(s)\|_\infty < \gamma$, permettant ainsi la prise en compte a priori d'objectifs de robustesse lors du choix des pondérations pour la synthèse du correcteur (Figure 3-4).

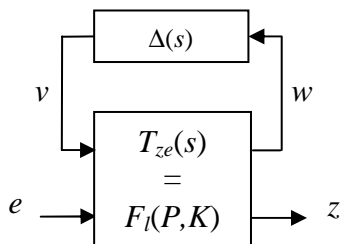


Figure 3-4: Forme standard pour l'analyse de robustesse

Ainsi avec l'approche standard de la commande robuste, la complexité du calcul du correcteur, basé jusqu'alors classiquement sur l'examen de la boucle ouverte, a été reportée sur la détermination de pondérations fréquentielles pertinentes qui contribuent au premier ordre aux performances du correcteur final. La difficulté que représente le réglage de ces pondérations et le décalage conceptuel vis-à-vis de l'automatique fréquentielle classique rend la méthode vite complexe à mettre en œuvre à cause du savoir-faire qu'elle nécessite. De plus le nombre de degrés de liberté de réglage croît rapidement avec le nombre de mesures disponibles, le nombre de signaux exogènes et le nombre de signaux à surveiller (au minimum, une pondération à régler par signal supplémentaire). Si la spécification ne porte réellement que sur certains transferts de boucle, le nombre total de transferts de boucle définis par la forme standard (et donc à régler) devient vite très important (présence de termes croisés, etc...).

Dans le même temps que l'approche standard, fut publiée la solution explicite du problème de stabilisation robuste d'un système modélisé par ses factorisations premières [MCF 90] selon le schéma de la Figure 3-5 :

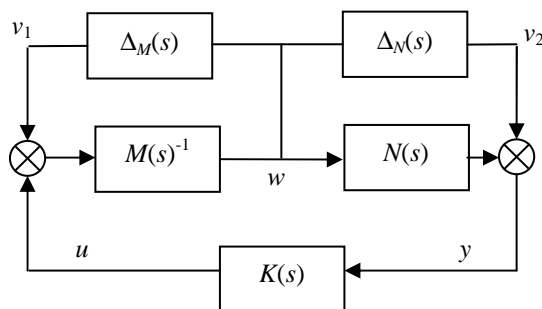


Figure 3-5: Stabilisation robuste d'un système modélisé par ses facteurs premiers.

- Très séduisante par sa simplicité, elle consiste en la résolution de seulement deux équations de Riccati de type LQG. Sous sa forme 4-blocs équivalente, elle est un cas particulier de l'approche standard de la commande robuste H_∞ présentée précédemment. En remarquant que l'on peut modéliser les fonctions de sensibilité directe et complémentaire par un modelage de la boucle ouverte et en voyant que tout transfert de boucle est proportionnel à ces mêmes fonctions de sensibilité, il est alors possible de modéliser tout transfert de boucle en travaillant sur un transfert unique, la boucle ouverte, ce qui définit les fondements de la synthèse par *loop-shaping*. Elle consiste en un modelage des valeurs singulières de la boucle ouverte par des pondérations en entrée et

en sortie du système (respectivement $W_o(s)$ et $W_i(s)$ sur la Figure 3-6), constituant un *loop-shape* pour lequel un correcteur stabilisant peut être calculé, définissant ainsi la synthèse H_∞ par *loop-shaping* ; le correcteur est alors calculé en résolvant le problème (3-3).

$$\begin{aligned}
 H_p(s) &= W_o(s)H(s)W_i(s) \\
 K_p(s) &= \arg \min \gamma, \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_\infty < \gamma \\
 K(s) &= W_i(s)K_p(s)W_o(s)
 \end{aligned}
 \tag{3-3}$$

- d'autre part, grâce à la notion de gap-métrique (qui traduit mathématiquement une distance entre deux systèmes) et au théorème du petit gain, la robustesse de la stabilité de la boucle vis-à-vis des incertitudes sur les facteurs premiers peut être évaluée avant même d'avoir calculé explicitement le correcteur.

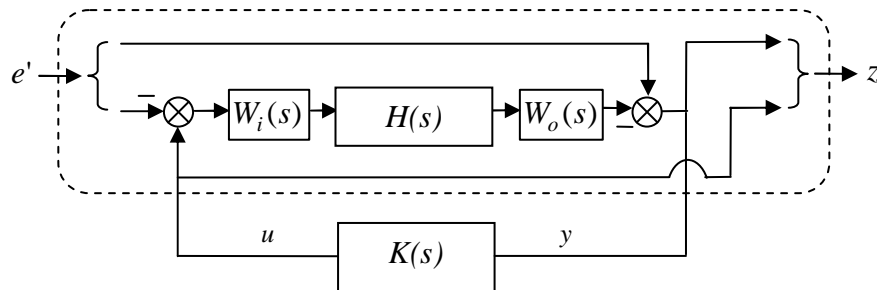


Figure 3-6: Approche par *loop-shaping*

Contrairement à l'approche standard, le nombre de degrés de liberté de réglage est contraint par les dimensions du système ce qui rend l'approche beaucoup plus simple. De plus on peut régler les pondérations par les concepts de l'automatique fréquentielle classique tout en disposant des qualités de robustesse inhérentes aux techniques H_∞ .

Cependant, pour les deux approches, le réglage des pondérations fréquentielles $W_o(s)$ et $W_i(s)$ reste le point dur (en termes de difficulté et de coût de développement). En effet, lors du calcul du correcteur, si γ est trop grand cela signifie que les gabarits fréquentiels en boucle fermée ou le *loop-shape* sont trop contraignants et qu'il faut les retoucher¹³. Se pose alors la question de leur adéquation avec la spécification haut-niveau (3-1) et le compromis de robustesse. Quand γ est convenable, l'ingénieur doit alors valider que le correcteur issu du réglage de ces pondérations satisfait au final (et cela constitue l'objectif prioritaire) la spécification (3-1) associée aux contraintes de robustesse précédemment énoncées. Cette phase de validation est particulièrement coûteuse car elle peut faire intervenir des simulations et des validations expérimentales. Si le correcteur n'est pas satisfaisant, alors le jeu de pondérations $W_o(s)$ et $W_i(s)$ doit être modifié et ainsi de suite.

¹³ Noter qu'à gabarit fréquentiel fixé, le réglage des pondérations est lui-même un processus itératif plus ou moins long

Par cette approche itérative coûteuse décrite sur la Figure 3-7, l'automaticien cherche donc le jeu de pondérations linéaires qui satisfait au mieux la spécification système haut-niveau (3-1) tout en assurant le degré de robustesse désiré. Il cherche donc à résoudre le problème suivant :

$$\begin{aligned} & \min_{W_o(s), W_i(s)} \gamma \\ & \text{sous contraintes que :} \\ & K(s) \text{ assure la robustesse désirée et} \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned} \quad (3-4)$$

Notons que l'emploi de l'approche standard de la synthèse H_∞ nécessite la stabilité des pondérations $W_o(s)$ et $W_i(s)$. Cette contrainte est levée dans le cadre de l'approche loop-shaping. Toutefois, on a tendance à implémenter des correcteurs stables (aux intégrateurs purs près). Le correcteur final étant déterminé par recombinaison du précorrecteur (stable) et des pondérations, on a donc tendance à imposer aussi la stabilité des pondérations dans l'approche loop-shaping (hors intégrateur pur).

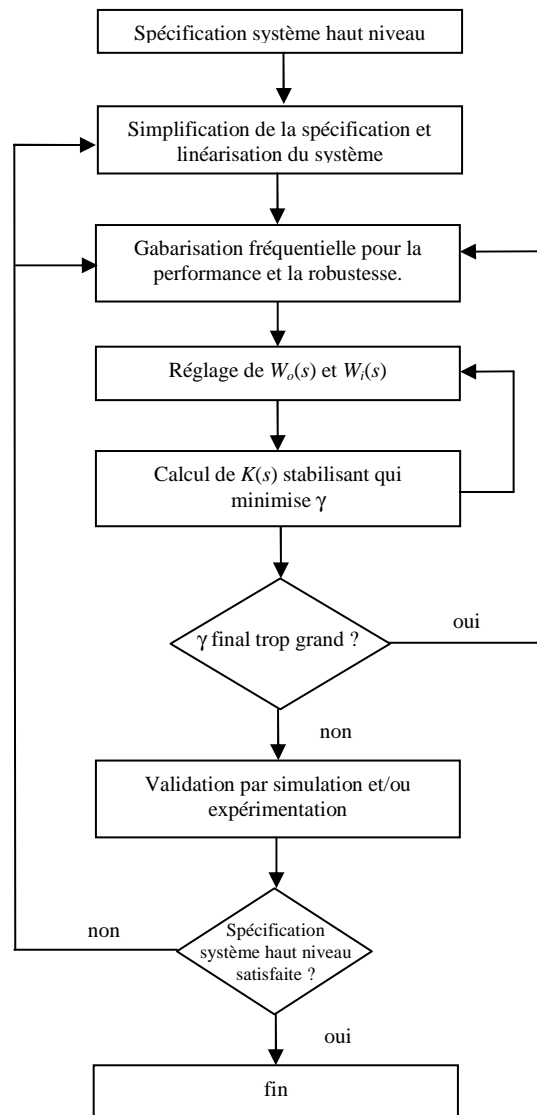


Figure 3-7: Processus itératif de réglage de correcteurs H_∞

Dans le but d'éviter la phase coûteuse de déclinaison fréquentielle des spécifications, on se propose dans cette partie d'utiliser l'optimisation stochastique pour attaquer le problème (3-4) de manière frontale comme sur la Figure 3-8.

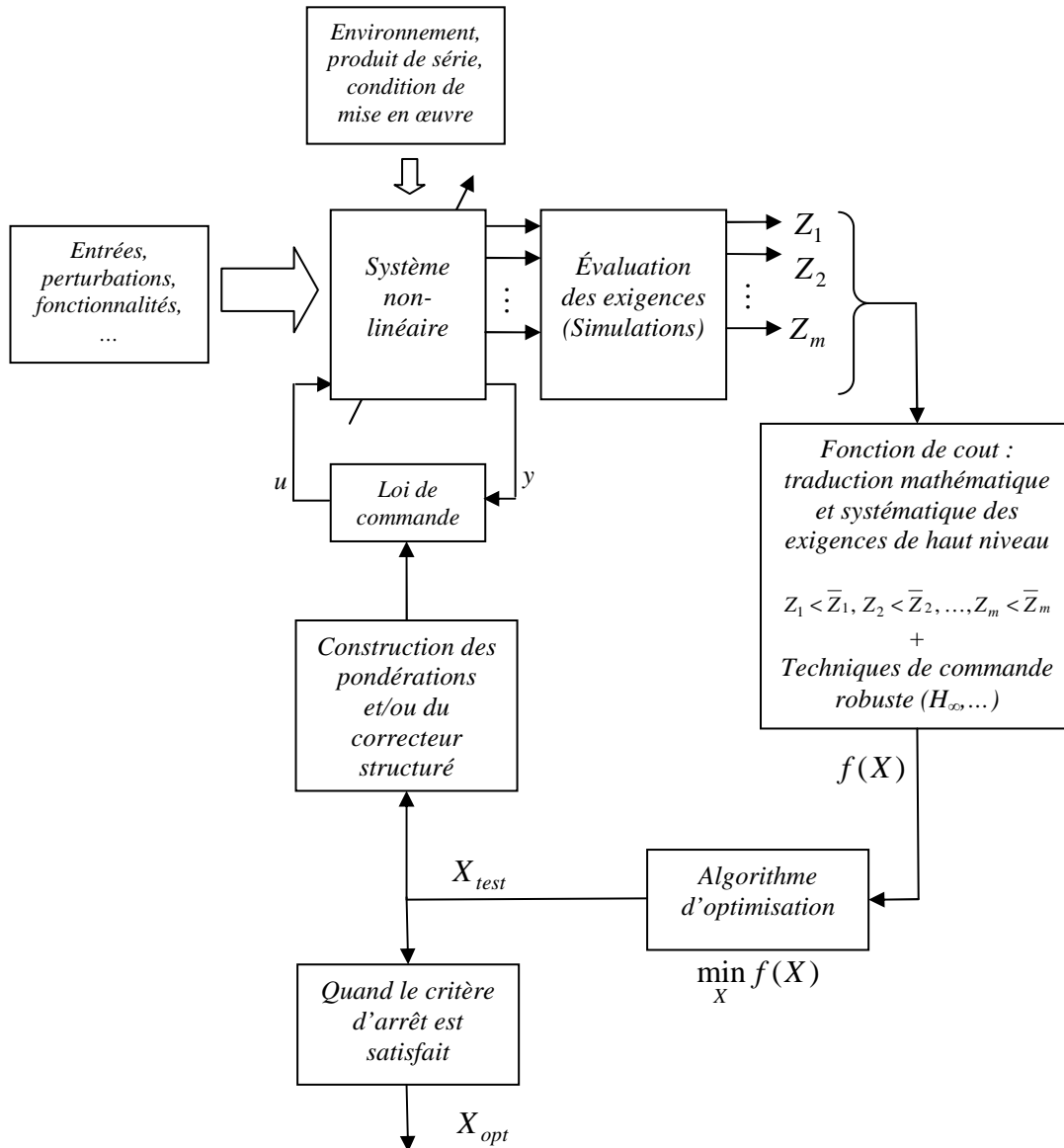


Figure 3-8: Optimisation « haut niveau »

Dans la littérature on trouve plusieurs travaux intéressants se rapportant à ce sujet. Par exemple dans [CHI 96], des pondérations d'ordre faible sont réglées avec un algorithme évolutionnaire et [SAN 10] montre qu'une spécification complexe avec des non-linéarités telles que des saturations peut être efficacement optimisée avec les essais particulières. Ces travaux montrent clairement que des critères complexes peuvent être considérés même si leur gradient n'est pas exploitable (car trop complexe à formuler ou non défini partout), la seule contrainte étant la possibilité d'évaluer la fonction de coût considérée. Bien que d'un intérêt certain, la mise en œuvre de ces

travaux souffre de deux inconvénients majeurs que nous nous proposons de lever dans ce travail :

- Tout d'abord, la fonction de coût à optimiser est obtenue en remplaçant le problème d'optimisation sous contraintes (3-4) par un problème d'optimisation sans contrainte. Ceci est possible par l'emploi de pénalités. Or on constate à l'usage que le réglage de ces pénalités est délicat car il influence grandement le processus d'optimisation : clairement, ce réglage fait déjà appel à un certain savoir-faire de la part de l'ingénieur. Partant du principe que remplacer un problème de réglage de pondérations par un réglage de pénalités n'a qu'un intérêt limité surtout dans le cas de nombreuses contraintes, nous proposons ici une méthode systématique et générique de réglage des pénalités de la fonction de coût.
- Les pondérations recherchées sont souvent d'ordre faible et structurées en pôles / zéros / amortissements. L'avantage est la positivité de tous les paramètres recherchés réduisant ainsi l'espace de recherche. L'inconvénient est que cette structuration a priori des pondérations recherchée nécessite un savoir-faire de l'ingénieur. De fait, imposer une telle structure aux pondérations ne permet pas de garantir l'obtention de la meilleure solution possible au problème (3-4), à ordre de pondération fixé. Nous proposons ici de rechercher les pondérations directement dans l'espace d'état en employant la forme tridiagonale et d'améliorer la rapidité du processus d'optimisation par la transformation de l'espace de recherche : ces techniques ont déjà été vues en première partie. De même dans le cadre de pondérations MIMO, une structure centralisée ou non peut être envisagée.

Nous privilégierons l'approche par *loop-shaping* car cette dernière s'avère bien plus adaptée pour ce besoin d'optimisation que l'approche standard et ce, pour les raisons suivantes :

- A ordre plein, le calcul du précorrecteur $K_p(s)$ est directement déduit de la résolution de deux équations de Riccati contrairement à l'approche standard basée sur la γ -itération, qui consiste en une dichotomie sur la résolution d'équations de Riccati. En termes de temps de calcul, l'approche *loop-shaping* s'avère donc beaucoup plus adaptée à l'emploi de l'optimisation stochastique qui demande un nombre important d'évaluations de la fonction de coût suivant des lois d'évolution particulières.
- Comme vu précédemment le nombre de degrés de liberté de l'approche est contraint par les dimensions du système. Le nombre de variables de décision est donc de fait beaucoup plus faible par cette approche que pour l'approche standard, ce qui s'avère déterminant pour le processus d'optimisation.
- Une propriété fondamentale de l'approche *loop-shaping* est que les marges de stabilité du bouclage sont directement reliées à la valeur de γ [VIN 00]. On rappelle que :

$$\Delta G \geq \frac{1 + \gamma^{-1}}{1 - \gamma^{-1}}, \Delta \Phi \geq 2 \arcsin(\gamma^{-1}) \quad (3-5)$$

Où ΔG et $\Delta\Phi$ sont respectivement les marges de gain et de phases (généralisées dans le cas MIMO). Pour $\gamma = 3$, on obtient $\Delta G \geq 6dB$, $\Delta\Phi \geq 39^\circ$. Ainsi en minimisant γ par (3-4), on est sûr de maximiser simultanément ΔG et $\Delta\Phi$ alors qu'il faudrait mettre en œuvre des contraintes supplémentaires pour régler les marges de stabilité avec l'approche standard tel que cela est fait dans [SAN 10].

Ainsi, dans le cadre loop-shaping, on note la reformulation suivante :

$$\begin{aligned} & \min_{W_o(s), W_i(s)} \gamma \\ & \text{sous contraintes que :} \\ & K(s) \text{ assure la robustesse désirée et} \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \\ & \Rightarrow \end{aligned} \tag{3-6}$$

$$\begin{aligned} & \max_{W_o(s), W_i(s)} \{\Delta G, \Delta\phi\} \\ & \text{sous contraintes que :} \\ & K(s) \text{ assure la robustesse désirée et} \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned}$$

Dans le paragraphe suivant on présente succinctement l'approche de la synthèse H_∞ par *loop-shaping*.

3.2. La synthèse H_∞ par loop-shaping

On présente ici succinctement l'approche loop-shaping de la synthèse H_∞ . Bien qu'elle tire son origine du problème de stabilisation robuste d'un système modélisé par ses factorisations premières [MCF 90]-[VIN 00], il est possible de présenter l'approche sans faire appel à ces notions [MCF 92].

3.2.1. Principe de l'approche

Le principe de l'approche *loop-shaping* est de procéder à un *modelage de la boucle ouverte* par la mise en place de *pondérations fréquentielles* telles que sur le schéma 4-blocs de la Figure 3-9.

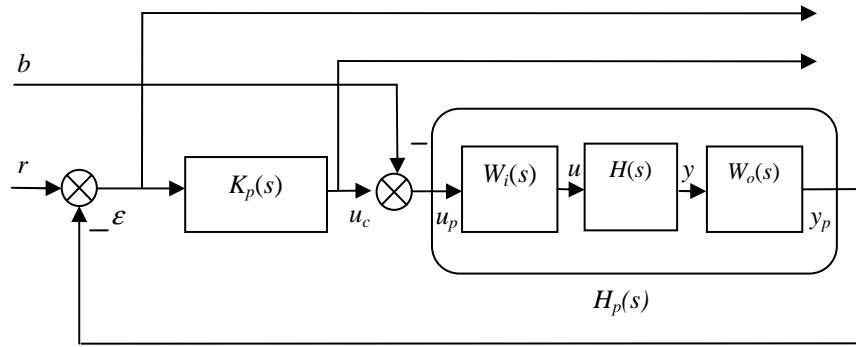


Figure 3-9: Synthèse par loop-shaping

On pose $H_p(s) = W_o(s)H(s)W_i(s)$. $H_p(s)$ est appelé *loop-shape*. Le *loop-shape* est composé du système $H(s)$ pré-compensé d'une pondération en entrée $W_i(s)$ et post-compensé d'une pondération en sortie $W_o(s)$.

y_p est le vecteur des mesures et u_p le vecteur des commandes pour le système pondéré¹⁴. Il est de plus à noter que b et r sont des entrées externes analogues aux bruits d'état et d'observation du filtre de Kalman de la commande LQG [KWA 72].

Dans ce schéma, r est voulu ici comme représentant un signal de référence (cas concret le plus fréquent) : il peut tout aussi bien représenter un bruit sur la mesure, et dans ce cas ε est la mesure bruitée.

Il est établi que :

$$\begin{pmatrix} \varepsilon(s) \\ u_c(s) \end{pmatrix} = \begin{pmatrix} S_p(s) & S_p(s)H_p(s) \\ K_p(s)S_p(s) & K_p(s)S_p(s)H_p(s) \end{pmatrix} \begin{pmatrix} r(s) \\ b(s) \end{pmatrix} \quad (3-7)$$

Avec $S_p(s)$, fonction de sensibilité directe en sortie, définie par :

$$S_p(s) = (I + H_p(s)K_p(s))^{-1} \quad (3-8)$$

Le but est de chercher un correcteur $K_p(s)$ stabilisant et assurant pour une valeur de $\gamma > 0$ donnée :

$$\left\| \begin{pmatrix} S_p(s) & S_p(s)H_p(s) \\ K_p(s)S_p(s) & K_p(s)S_p(s)H_p(s) \end{pmatrix} \right\|_{\infty} < \gamma \quad (3-9)$$

Supposons avoir calculé un tel correcteur $K_p(s)$; dans ce cas la boucle ouverte en sortie¹⁵ s'écrit :

¹⁴ Tous les signaux relatifs au *loop-shape* sont indicés « p ».

¹⁵ C'est-à-dire en ouvrant la boucle en sortie de $H_p(s)$.

$$H_{BO_y}(s) = H_p(s)K_p(s) \quad (3-10)$$

On montre alors [FEY 13c] les résultats suivants :

- Dans les zones où $\overline{\sigma}(H_{BO_y}(j\omega)) \ll 1$ (par exemple en hautes fréquences) :

$$\overline{\sigma}(H_{BO_y}(j\omega)) < \gamma \overline{\sigma}(H_p(j\omega)) \quad (3-11)$$

- Dans les zones où $\underline{\sigma}(H_{BO_y}(j\omega)) \gg 1$ (par exemple en basses fréquences) alors :

$$\underline{\sigma}(H_{BO_y}(j\omega)) \geq \frac{1}{\gamma} \underline{\sigma}(H_p(j\omega)) \quad (3-12)$$

Le *loop-shape* constitue donc un *modélage objectif de la boucle ouverte*. En effet, les valeurs singulières de la boucle ouverte suivent celles du *loop-shape* décrit par $H_p(s)$ au rapport γ près et on peut montrer [MCF 92] que ce rapport tend asymptotiquement vers la valeur $\sqrt{\gamma^2 - 1}$ si les pondérations du *loop-shape* sont telles que $\lim_{\omega \rightarrow \infty} \overline{\sigma}(H_p(j\omega)) = 0$ et $\lim_{\omega \rightarrow 0} \underline{\sigma}(H_p(j\omega)) = \infty$.

Enfin γ sera d'autant plus grand que le *loop-shape* est contraignant, la boucle ouverte s'éloignant alors davantage de ce dernier ; γ est donc un indicateur de performance de la tenue du *loop-shape*.

Finalement un correcteur $K(s)$ est obtenu en combinant les pondérations et le précorrecteur stabilisant le *loop-shape* :

$$K(s) = W_i(s)K_p(s)W_o(s) \quad (3-13)$$

Le raisonnement et les résultats précédents peuvent être reconduits aisément sur la boucle ouverte en entrée $H_{BO_u}(s) = K_p(s)H_p(s)$.

3.2.2. Marge de gain et marge de phase

$\left\| (I_p + H_p(s)K_p(s))^{-1} \right\|_{\infty} < \gamma$ fournit une garantie de marge de stabilité (en d'autres termes dans le cas mono variable la marge de module est supérieure à γ^{-1}) et donne une indication sur les valeurs de marge de gain et de phase généralisées dans le cas MIMO.

Si l'on note ΔG la marge de gain et $\Delta\Phi$ la marge de phase, alors on a :

$$\begin{aligned}\Delta G &\geq \frac{1 + \gamma^{-1}}{1 - \gamma^{-1}} \\ \Delta\Phi &\geq 2 \arcsin(\gamma^{-1})\end{aligned}\tag{3-14}$$

Ainsi avec la valeur usuelle de 3 pour γ , on obtient $\Delta G \geq 6dB$, $\Delta\Phi \geq 39^\circ$.

3.2.3. Interprétation 4-blocs de la méthode

En supposant les pondérations inversibles :

$$\begin{aligned}H_p(s) &= W_o(s)H(s)W_i(s) \\ K_p(s) &= W_i^{-1}(s)K(s)W_o^{-1}(s)\end{aligned}\tag{3-15}$$

En les supposant de plus diagonales alors on a [MCF 92] :

$$\begin{aligned}&\left\| \left(\begin{bmatrix} I \\ K_p(s) \end{bmatrix} S_{p_y}(s) \begin{bmatrix} I & H_p(s) \end{bmatrix} \right) \right\|_\infty \\ &= \left\| \left(\begin{bmatrix} I \\ W_i^{-1}(s)K(s)W_o^{-1}(s) \end{bmatrix} (I + W_o(s)H(s)K(s)W_o^{-1}(s))^{-1} \begin{bmatrix} I & W_o(s)H(s)W_i(s) \end{bmatrix} \right) \right\|_\infty \\ &= \left\| \left(\begin{bmatrix} W_o(s) \\ W_i^{-1}(s)K(s) \end{bmatrix} W_o^{-1}(s) (I + W_o(s)H(s)K(s)W_o^{-1}(s))^{-1} W_o(s) \begin{bmatrix} W_o(s)^{-1} & H(s)W_i(s) \end{bmatrix} \right) \right\|_\infty \\ &= \left\| \left(\begin{bmatrix} W_o(s) \\ W_i^{-1}(s)K(s) \end{bmatrix} (I + H(s)K(s))^{-1} \begin{bmatrix} W_o(s)^{-1} & H(s)W_i(s) \end{bmatrix} \right) \right\|_\infty\end{aligned}\tag{3-16}$$

Ainsi l'on obtient un critère 4-blocs équivalent dans lequel l'adjonction de pondérations pour modeler le *loop-shape* revient à fixer des pondérations sur les entrées exogènes et les signaux à surveiller :

$$\left\| \left(\begin{bmatrix} W_o(s) \\ W_i^{-1}(s)K(s) \end{bmatrix} (I + H(s)K(s))^{-1} \begin{bmatrix} W_o(s)^{-1} & H(s)W_i(s) \end{bmatrix} \right) \right\|_\infty < \gamma\tag{3-17}$$

On a donc l'équivalence des schémas de commande de la Figure 3-10.

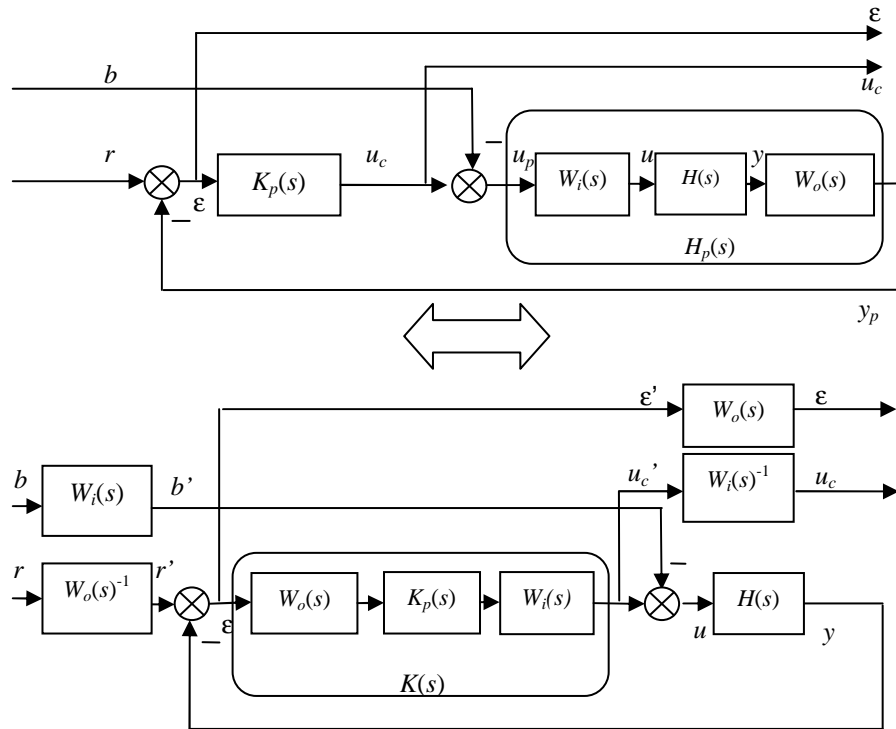


Figure 3-10: Interprétation 4-blocs de la méthode

La norme H_∞ des fonctions de sensibilité directe et complémentaire est donnée à l'avance par la valeur de γ , permettant ainsi d'apprécier les marges de stabilité du bouclage avant le calcul du correcteur. Par ailleurs, il faut noter que cette formulation nécessite que les pondérations satisfassent quelques hypothèses : stables et propres et à inverses stables et propres, hypothèse qu'il n'est pas nécessaire de satisfaire dans la formulation *loop-shape*.

3.2.4. Mise en œuvre pratique

Spécifications

Il est convenu de disposer d'une spécification *loop-shaping* sous la forme :

$$\begin{aligned}
 \underline{\sigma}(H_{BO}(j\omega))_{BF} &> |W_{BF}(j\omega)| \\
 \omega_{BP} &> \omega_{BP \min} \\
 \Delta\Phi &> \Delta\Phi_{\min} \\
 \Delta G &> \Delta G_{\min} \\
 \overline{\sigma}(H_{BO}(j\omega))_{HF} &< |W_{HF}(j\omega)|
 \end{aligned} \tag{3-18}$$

Où ω_{BP} est la bande passante escomptée.

Choix des pondérations

Les pondérations doivent être choisies de sorte que le système augmenté $H_p(s) = W_o(s)H(s)W_i(s)$ présente une forme satisfaisante, conformément au *loop-shaping* spécifié lors d'une déclinaison fréquentielle des spécifications. Le choix des pondérations se fait suivant les règles de l'automatique fréquentielle traditionnelle¹⁶ :

- en général, on place dans $W_o(s)$ la composante « basses fréquences » du modelage c'est-à-dire les intégrateurs éventuels, avances de phase, ... de sorte que la valeur singulière inférieure du *loop-shape* en basses fréquences soit telle que $\underline{\sigma}(H_p(j\omega)) > |W_{BF}(j\omega)|$. En supposant qu'en basses fréquences $H(s) \approx H_{BF}(s)$, alors un premier choix raisonnable peut être tel que :

$$\underline{\sigma}(W_o(j\omega)H_{BF}(j\omega)) \geq |W_{BF}(j\omega)| \quad (3-19)$$

- en général, on place dans $W_i(s)$ les composantes de *roll-off*, c'est-à-dire les filtrages hautes fréquences de sorte que la valeur singulière supérieure du *loop-shape* en hautes fréquences soit telle que $\overline{\sigma}(H_p(j\omega)) < |W_{HF}(j\omega)|$. En supposant qu'en hautes fréquences $H(s) \approx H_{HF}(s)$, alors un premier choix raisonnable peut être tel que :

$$\overline{\sigma}(H_{HF}(j\omega)W_i(j\omega)) \leq |W_{HF}(j\omega)| \quad (3-20)$$

Un cas particulier très fréquent est celui où H est de gain faible en hautes fréquences (la plupart des systèmes physiques sont passe-bas). Dans ce cas, on pourra simplement choisir :

$$\overline{\sigma}(W_i(j\omega)) = |W_{HF}(j\omega)| \quad (3-21)$$

- les pondérations sont placées de sorte que le module du *loop-shape* présente une forme satisfaisante sans se préoccuper de la phase, gérée pour une bonne partie par la synthèse H_∞ et on se contentera de faire tendre au voisinage de la bande passante le *loop-shape* vers un intégrateur ;
- lorsque le *loop-shape* dispose d'un certain nombre de pôles instables p_i et/ou d'un certain nombre de zéros instables z_i , alors la bande passante atteignable ω_{BP} est telle que [SKO 01] :

$$\max_{\text{Re}(p_i) > 0} (|p_i|) < \omega_{BP} < \min_{\text{Re}(z_i) > 0} (|z_i|) \quad (3-22)$$

- le réglage des pondérations se réalise avec les acquis de l'automatique fréquentielle classique (type réglage de PID) : bien que disposant de moins de flexibilité que la synthèse quatre blocs standard, elle fait le lien entre l'automatique fréquentielle classique et la commande robuste ;

¹⁶ Le domaine des basses fréquences est défini par $\omega < \omega_{BP}$. Le domaine des hautes fréquences est défini par $\omega > \omega_{BP}$.

- enfin, aucune restriction n'est faite *quant à la stabilité* des pondérations qui peuvent notamment contenir des intégrateurs purs.

Une fois les pondérations définies, il ne reste plus qu'à calculer le précorrecteur par les formules suivantes.

Calcul du précorrecteur

Le calcul se fait en résolvant le problème H_∞ présenté précédemment et vis-à-vis de la Figure 3-9, on pose :

$$\left. \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array} \right\} = H(s), \dim(A) = n$$

$$\left. \begin{array}{l} \dot{x}_i(t) = A_i x_i(t) + B_i u_p(t) \\ u(t) = C_i x_i(t) + D_i u_p(t) \end{array} \right\} = W_i(s), \dim(A_i) = n_i \quad (3-23)$$

$$\left. \begin{array}{l} \dot{x}_o(t) = A_o x_o(t) + B_o y(t) \\ y_p(t) = C_o x_o(t) + D_o y(t) \end{array} \right\} = W_o(s), \dim(A_o) = n_o$$

Le *loop-shape* dispose de la représentation d'état suivante :

$$\frac{d}{dt} \begin{pmatrix} x \\ x_i \\ x_o \end{pmatrix} = \begin{pmatrix} A & BC_i & 0 \\ 0 & A_i & 0 \\ B_o C & B_o DC_i & A_o \end{pmatrix} \begin{pmatrix} x \\ x_i \\ x_o \end{pmatrix} + \begin{pmatrix} BD_i \\ B_i \\ B_o DD_i \end{pmatrix} u_p(t) \quad (3-24)$$

$$y_p(t) = \begin{pmatrix} D_o C & D_o DC_i & C_o \end{pmatrix} \begin{pmatrix} x \\ x_i \\ x_o \end{pmatrix} + D_o DD_i u_p(t)$$

Noté :

$$\frac{dx_p}{dt} = A_p x_p + B_p u_p(t) \quad (3-25)$$

$$y_p(t) = C_p x_p + D_p u_p(t)$$

Et :

$$\dim(A_p) = n_p = n_i + n_o + n \quad (3-26)$$

On suppose que la paire (A_p, B_p) est stabilisable et que la paire (C_p, A_p) est détectable. Dans ce cas, le pré correcteur est directement calculé par la résolution des deux équations de Riccati suivantes :

$$\begin{aligned}
S &= I + D_p^T D_p \\
R &= I + D_p D_p^T \\
X \left(A_p - B_p S^{-1} D_p^T C_p \right) + \left(A_p - B_p S^{-1} D_p^T C_p \right)^T X - X B_p S^{-1} B_p^T X + C_p^T R^{-1} C_p &= 0 \\
Y \left(A_p - B_p D_p^T R^{-1} C_p \right)^T + \left(A_p - B_p D_p^T R^{-1} C_p \right) Y - Y C_p^T R^{-1} C_p Y + B_p S^{-1} B_p^T &= 0
\end{aligned} \quad (3-27)$$

Le coefficient γ_{opt} est calculé par la relation :

$$\gamma_{opt} = \sqrt{1 + \lambda_{sup}(YX)} \quad (3-28)$$

Pour un coefficient $\gamma_0 = \varepsilon \gamma_{opt} > \gamma_{opt}$ (typiquement $\varepsilon = 1.01$), le précorrecteur est alors déterminé par :

$$K_p(s) = \left[\begin{array}{c|c} A_{K_p} & B_{K_p} \\ \hline C_{K_p} & D_{K_p} \end{array} \right] \quad (3-29)$$

Avec :

$$\left. \begin{aligned}
A_{K_p} &= A_p + B_p F + \gamma_{opt}^2 Z Y C_p^T (C_p + D_p F) \\
B_{K_p} &= -\gamma_{opt}^2 Z Y C_p^T \\
C_{K_p} &= B_p^T X \\
D_{K_p} &= D_p^T \\
Z &= (I + YX - \gamma_{opt}^2 I)^{-1} \\
F &= -S^{-1} (D_p^T C_p + B_p^T X)
\end{aligned} \right\} \quad (3-30)$$

Le calcul du correcteur ne fait pas appel à la γ -itération et est donc direct.

Calcul de la boucle ouverte et du correcteur

Le précorrecteur $K_p(s)$ ayant été calculé précédemment, la structure de correction retenue $K(s)$ s'obtient en combinant les pondérations et le précorrecteur :

$$K(s) = W_i(s) K_p(s) W_o(s) \quad (3-31)$$

La boucle ouverte (en entrée) s'écrit :

$$H_{BO_Y}(s) = W_i(s) K_p(s) W_o(s) H(s) = K(s) H(s) \quad (3-32)$$

Remarques pour la résolution

- Les pondérations et le système pouvant être d'ordre important, on peut normaliser en fréquence afin de diminuer les problèmes calculatoires lors de la résolution des équations de Riccati.
- Soit $M(s)$ une matrice de transfert. On désigne par $\partial^\circ M$ l'ordre de M . L'ordre du précorrecteur est égal à celui de du système pondéré c'est-à-dire :

$$\partial^\circ K_p = \partial^\circ H + \partial^\circ W_i + \partial^\circ W_o \Rightarrow \partial^\circ K = \partial^\circ H + 2\partial^\circ W_i + 2\partial^\circ W_o \quad (3-33)$$

3.2.5. Implémentation des correcteurs

Plusieurs modes d'implémentation existent. Nous retiendrons l'implémentation fréquentielle et l'implémentation sous forme d'observateur/retour d'état.

3.2.5.1. Implémentation fréquentielle

C'est une implémentation fréquente. Dans la mesure où la stabilité de la boucle est entièrement régie par la boucle ouverte, qui, quelle que soit l'implémentation du correcteur, dispose toujours de la même expression, plusieurs implémentations sont possibles. En général on privilégie pour correcteur le produit en série du précorrecteur et des pondérations :

$$K(s) = W_i(s)K_p(s)W_o(s) \quad (3-34)$$

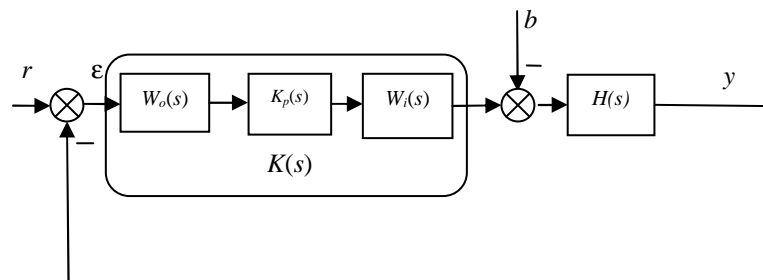


Figure 3-11: Implémentation fréquentielle du correcteur

Cependant en fonction des besoins, les pondérations de *roll-off* composant $W_i(s)$ peuvent être placées dans le retour comme par exemple sur la Figure 3-12.

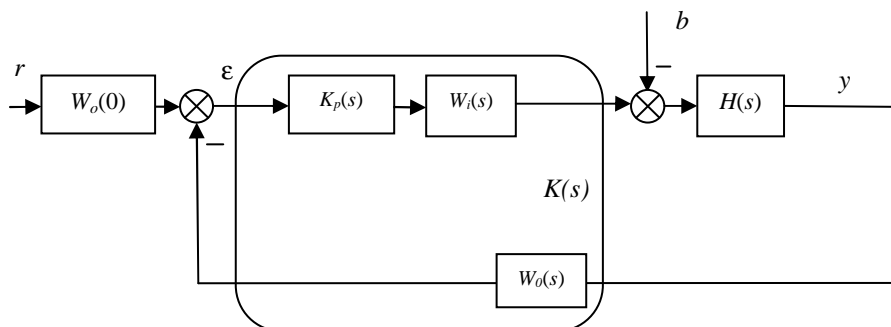


Figure 3-12: Implémentation fréquentielle particulière

Cette représentation permet par exemple d'augmenter la bande passante en boucle fermée du bouclage vis-à-vis de l'entrée r car la boucle fermée se comporte en basse fréquence comme l'inverse du retour de sortie, c'est à-dire $W_o(s)^{-1}$.

Signalons enfin dans [SKO 01] la possibilité d'implémenter simplement une structure d'*anti-windup* sur la pondération $W_o(s)$ lorsque cette dernière contient les intégrateurs et est à déphasage non minimal.

3.2.5.2. Implémentation observateur/retour d'état

La synthèse H_∞ par loop-shaping rend possible par nature et de manière immédiate l'implémentation sous forme observateur/retour d'état : on trouve souvent ce type d'implémentation dans l'aéronautique où on utilise l'observateur aussi à des fins de détection de défauts.

Soit le *loop-shape* :

$$\left. \begin{aligned} \frac{dx_p}{dt} &= A_p x_p + B_p u_p \\ y_p &= C_p x_p + D_p u_p \end{aligned} \right\} = H_p(s) \quad (3-35)$$

On suppose disposer d'une commande par retour d'état stabilisante :

$$u_p = K_c x_p \quad (3-36)$$

Lorsque l'état n'est pas complètement mesurable, on peut chercher à le reconstruire par l'observateur suivant :

$$\left. \begin{aligned} \frac{d\hat{x}_p}{dt} &= A_p \hat{x}_p + B_p u_p + K_f (\hat{y}_p - y_p) \\ \hat{y}_p &= C_p \hat{x}_p + D_p u_p \end{aligned} \right\} \quad (3-37)$$

L'emploi de l'état estimé pour la commande conduit à :

$$\left. \begin{aligned} \frac{d\hat{x}_p}{dt} &= A_p \hat{x}_p + B_p K_c \hat{x}_p + K_f (C_p \hat{x}_p + D_p K_c \hat{x}_p - y_p) \\ u_p &= K_c \hat{x}_p \end{aligned} \right\} \quad (3-38)$$

C'est-à-dire :

$$\left. \begin{aligned} \frac{d\hat{x}_p}{dt} &= (A_p + B_p K_c + K_f (C_p + D_p K_c)) \hat{x}_p - K_f y_p \\ u_p &= K_c \hat{x}_p \end{aligned} \right\} \quad (3-39)$$

Le pré correcteur peut donc être implémenté sous la forme retour d'état observateur en choisissant [VIN 00] :

$$\begin{aligned} K_f &= \gamma_{opt}^2 Z Y C_p^T \\ K_c &= F \end{aligned} \quad (3-40)$$

Et en complétant la commande de la composante propre du système, on obtient alors le schéma de la Figure 3-13.

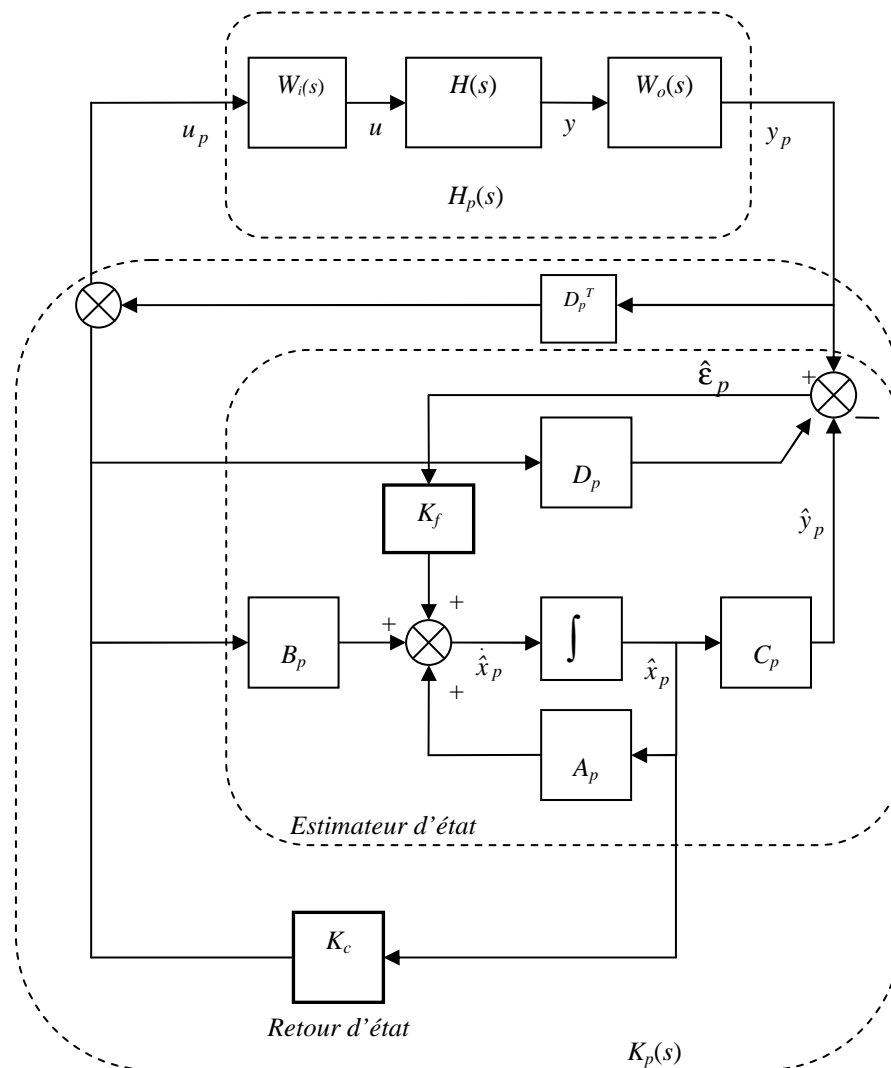


Figure 3-13: Implémentation retour d'état / observateur

L'intérêt de ce type de représentation est qu'il permet et de donner aux états du précorrecteur un sens physique de par l'observateur et ainsi de contrôler l'évolution de l'état du système. Il permet aussi de prendre en compte certaines non-linéarités (liées au conditionnement de la commande par exemple) directement dans l'observateur.

3.3. Une méthode générique pour la déclinaison des exigences

3.3.1. Principe général

Dans le milieu industriel, l'ingénieur a souvent affaire à une spécification qui prend la forme de m exigences sous la forme $Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m$. Cherchant à éviter l'étape de déclinaison fréquentielle car source d'approximations inévitables, on se

propose de chercher les pondérations stables solutions du problème d'optimisation suivant :

$$\begin{aligned} & \min_{W_o(s), W_i(s)} \gamma \\ & \text{sous contraintes que :} \\ & W_o(s), W_i(s) \in \mathbf{RH}_\infty \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned} \quad (3-41)$$

En supposant pouvoir évaluer, pour un correcteur donné, chacune de ces exigences (par exemple par simulation), ce problème sous contraintes peut être transformé en un problème non contraint en introduisant des fonctions de pénalités. Sa forme générale s'écrit :

$$\min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma + f_c(H, K), \quad f_c(H, K) = \sum_{j=1}^m \frac{e^{\alpha_j(Z_j - \bar{Z}_j')}}{\beta_j} \quad (3-42)$$

La fonction de contraintes $f_c(H, K)$ est évaluée lorsque le système H est bouclé avec le correcteur K .

Nous avons donc 3 paramètres $\alpha_j, \beta_j, \bar{Z}_j'$ à régler pour chaque contrainte. Pour satisfaire les exigences sur les Z_j , de toute évidence on doit choisir :

$$\begin{aligned} \bar{Z}_j' & \leq \bar{Z}_j \\ \alpha_j & \gg 1 \end{aligned} \quad (3-43)$$

Cependant cela n'a guère de sens de transformer un problème de réglage de pondérations en problème de réglage de paramètres de la fonction de coût : une méthode de réglage systématique est nécessaire.

Considérons un problème à une seule exigence :

$$\min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma + e^{\alpha(Z - \bar{Z}')} \quad (3-44)$$

Pour satisfaire cette exigence plus facilement, introduisons une marge de sécurité $\varepsilon > 0$ de sorte que :

$$\bar{Z} = \bar{Z}' + \varepsilon \quad (3-45)$$

A la fin du processus d'optimisation, γ vaut γ_{opt} , Z vaut au plus \bar{Z} et la fonction de pénalité doit être du même ordre de grandeur que γ_{opt} , c'est-à-dire :

$$e^{\alpha(Z - \bar{Z}')} \Big|_{Z = \bar{Z}} \approx \gamma_{opt} \Leftrightarrow e^{\alpha(\bar{Z} - \bar{Z}')} \approx \gamma_{opt} \Leftrightarrow e^{\alpha\varepsilon} \approx \gamma_{opt} \Leftrightarrow \alpha \approx \frac{\ln(\gamma_{opt})}{\varepsilon} \quad (3-46)$$

En général il suffit de choisir :

$$0,1\bar{Z} \leq \varepsilon \leq 0,3\bar{Z} \quad (3-47)$$

Par exemple pour une marge de 20 % :

$$\varepsilon = 0,2\bar{Z} \quad (3-48)$$

Par ailleurs dans le cadre de la synthèse par *loop-shaping*, un bon objectif de réglage se caractérise par (§3.2.2) :

$$\gamma_{obj} = 3 \quad (3-49)$$

Revenons au problème à m exigences (3-42). A la fin de l'optimisation, f_c doit être du même ordre de grandeur que γ_{obj} . En supposant que les m exigences aient la même importance, alors cela implique que :

$$\begin{aligned} \alpha_j &= \frac{\ln(\gamma_{obj})}{\varepsilon_j}, \quad \beta_j = m \\ \varepsilon_j &= 0,2\bar{Z}_j, \quad \gamma_{obj} = 3 \end{aligned} \quad (3-50)$$

Ainsi :

$$f_c(H, K) = \frac{1}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{\varepsilon_j}(Z_j - \bar{Z}_j)} \quad (3-51)$$

A noter que :

$$\begin{aligned} f_c(H, K) &= \frac{1}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{\varepsilon_j}(Z_j - \bar{Z}_j)} \\ &= \frac{1}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{\varepsilon_j}(Z_j - \bar{Z}_j + \varepsilon_j)} \\ &= \frac{1}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{\varepsilon_j}(Z_j - \bar{Z}_j)} e^{\ln(\gamma_{obj})} \\ &= \frac{\gamma_{obj}}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{\varepsilon_j}(Z_j - \bar{Z}_j)} \end{aligned} \quad (3-52)$$

Ceci définit donc une méthode de déclinaison systématique de spécifications haut-niveau sans pertes de généralité. Les coefficients de pénalité sont formellement établis par les relations précédentes. On peut finalement réécrire :

$$\begin{array}{c}
 \min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma \\
 \text{sous contraintes que} \\
 Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \\
 \Downarrow \\
 \min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma + f_c(H, K), f_c(H, K) = \frac{\gamma_{obj}}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})(Z_j - \bar{Z}_j)}{\varepsilon_j}}
 \end{array} \tag{3-53}$$

3.3.2. Gestion du niveau de priorité sur les exigences

Un des intérêts de la méthode de déclinaison précédente est la possibilité de gérer facilement des niveaux de priorité sur les différentes exigences. En effet, on peut écrire la fonction de pénalité sous la forme d'un développement limitée :

$$e^{\alpha(Z - \bar{Z}')} = \sum_{k=0}^{+\infty} \frac{(\alpha(Z - \bar{Z}'))^k}{k!} \tag{3-54}$$

On peut ainsi définir la priorité d'ordre n d'une contrainte en procédant au développement limité de la pénalité à l'ordre n . Toutefois :

- la pénalité résultante doit rester positive,
- la pénalité doit rester monotone, ce qui restreint l'emploi du développement limité aux ordres impairs.

On définit donc la pénalité de priorité n de la manière suivante :

$$\begin{array}{c}
 \text{priorité forte } (\infty) : e^{\alpha(Z - \bar{Z}')} \\
 \Downarrow \\
 \text{priorité d'ordre } (n) : \max \left(0, 1 + \sum_{k=1}^{2n+1} \frac{(\alpha(Z - \bar{Z}'))^k}{k!} \right)
 \end{array} \tag{3-55}$$

L'intérêt de la méthode est que l'on conserve la règle de réglage des α_j et β_j en considérant d'abord toutes les exigences comme étant de priorité forte.

Nous traçons sur la Figure 3-14 les premiers ordres de développement au regard de la fonction exponentielle correspondante. Seuls les ordres impairs doivent être employés (ici les ordres 1 et 3).

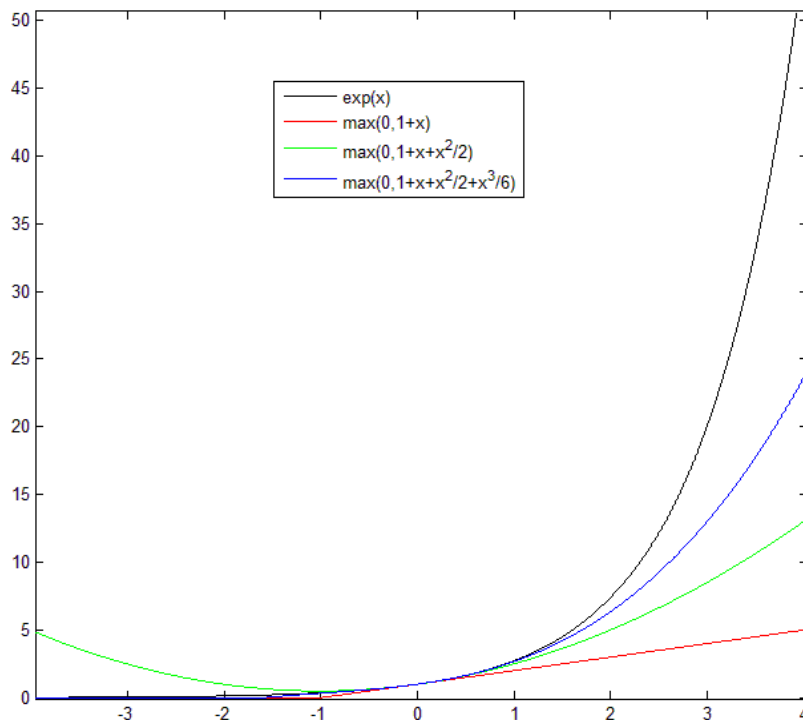


Figure 3-14: gestion de la priorité des exigences

3.4. Réglage optimal des pondérations

3.4.1. Optimisation sur modèle nominal

3.4.1.1. Définition de la fonction de cout

On reprend ici l'approche que nous avons proposée dans [FEY 13b] où on cherche à résoudre le problème suivant :

$$\begin{aligned}
 & \min_{W_o(s), W_i(s)} \gamma \\
 & \text{sous contraintes que :} \\
 & W_o(s), W_i(s) \in \mathbf{RH}_\infty \\
 & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
 \end{aligned} \tag{3-56}$$

Nous avons montré que ce problème pouvait être approché par le suivant :

$$\min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma + f_c(H, K), \quad f_c(H, K) = \frac{\gamma_{obj}}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{0.2Z_j}(Z_j - \bar{Z}_j)} \quad (3-57)$$

On rappelle que la synthèse par *loop-shaping* consiste à calculer sur la base d'un jeu de pondérations le correcteur de boucle par les équations suivantes :

$$\begin{aligned} H_p(s) &= W_o(s)H(s)W_i(s) \\ K_p(s) &= \arg \min \gamma, \quad \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_\infty < \gamma \\ K(s) &= W_i(s)K_p(s)W_o(s) \end{aligned} \quad (3-58)$$

Nous avons choisi de ne pas structurer les pondérations en termes de pôles/zéros/amortissement. Ainsi décidons-nous de rechercher les pondérations (centralisées ou pas) directement dans l'espace d'état :

$$W_i(s) = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad W_o(s) = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix} \quad (3-59)$$

En posant :

$$\bar{\lambda}_o = \max_j (\operatorname{Re}(\lambda_j(A_o))), \quad \bar{\lambda}_i = \max_j (\operatorname{Re}(\lambda_j(A_i))) \quad (3-60)$$

Et en notant pour simplifier :

$$\Theta_i = \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix}, \quad \Theta_o = \begin{pmatrix} A_o & B_o \\ C_o & D_o \end{pmatrix} \quad (3-61)$$

Alors on calcule le correcteur H_∞ -optimal, solution du problème (3-57), en considérant le problème d'optimisation suivant :

$$\begin{aligned}
& \min_{\Theta_i, \Theta_o} f(\Theta_i, \Theta_o) \\
& \text{si } \max(\bar{\lambda}_o, \bar{\lambda}_i) \geq 0 \\
& \quad f(\Theta_i, \Theta_o) = \max(\bar{\lambda}_o, \bar{\lambda}_i) \\
& \text{sinon} \\
& \quad H_p(s) = W_o(s)H(s)W_i(s) \\
& \quad K_p(s) = \arg \min \gamma, \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_{\infty} < \gamma \quad (3-62) \\
& \quad K(s) = W_i(s)K_p(s)W_o(s) \\
& \quad f(\Theta_i, \Theta_o) = -\frac{1}{\gamma + f_c(H, K)} \\
& \text{fin si}
\end{aligned}$$

Ce schéma d'optimisation est identique à celui de la synthèse H_{∞} structurée. Il se décompose en deux phases : une première phase de stabilisation des pondérations et une seconde phase liée à l'optimisation H_{∞} proprement dite. Là encore la fonction de coût est non différentiable mais quasi-continue.

Comme on le voit, en fonction de l'architecture de pondérations retenue (centralisée ou non), le seul degré d'intervention de l'ingénieur qui nécessite du retour d'expérience est réduit au choix de l'ordre des pondérations.

3.4.1.2. Mise en œuvre

Dans un souci d'optimiser le temps de convergence du processus d'optimisation, les pondérations sont cherchées sous forme tridiagonale et l'espace de recherche est transformé afin d'augmenter la sensibilité de l'algorithme (voir en première partie le paragraphe « Amélioration de la rapidité de convergence pour la synthèse des correcteurs »).

Comme pour la synthèse de correcteurs à ordre fixé, les pondérations peuvent être décentralisées ou non. Leur reconstruction suit exactement le même schéma. Dans les deux cas, le vecteur de décision rassemblera les coefficients des représentations d'état des deux pondérations cherchées :

$$X = (X_o \quad X_i)^T \quad (3-63)$$

Par exemple :

- si $W_o(s)$ est centralisée alors on posera :

$$\begin{aligned}
 X_o &= (x_{A_o}, x_{B_o}, x_{C_o}, x_{D_o}) \\
 x_{A_o} &= (a_{1,1}, \dots, a_{n,n}, a_{1,2}, \dots, a_{n-1,n}, a_{2,1}, \dots, a_{n,n-1}) \\
 x_{B_o} &= (b_{1,1}, \dots, b_{n,1}, b_{1,2}, \dots, b_{n,2}, \dots, b_{1,n_y}, \dots, b_{n,n_y}) \\
 x_{C_o} &= (c_{1,1}, \dots, c_{1,n}, c_{2,1}, \dots, c_{2,n}, \dots, c_{n_y,1}, \dots, c_{n_y,n}) \\
 x_{D_o} &= (d_{1,1}, \dots, d_{1,n_y}, d_{2,1}, \dots, d_{2,n_y}, \dots, d_{n_y,1}, \dots, d_{n_y,n_y})
 \end{aligned} \tag{3-64}$$

- si $W_o(s)$ est décentralisée¹⁷ alors on posera :

$$W_o(s) = \begin{bmatrix} W_{o1,1}(s) & \dots & W_{o1,n_y}(s) \\ \vdots & \ddots & \vdots \\ W_{on_y,1}(s) & \dots & W_{on_y,n_y}(s) \end{bmatrix} \tag{3-65}$$

L'ordre de chaque transfert est défini par :

$$\partial^\circ W_o = \begin{bmatrix} n_{o1,1} & \dots & n_{o1,n_y} \\ \vdots & \ddots & \vdots \\ n_{on_y,1} & \dots & n_{on_y,n_y} \end{bmatrix} \tag{3-66}$$

Le vecteur de décision s'écrira :

$$\begin{aligned}
 X &= (X_{o1,1}, \dots, X_{o1,n_y}, X_{o2,1}, \dots, X_{o2,n_y}, \dots, X_{on_y,1}, \dots, X_{on_y,n_y})^T \\
 X_{o_{i,l}} &= (x_D)_{i,j} \text{ si } n_{o_{i,j}} = 0 \\
 X_{o_{i,l}} &= (x_A, x_B, x_C, x_D)_{i,j} \text{ si } n_{o_{i,j}} \geq 1 \\
 \left. \begin{aligned}
 x_A &= (a_{1,1}, \dots, a_{n,n}, a_{1,2}, \dots, a_{n-1,n}, a_{2,1}, \dots, a_{n,n-1}) \\
 x_B &= (b_{1,1}, \dots, b_{n,1}) \\
 x_C &= (c_{1,1}, \dots, c_{1,n}) \\
 x_D &= d
 \end{aligned} \right\} n = n_{o_{i,j}}
 \end{aligned} \tag{3-67}$$

Nous cherchons à résoudre le problème d'optimisation suivant :

$$\begin{aligned}
 \min f(X) \\
 X \in [x; \bar{x}]
 \end{aligned} \tag{3-68}$$

Où la variable X est définie précédemment et où la fonction $f(X)$ consiste à réaliser les étapes suivantes :

$$- \text{ construire } W_o(s) = \left[\begin{array}{c|c} A_o & B_o \\ \hline C_o & D_o \end{array} \right] \text{ et } W_i(s) = \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array} \right] \text{ à partir de } X ;$$

¹⁷ Au sens du paragraphe « Correcteur centralisé et décentralisé » de la première partie.

– calculer l’abscisse spectrale suivante :

$$\bar{\lambda}_o = \max_j (\operatorname{Re}(\lambda_j(A_o))), \bar{\lambda}_i = \max_j (\operatorname{Re}(\lambda_j(A_i))), \bar{\lambda} = \max(\bar{\lambda}_o, \bar{\lambda}_i) \quad (3-69)$$

– si $\bar{\lambda} \geq 0$ alors $f(X) = \bar{\lambda}$,

– sinon :

- construire le loop-shape $H_p(s) = W_o(s)H(s)W_i(s)$;
- calculer le précorrecteur $K_p(s)$ qui minimise γ tel que :

$$\left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_{\infty} < \gamma \quad (3-70)$$

- calculer le correcteur final $K(s) = W_i(s)K_p(s)W_o(s)$;

- évaluer $f(X) = -\frac{1}{\gamma + f_c(H, K)}$.

3.4.2. Optimisation multi-modèles

Dans ce paragraphe, on suppose disposer de la boule de modèles suivante centrée sur un modèle nominal $H_0(s)$:

$$\wp = \{H_i(s), i = 0, \dots, N\} \quad (3-71)$$

3.4.2.1. Optimisation robuste en stabilité

On souhaite que le correcteur obtenu précédemment, performant sur le modèle nominal, stabilise en plus l’ensemble \wp . Cet objectif de stabilisation est équivalent au suivant :

$$\begin{aligned} &K(s) \text{ stabilise } \wp \\ &\Leftrightarrow \\ &H_{clj}(s) \in \mathbf{RH}_{\infty}, j = 1, \dots, N \\ &H_{clj}(s) = \begin{bmatrix} I \\ K(s) \end{bmatrix} (I + H_j(s)K(s))^{-1} \begin{bmatrix} I & H_j(s) \end{bmatrix} \end{aligned} \quad (3-72)$$

L’optimisation robuste en stabilité consiste à résoudre le problème suivant :

$$\begin{aligned}
 & \min_{W_o(s), W_i(s)} \gamma \\
 & \text{sous contraintes que :} \\
 & \quad W_o(s), W_i(s) \in \mathbf{RH}_\infty, \\
 & \quad H_{cl_j}(s) \in \mathbf{RH}_\infty, \quad j = 1, \dots, N,
 \end{aligned} \tag{3-73}$$

$$Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \text{ pour } H_0(s).$$

A noter que le correcteur stabilise déjà le modèle nominal.

Alors on calcule le correcteur H_∞ -optimal, solution du problème (3-73), en considérant le problème d'optimisation suivant :

$$\begin{aligned}
 & \min_{\Theta_i, \Theta_o} f(\Theta_i, \Theta_o) \\
 & \text{si } \max(\bar{\lambda}_o, \bar{\lambda}_i) \geq 0 \\
 & \quad f(\Theta_i, \Theta_o) = \max(\bar{\lambda}_o, \bar{\lambda}_i) \\
 & \text{sinon} \\
 & \quad H_p(s) = W_o(s)H_0(s)W_i(s) \\
 & \quad K_p(s) = \arg \min \gamma, \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_\infty < \gamma \\
 & \quad K(s) = W_i(s)K_p(s)W_o(s) \\
 & \quad H_{cl_j}(s) = \begin{bmatrix} I \\ K(s) \end{bmatrix} (I + H_j(s)K(s))^{-1} \begin{bmatrix} I & H_j(s) \end{bmatrix} \left. \vphantom{\begin{bmatrix} I \\ K(s) \end{bmatrix}} \right\} \quad j = 1, \dots, N \\
 & \quad \bar{\lambda}_{cl_j} = \max(\text{Re}(\lambda(A_{cl_j}))) \\
 & \text{si } \max_{j=1, \dots, N} \bar{\lambda}_{cl_j} \geq 0 \\
 & \quad f(\Theta_i, \Theta_o) \rightarrow \infty \\
 & \text{sinon} \\
 & \quad f(\Theta_i, \Theta_o) = -\frac{1}{\gamma + f_c(H_0, K)} \\
 & \text{fin si} \\
 & \text{fin si}
 \end{aligned} \tag{3-74}$$

3.4.2.2. Optimisation robuste en marge de stabilité

On souhaite que le correcteur obtenu précédemment, performant sur le modèle nominal et stabilisant φ , maximise en plus les marges de stabilité « pire cas » lorsqu'il boucle les $H_j(s)$, ce qui revient à minimiser le γ pire cas.

L'optimisation robuste en marges de stabilité consiste à résoudre le problème suivant :

$$\begin{aligned}
& \min_{W_o(s), W_i(s)} \gamma, \gamma = \max_{i=0, \dots, N} \gamma_i \\
& \text{sous contraintes que :} \\
& \quad W_o(s), W_i(s) \in \mathbf{RH}_\infty, \\
& \quad H_{clj}(s) \in \mathbf{RH}_\infty, j = 1, \dots, N, \\
& \quad Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \text{ pour } H_0(s).
\end{aligned} \tag{3-75}$$

On calcule alors le correcteur H_∞ -optimal, solution du problème (3-75), en considérant le problème d'optimisation suivant :

$$\begin{aligned}
& \min_{\Theta_i, \Theta_o} f(\Theta_i, \Theta_o) \\
& \text{si } \max(\bar{\lambda}_o, \bar{\lambda}_i) \geq 0 \\
& \quad f(\Theta_i, \Theta_o) = \max(\bar{\lambda}_o, \bar{\lambda}_i) \\
& \text{sinon} \\
& \quad H_p(s) = W_o(s)H_0(s)W_i(s) \\
& \quad K_p(s) = \arg \min \gamma_0, \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_\infty < \gamma_0 \\
& \quad K(s) = W_i(s)K_p(s)W_o(s) \\
& \quad H_{pj}(s) = W_o(s)H_j(s)W_i(s) \\
& \quad H_{clj}(s) = \left. \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_{pj}(s)K_p(s))^{-1} \begin{bmatrix} I & H_{pj}(s) \end{bmatrix} \right\} j = 1, \dots, N \\
& \quad \bar{\lambda}_{clj} = \max(\text{Re}(\lambda(A_{clj}))) \\
& \text{si } \max_{j=1, \dots, N} \bar{\lambda}_{clj} \geq 0 \\
& \quad f(\Theta_i, \Theta_o) \rightarrow \infty \\
& \text{sinon} \\
& \quad \gamma_j = \|H_{clj}(s)\|_\infty, j = 1, \dots, N \\
& \quad f(\Theta_i, \Theta_o) = - \frac{1}{\max_{j=0, \dots, N} \gamma_j + f_c(H_0, K)} \\
& \text{fin si} \\
& \text{fin si}
\end{aligned} \tag{3-76}$$

3.4.2.3. Optimisation robuste en performance

On souhaite que le correcteur obtenu précédemment, performant sur le modèle nominal, stabilisant \wp , maximisant les marges de stabilité « pire cas » sur \wp , soit en plus performant sur tout \wp . C'est donc un problème de robustesse en performance.

Un objectif ambitieux serait ainsi de chercher le correcteur solution du problème suivant :

$$\begin{aligned}
 & \min_{W_o(s), W_i(s)} \gamma, \gamma = \max_{i=0, \dots, N} \gamma_i \\
 & \text{sous contraintes que} \\
 & W_o(s), W_i(s) \in \mathbf{RH}_\infty, \\
 & \left. \begin{aligned} & H_{cl}(s) \in \mathbf{RH}_\infty, \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned} \right\} \forall H_j(s) \in \mathcal{S}
 \end{aligned} \tag{3-77}$$

Il existe plusieurs formulations plus ou moins conservatives pour approcher ce problème.

La plus directe consiste à déterminer le correcteur H_∞ -optimal, solution du problème (3-77), en considérant le problème d'optimisation suivant :

$$\begin{aligned}
 & \min_{\Theta_i, \Theta_o} f(\Theta_i, \Theta_o) \\
 & \text{si } \max(\bar{\lambda}_o, \bar{\lambda}_i) \geq 0 \\
 & f(\Theta_i, \Theta_o) = \max(\bar{\lambda}_o, \bar{\lambda}_i) \\
 & \text{sinon} \\
 & H_p(s) = W_o(s)H_0(s)W_i(s) \\
 & K_p(s) = \arg \min \gamma_0, \left\| \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_p(s)K_p(s))^{-1} \begin{bmatrix} I & H_p(s) \end{bmatrix} \right\|_\infty < \gamma_0 \\
 & K(s) = W_i(s)K_p(s)W_o(s) \\
 & \left. \begin{aligned} & H_{p_j}(s) = W_o(s)H_j(s)W_i(s) \\ & H_{clj}(s) = \begin{bmatrix} I \\ K_p(s) \end{bmatrix} (I + H_{p_j}(s)K_p(s))^{-1} \begin{bmatrix} I & H_{p_j}(s) \end{bmatrix} \\ & \bar{\lambda}_{clj} = \max(\operatorname{Re}(\lambda(A_{clj}))) \end{aligned} \right\} j = 1, \dots, N \tag{3-78} \\
 & \text{si } \max_{j=1, \dots, N} \bar{\lambda}_{clj} \geq 0 \\
 & f(\Theta_i, \Theta_o) \rightarrow \infty \\
 & \text{sinon} \\
 & \gamma_j = \left\| H_{clj}(s) \right\|_\infty, j = 1, \dots, N \\
 & f(\Theta_i, \Theta_o) = - \frac{1}{\max_{j=0, \dots, N} (\gamma_j + f_c(H_j, K))} \\
 & \text{fin si} \\
 & \text{fin si}
 \end{aligned}$$

3.4.3. Exemple applicatif – la stabilisation inertielle de ligne de visée

3.4.3.1. Le viseur d'observation

3.4.3.1.1. Description du système et des exigences

Afin d'illustrer la méthode de synthèse, nous proposons d'appliquer les développements théoriques de cette partie sur un problème de stabilisation inertielle de ligne de visée.

Le but d'une plate-forme gyrostabilisée deux axes est de maintenir inertielle (c'est-à-dire indépendante des mouvements d'un porteur), l'orientation d'un dispositif d'observation en site et en gisement. Cette inertialité est rendue possible par l'emploi de gyromètres (respectivement gyroscopes) pour mesurer la vitesse absolue (respectivement l'orientation absolue) des cardans, et de moteurs électriques en guise d'actionneur (Figure 3-15).

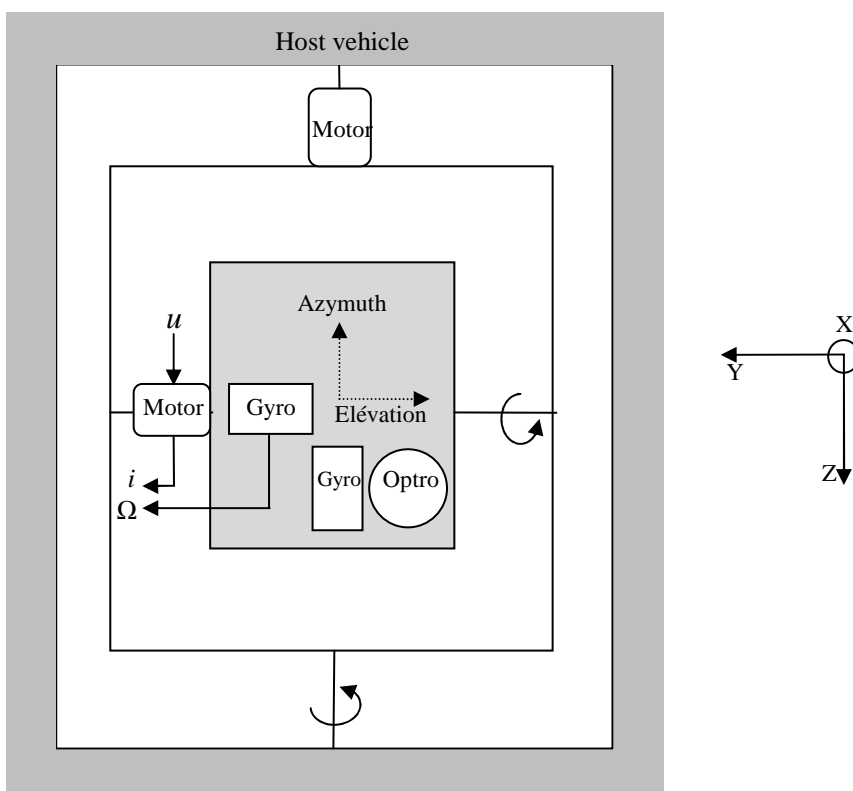


Figure 3-15: Synoptique simplifiée d'une plate-forme gyrostabilisée

Pour plus de détails sur les viseurs gyrostabilisés, on se reportera à [MAS 08] et [HIL 08] qui décrivent de manière exhaustive les différentes architectures possibles. Chaque axe peut être considéré comme un système SIMO $H_0(s)$ d'entrée u (la commande moteur) et de sorties i et Ω_a , respectivement le courant moteur et la vitesse absolue de la charge mesurée par le gyromètre.

$$Y(s) = \begin{pmatrix} \Omega_a(s) \\ I(s) \end{pmatrix} = H_0(s)U(s) \quad (3-79)$$

$$H_0(s) = \begin{pmatrix} H_m(s) \\ 1 \end{pmatrix} H_{el}(s) = \begin{pmatrix} H_{u \rightarrow \Omega}(s) \\ H_{u \rightarrow i}(s) \end{pmatrix}$$

Un exemple de représentation fréquentielle pour $H_0(s)$ est donné sur la Figure 3-16. Le système considéré est d'ordre 17.

Pour des raisons de confidentialité, les fréquences et les bases temporelles des signaux mis en jeu dans cette étude ont été normalisées.

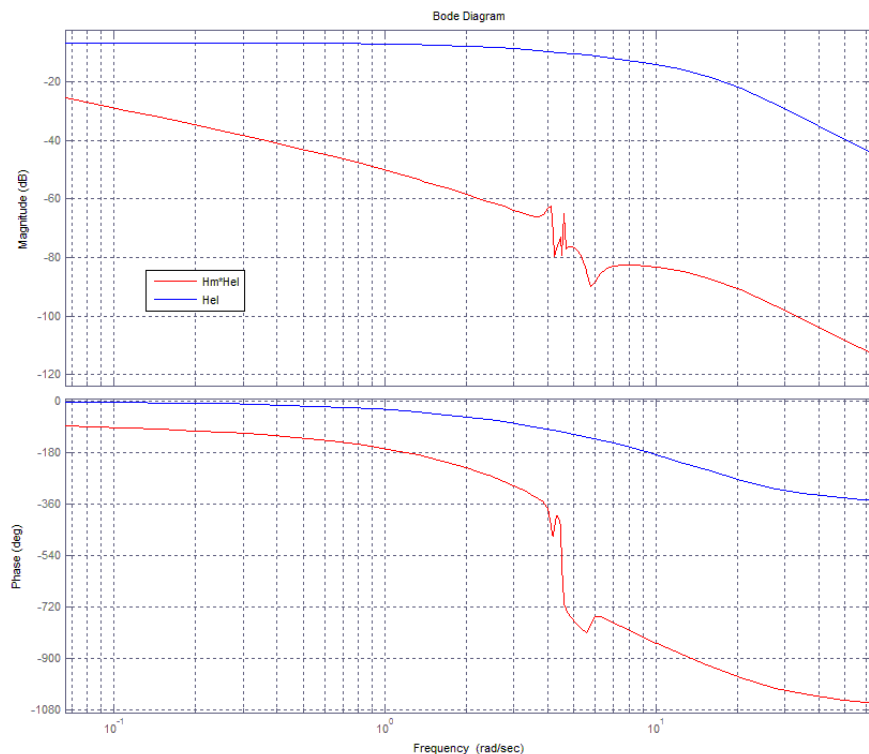


Figure 3-16: Représentation fréquentielle du système SIMO

En général, la plate-forme électromécanique porte un dispositif optronique conférant au viseur certaines fonctionnalités fondamentales : observations, détections, identification, ... Pour satisfaire à ces fonctionnalités, le viseur doit disposer de la portée suffisante. Cette portée est directement reliée à la performance de stabilisation du bouclage gyrométrique vis-à-vis d'environnements mécaniques et de conditions d'emploi plus ou moins sévères et doit être compatible avec le dimensionnement des capteurs optroniques ; ces derniers sont caractérisés par un temps d'intégration (T_i) (temps pendant lequel l'image est acquise et générée électroniquement) et par la taille de leurs détecteurs pixéliques (IFOV). On comprend donc que pour une observation de qualité, il faut limiter les écarts angulaires absolus de ligne de visée à une valeur compatible de l'IFOV et ce, sur le temps d'intégration T_i , en réponse à toutes les perturbations.

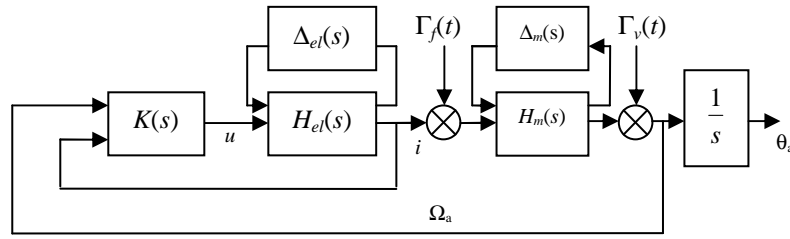


Figure 3-17: Schéma de commande du viseur

Comme le montre le schéma de commande de la Figure 3-17, le problème de stabilisation inertielle consiste à rejeter principalement deux types de perturbations externes à l'aide d'un bouclage gyrométrique :

- la première est le couple de frottements engendré lors d'un mouvement relatif entre le porteur du viseur et les cardans. La plupart du temps, ce couple est modélisé par un échelon de perturbation Γ_{f0} en entrée de modèle dont le signe dépend du signe de la vitesse relative (modèle de Coulomb) :

$$\Gamma_f(t) = \Gamma_{f0} \text{signe}(\Omega_{relative}(t)), \Gamma_{f0} < 0 \quad (3-80)$$

Cette perturbation apparaît donc lorsque l'attitude du porteur et/ou la consigne de précession du viseur évoluent. Afin d'éviter le bougé d'image lors de l'observation (et ainsi assurer un confort visuel), il convient de rendre compatible le déplacement angulaire dynamique de la ligne de visée en réponse aux frottements (lorsque ce dernier passe de $-\Gamma_{f0}$ à $+\Gamma_{f0}$, soit un échelon relatif de $2\Gamma_{f0}$) compatible avec l'IFOV. De même, l'erreur statique occasionnée doit être nulle. Plus particulièrement la dynamique temporelle du rejet du frottement doit être compatible avec la dynamique maximale du régime oscillant du porteur qui n'est que de quelques Hz, une dizaine tout au plus. L'erreur statique en réponse aux frottements doit donc être nulle (ou tout du moins très faible) au bout d'un temps T_f :

$$\left. \begin{array}{l} |\theta_a(t)| \leq \bar{\theta}_f \\ \max_{t > T_f} |\theta_a(t)| < \varepsilon_f \end{array} \right\} \text{en réponse à l'échelon } \Gamma_f(t) = 2\Gamma_{f0}, t > 0 \quad (3-81)$$

- la seconde perturbation provient des déformations mécaniques de la structure du viseur qui de par sa flexibilité transmet à la ligne de visée, voire amplifiée, une partie de l'environnement mécanique auquel le viseur est soumis, et ce suivant 3 directions orthogonales : la direction longitudinale (Axe X), la direction transverse (Axe Y) et la direction verticale (Axe Z). Pour chaque direction d'excitation, il est possible de disposer d'une acquisition temporelle de la perturbation induite sur la ligne de visée (non bouclée) (Figure 3-18) nommé ici respectivement Γ_{vx} , Γ_{vy} et Γ_{vz} .

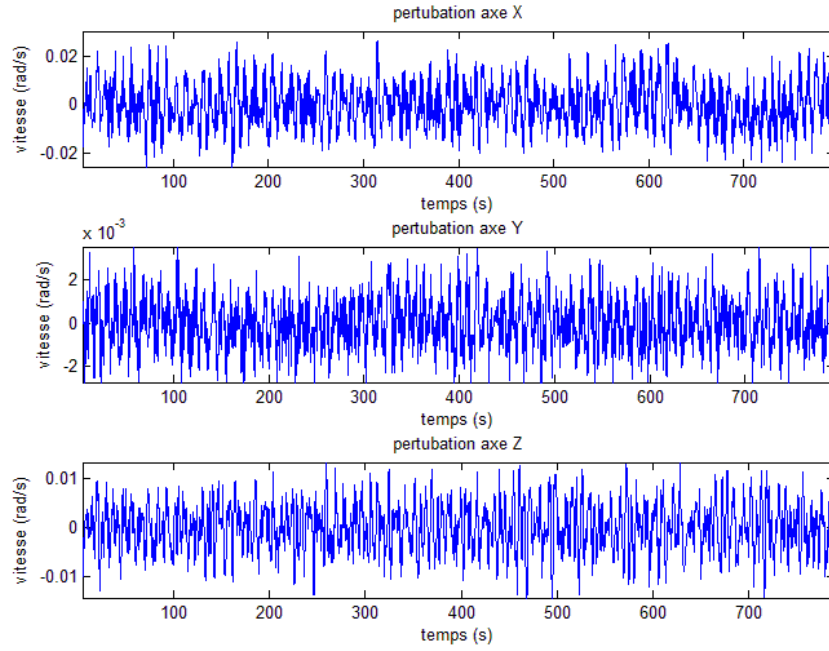


Figure 3-18: Acquisition temporelle de Γ_{vx} , Γ_{vy} et Γ_{vz}

Afin d'assurer au viseur ses fonctionnalités d'observation, il convient de limiter le flou généré par cette perturbation « haute fréquence » à une valeur compatible avec l'IFOV pendant le temps d'intégration T_i . Ainsi la spécification relative à cette perturbation pourra être de la forme :

$$\begin{aligned}
 \sigma_{\theta_{vx}}(t) &= \sqrt{\int_t^{t+T_i} \theta_a(\tau)^2 d\tau} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vx}(t) \\
 \sigma_{\theta_{vy}}(t) &= \sqrt{\int_t^{t+T_i} \theta_a(\tau)^2 d\tau} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vy}(t) \\
 \sigma_{\theta_{vz}}(t) &= \sqrt{\int_t^{t+T_i} \theta_a(\tau)^2 d\tau} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vz}(t)
 \end{aligned} \tag{3-82}$$

A noter que souvent, on considère un majorant en prenant $T_i \rightarrow \infty$.

Enfin un besoin important concerne la consommation du bouclage. En effet, les moteurs ont une dynamique maximale en entrée à respecter sous peine de les endommager. Ainsi faut-il assurer le rejet de perturbations tout en limitant la valeur maximale du courant moteur, c'est à-dire ajouter l'exigence (3-83).

$$|i(t)| < i_{\max} \text{ en réponse aux perturbations} \tag{3-83}$$

La spécification rassemble donc un ensemble d'exigences système :

$$\left. \begin{array}{l}
|\theta_a(t)| \leq \bar{\theta}_f, \forall t > 0 \text{ en réponse à l'échelon } \Gamma_f(t) \\
\max_{t > T_f} |\theta_a(t)| \leq \varepsilon_f \text{ en réponse à l'échelon } \Gamma_f(t) \\
|i(t)| \leq i_{\max}, \forall t > 0 \text{ en réponse à l'échelon } \Gamma_f(t) \\
\sigma_{\theta_{vx}} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vx}(t) \\
|i(t)| \leq i_{\max}, \forall t > 0 \text{ en réponse à l'excitation } \Gamma_{vx}(t) \\
\sigma_{\theta_{vy}} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vy}(t) \\
|i(t)| \leq i_{\max}, \forall t > 0 \text{ en réponse à l'excitation } \Gamma_{vy}(t) \\
\sigma_{\theta_{vz}} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vz}(t) \\
|i(t)| \leq i_{\max}, \forall t > 0 \text{ en réponse à l'excitation } \Gamma_{vz}(t)
\end{array} \right\} := \chi \quad (3-84)$$

En posant :

- $\theta_f(t)$ la réponse en position à l'excitation $\Gamma_f(t)$,
- $i_f(t)$ la réponse en courant à l'excitation $\Gamma_f(t)$,
- $\theta_{vx}(t)$ la réponse en position à l'excitation $\Gamma_{vx}(t)$,
- $\theta_{vy}(t)$ la réponse en position à l'excitation $\Gamma_{vy}(t)$,
- $\theta_{vz}(t)$ la réponse en position à l'excitation $\Gamma_{vz}(t)$,
- $i_{vx}(t)$ la réponse en courant à l'excitation $\Gamma_{vx}(t)$,
- $i_{vy}(t)$ la réponse en courant à l'excitation $\Gamma_{vy}(t)$,
- $i_{vz}(t)$ la réponse en courant à l'excitation $\Gamma_{vz}(t)$,

Alors la spécification se reformule aisément comme suit :

$$\left. \begin{array}{l}
\max_{t > 0} (|\theta_f(t)|) \leq \bar{\theta}_f, \max_{t > T_f} (|\theta_f(t)|) \leq \varepsilon_f, \max_{t > 0} (|i_f(t)|) \leq i_{\max} \\
\sigma_{\theta_{vx}} \leq \bar{\theta}_v, \max_{t > 0} (|i_{vx}(t)|) \leq i_{\max} \\
\sigma_{\theta_{vy}} \leq \bar{\theta}_v, \max_{t > 0} (|i_{vy}(t)|) \leq i_{\max} \\
\sigma_{\theta_{vz}} \leq \bar{\theta}_v, \max_{t > 0} (|i_{vz}(t)|) \leq i_{\max}
\end{array} \right\} := \chi \quad (3-85)$$

Pour des raisons de confidentialité, les spécifications numériques sont occultées. On mesurera la tenue des exigences en relatif par rapport au spécifié.

3.4.3.1.2. Optimisation des pondérations sur modèle nominal

On se propose de chercher par *loop-shaping* le correcteur H_∞ solution du problème d'optimisation suivant (Figure 3-19) :

$$\min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma \quad (3-86)$$

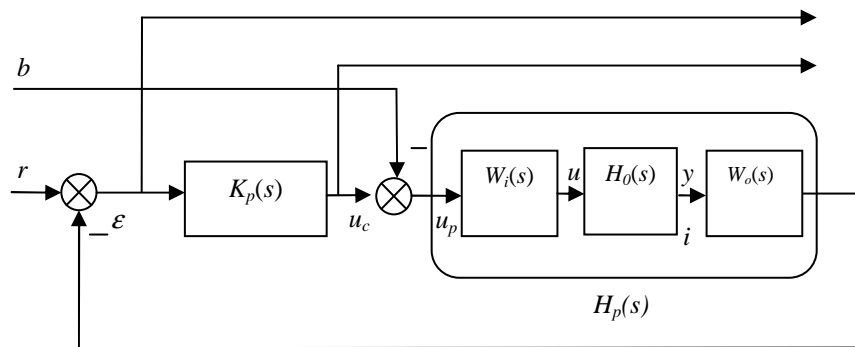
sous contrainte de χ

Comme on l'a vu, un des intérêts de l'approche *loop-shaping* est le lien direct entre γ et les marges de stabilité du bouclage. Ainsi par le problème (3-86) on est assuré de disposer de marges de stabilité convenables lorsque les contraintes sont satisfaites.

C'est donc une application directe du problème (3-62) avec ici :

$$f_c(H_0, K) = \frac{\gamma_{obj}}{9} \left[\begin{array}{l} e^{\frac{\ln(\gamma_{obj})}{0.2\theta_f} \left(\max_{t>0}(|\theta_f(t)|) - \bar{\theta}_f \right)} + e^{\frac{\ln(\gamma_{obj})}{0.2\varepsilon_f} \left(\max_{t>T_f}(|\theta_f(t)|) - \varepsilon_f \right)} \\ + e^{\frac{\ln(\gamma_{obj})}{0.2i_{max}} \left(\max_{t>0}(|i_f(t)|) - i_{max} \right)} \\ + e^{\frac{\ln(\gamma_{obj})}{0.2\theta_v} (\sigma_{\theta_{vx}} - \bar{\theta}_v)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{max}} \left(\max_{t>0}(|i_{vx}(t)|) - i_{max} \right)} \\ + e^{\frac{\ln(\gamma_{obj})}{0.2\theta_v} (\sigma_{\theta_{vy}} - \bar{\theta}_v)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{max}} \left(\max_{t>0}(|i_{vy}(t)|) - i_{max} \right)} \\ + e^{\frac{\ln(\gamma_{obj})}{0.2\theta_v} (\sigma_{\theta_{vz}} - \bar{\theta}_v)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{max}} \left(\max_{t>0}(|i_{vz}(t)|) - i_{max} \right)} \end{array} \right] \quad (3-87)$$

$$\gamma_{obj} = 3$$



$$K_p(s) = \arg \min \gamma, \left\| \left[\begin{array}{c} I \\ K_p(s) \end{array} \right] (I + H_p(s)K_p(s))^{-1} \left[\begin{array}{cc} I & H_p(s) \end{array} \right] \right\|_\infty < \gamma$$

Figure 3-19: Synthèse par *loop-shaping*

Nous choisissons en sortie de modèle une pondération *MIMO* décentralisée :

$$W_o(s) = \begin{bmatrix} W_\Omega(s) & 0 \\ 0 & W_I(s) \end{bmatrix} \quad (3-88)$$

$$W_i(s) = W_u(s)$$

Nous choisissons les ordres suivants :

$$\partial^\circ W_\Omega = 6, \quad \partial^\circ W_u = 6, \quad \partial^\circ W_I = 1 \quad (3-89)$$

Ce qui conduit à 62 paramètres à optimiser.

Nous procédons à la résolution du problème avec plusieurs algorithmes et nous procédons à l'analyse statistique.

La fonction de coût f est définie par (3-87).

Les algorithmes d'optimisation testés étant stochastiques, une évaluation statistique des résultats est nécessaire pour chaque test.

Ainsi la logique de mesure de performance est la suivante.

Chaque test fait l'objet de 10 *runs* successifs : nous relevons alors la valeur minimale de f obtenue (f_{min}), la valeur moyenne de f (f_{mean}), son écart-type (f_{std}) et finalement le temps de calcul (t_{cpu}) mis pour la réalisation de ces 10 *runs*.

$$\left. \begin{aligned} f_{min} &= \min_{i=1, \dots, n \text{ runs}} f_i \\ f_{mean} &= \frac{1}{n} \sum_{i=1}^{n \text{ runs}} f_i \\ f_{std} &= \sqrt{\frac{1}{n} \sum_{i=1}^{n \text{ runs}} (f_i - f_{mean})^2} \end{aligned} \right\} \quad (3-90)$$

Ce qui nous permet d'établir un rang « moyen » pour chaque algorithme à partir des rangs obtenus pour les grandeurs (2-89) comme nous l'avons fait en première partie sur la synthèse H_∞ structurée. Les résultats sont présentés sur le Tableau 3.1.

Nous considérons que l'algorithme disposant du rang moyen le plus bas est finalement le meilleur pour notre problème de réglage des pondérations.

La mesure du temps de calcul est ici intéressante car :

- Certains algorithmes disposent d'une phase de recherche locale associée à une phase de recherche globale (ABC, CS) ; ainsi lors d'une itération, on peut évaluer la fonction de coût un nombre de fois plus important que pour des approches plus directes (PSO, QPSO, DE).
- La fonction de coût (3-62) se déroule en deux étapes : la stabilisation et la performance. Or ces deux étapes ne disposent pas du même temps d'exécution (la seconde prend plus de temps).

Ainsi à nombre d'itérations égales, les temps de calcul employés par chaque algorithme pour évaluer chaque cas test peuvent être très disparates.

La totalité du test a été réalisée sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

Chaque *run* fait l'objet de 10000 itérations.

<i>IT / run = 10000</i>	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>StaticCEC_QPSO</i>	-0,27	-0,21	0,030	130,27
<i>TVDecCEC_QPSO</i>	-0,23	-0,18	0,038	89,70
<i>Rnd_DE</i>	-0,24	-0,15	0,081	62,76
<i>Rnd2Bst_DE</i>	-0,29	-0,26	0,020	91,63
<i>Std_PSO</i>	-0,22	-0,14	0,077	93,84
<i>TVRndIW_PSO</i>	-0,22	-0,15	0,065	70,46
<i>ABC</i>	-0,16	-0,14	0,029	132,33
<i>CS</i>	-0,14	-0,06	0,046	21,21
<i>CS/DE</i>	-0,27	-0,23	0,049	290,63

Tableau 3.1 Comparatif d'algorithmes pour l'optimisation des pondérations

On constate que le calcul évolutionnaire et particulièrement *Rnd2Bst_DE* est le meilleur des algorithmes du point de vue des résultats pour le problème du réglage des pondérations. *CS* conduit à un temps de calcul bien plus faible mais à des résultats nettement moins bons.

Le meilleur des *runs* conduit au résultat suivant de la Figure 3-20.

Sur la Figure 3-20, on a de gauche à droite : le diagramme de Black-Nichols de la boucle ouverte (en entrée), diverses réponses fréquentielles ainsi que les réponses temporelles des signaux optimisés : résiduel de stabilisation en vibration, courant moteur en vibration, résiduel de stabilisation en réponse au frottement, courant correspondant.

Les résultats quant à l'évaluation des exigences temporelles peuvent être rassemblés dans le Tableau 3.2. On rappelle que l'on présente pour des raisons de confidentialité les résultats relatifs aux valeurs maximales temporelles spécifiées (confidentielles), c'est-à-dire par rapport à 1.

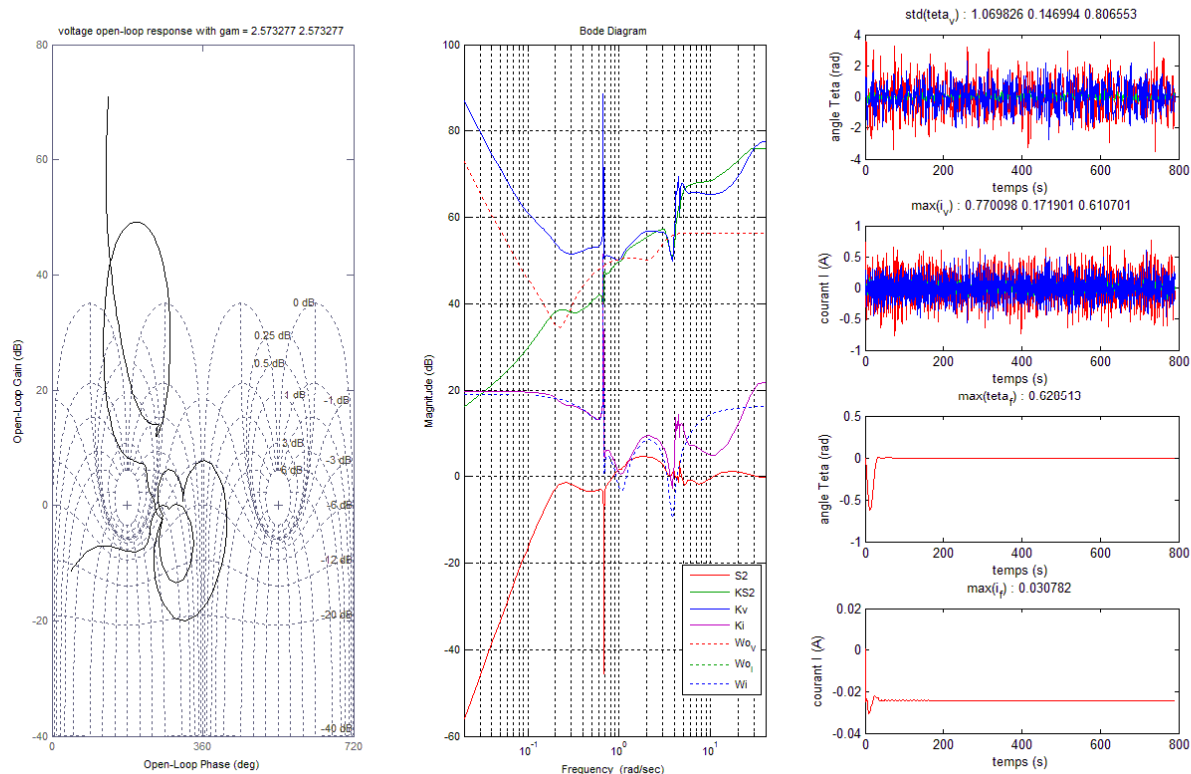


Figure 3-20: Régler optimal des pondérations

Exigences	Evaluations finales
γ	2,57
$\max_{t>0}(\theta_f(t)) \leq \bar{\theta}_f$	0,62
$\max_{t>T_f}(\theta_f(t)) \leq \varepsilon_f$	0
$\max_{t>0}(i_f(t)) \leq i_{\max}$	0,03
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,07
$\max_{t>0}(i_{vx}(t)) \leq i_{\max}$	0,77
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,15
$\max_{t>0}(i_{vy}(t)) \leq i_{\max}$	0,17
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,80
$\max_{t>0}(i_{vz}(t)) \leq i_{\max}$	0,61

Tableau 3.2 Evaluation des exigences à l'issue de l'optimisation des pondérations

On constate que toutes les exigences sont satisfaites, sauf la quatrième qui n'excède que de 7% la spécification.

3.4.3.1.3. Optimisation multi-modèles

A l'ensemble d'exigences χ , on peut ajouter un besoin de robustesse en stabilité voire en performances sur un ensemble de N modèles centré sur le modèle nominal $H_0(s)$:

$$\wp = \{H_i(s), i = 0, \dots, N\} \quad (3-91)$$

Nous supposons ici que les N modèles ne sont pas forcément mesurés mais peuvent être obtenus à partir de la réponse nominale de la Figure 3-16.

- Concernant le bloc « mécanique » $H_m(s)$:

Plusieurs méthodes existent pour disperser ce système. Nous proposons ici de faire varier aléatoirement sur des intervalles bornés la fréquence propre et les amortissements des pôles du système autour de leur valeur nominale. En effet, c'est simple et physique et cela s'avère adapté aux systèmes de la forme (3-92).

$$\frac{\Omega_a(s)}{C_m(s)} = \frac{1}{Js} + \sum_{k=1}^n \frac{\lambda_k s}{s^2 + 2\xi_k \omega_{0k} s + \omega_{0k}^2} \quad (3-92)$$

Un moyen simple pour réaliser cela est d'exprimer la représentation d'état de $H(s)$ dans sa base modale. Dans cette base la matrice d'état du système $H(s)$ est de la forme :

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 & & \\ 0 & \sigma_2 & \omega_2 & & \\ 0 & -\omega_2 & \sigma_2 & & \\ & & & \ddots & \end{bmatrix} \quad (3-93)$$

$$\sigma_i + j\omega_i = -\xi_i \omega_{0i} \pm j\omega_{0i} \sqrt{1 - \xi_i^2}$$

Ainsi en supposant que les fréquences propres et les amortissements varient respectivement de $\pm \delta_\omega$ % et $\pm \delta_\xi$ %, alors :

$$\begin{aligned} \sigma_i + j\omega_i &= -\xi_i (1 + \delta_\xi) \omega_{0i} (1 + \delta_\omega) \pm j\omega_{0i} (1 + \delta_\omega) \sqrt{1 - \xi_i^2 (1 + \delta_\xi)^2} \\ &= -\xi_i \omega_{0i} (1 + \delta_\xi + \delta_\omega + \delta_\xi \delta_\omega) \pm j\omega_{0i} (1 + \delta_\omega) \sqrt{1 - \xi_i^2 (1 + \delta_\xi)^2} \\ &\approx -\xi_i \omega_{0i} (1 + \delta_\xi + \delta_\omega) \pm j\omega_{0i} (1 + \delta_\omega) \sqrt{1 - \xi_i^2} \\ &= \sigma_i (1 + \delta_\xi + \delta_\omega) \pm j\omega_i (1 + \delta_\omega) \end{aligned} \quad (3-94)$$

Ce qui revient donc à faire varier la partie réelle du pôle correspondant de $\pm (\delta_\xi + \delta_\omega)$ % et sa partie imaginaire de $\pm \delta_\omega$ % .

- Concernant le bloc « électrique » $H_{el}(s)$:

Le contributeur majoritaire est la résistance électrique du moteur. Sa variation de $\pm \delta_R \%$ est induite par l'environnement thermique (environ $\pm 25\%$ quand la température varie de -40°C à $+70^\circ\text{C}$). En considérant que le gain statique de ce bloc varie en R^{-1} , au premier ordre on peut considérer que cela induit une variation du gain de $H_{el}(s)$ de $\pm \delta_R \%$. On peut donc générer les modèles en faisant varier aléatoirement le gain statique de $\pm \delta_R \%$ autour de sa valeur nominale.

Par exemple en supposant ici que $\delta_\omega = \pm 15\%$, $\delta_\xi = \pm 15\%$ et $\delta_R = \pm 25\%$, on peut définir la boule de 25 modèles dont les réponses fréquentielles sont données sur les Figure 3-21 et Figure 3-22.

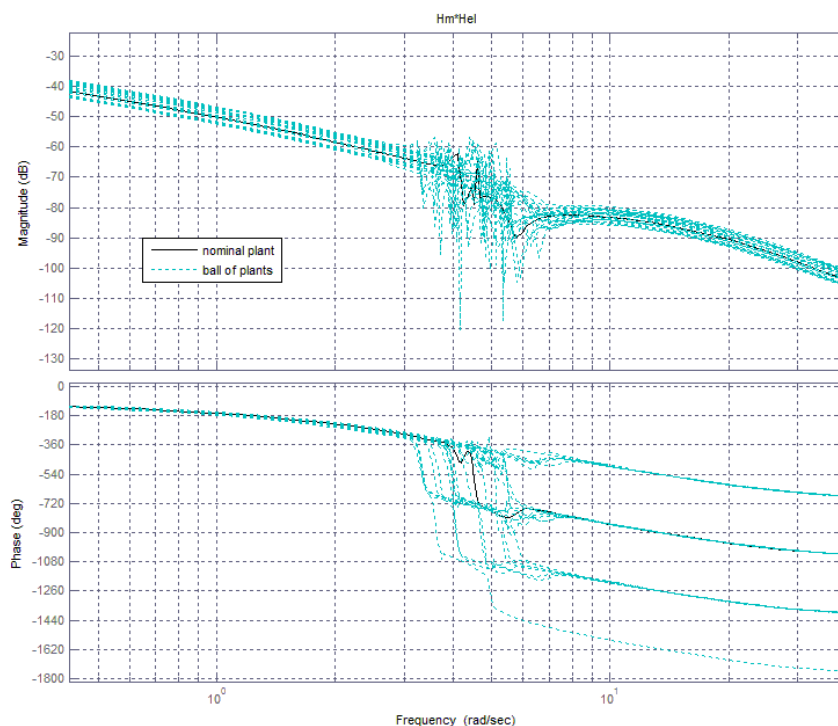


Figure 3-21: Bloc mécanique dispersé

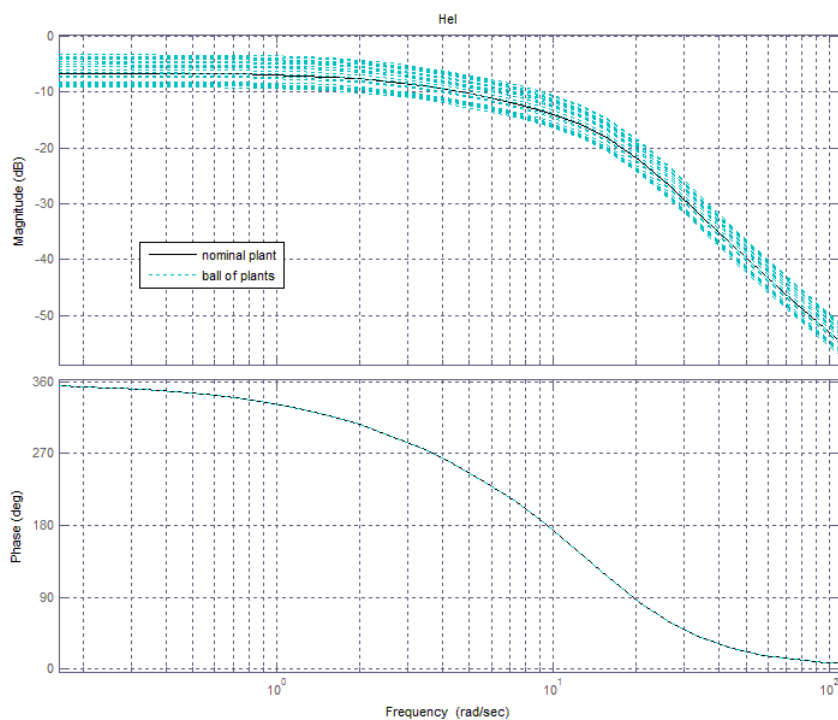


Figure 3-22: Bloc électrique dispersé

On constate sur la Figure 3-23 qu'en l'état, le correcteur obtenu précédemment est peu robuste vis-à-vis de la boule de modèles. De fait, la valeur pire-cas de γ obtenu pour l'ensemble de la boule de modèle vaut 23,03. Aussi allons-nous chercher dans la suite à robustifier le correcteur par les techniques multi-modèles développées précédemment.

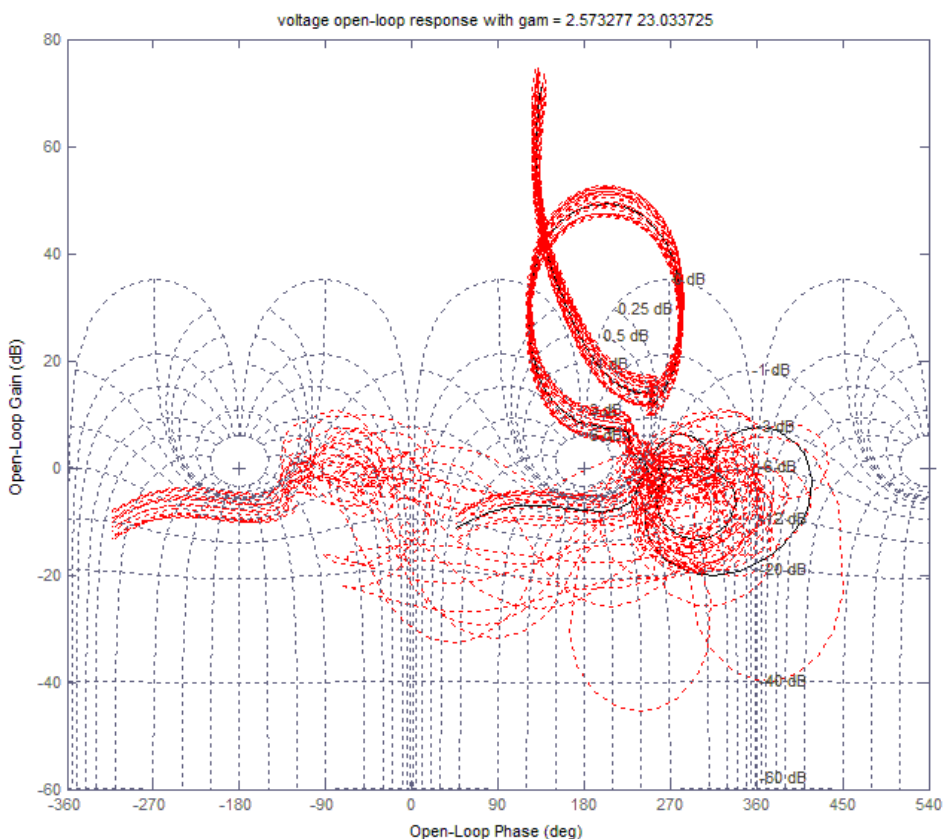


Figure 3-23: Réponse fréquentielle de la boucle ouverte pour la boucle de modèles

On peut chercher (avec *Rnd_2_Bst*) un correcteur solution des problèmes suivants :

- problème de robustesse de la stabilité :

$$\begin{aligned}
 & \min_{W_o(s), W_f(s) \in \mathbf{RH}_\infty} \gamma \\
 & \text{sous contrainte de } \chi \\
 & \text{et } K(s) \text{ stabilise tout } \varphi
 \end{aligned}
 \tag{3-95}$$

On obtient le résultat du Tableau 3.3.

<i>IT / run = 10000</i>	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>Rnd2Bst_DE</i>	-0,26	-0,20	-0,036	90,27

Tableau 3.3 Optimisation des pondérations – robustesse de la stabilité

On constate que les temps de calcul entre l’approche sur modèle nominal et le problème de robustesse en stabilité sont comparables.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-24.

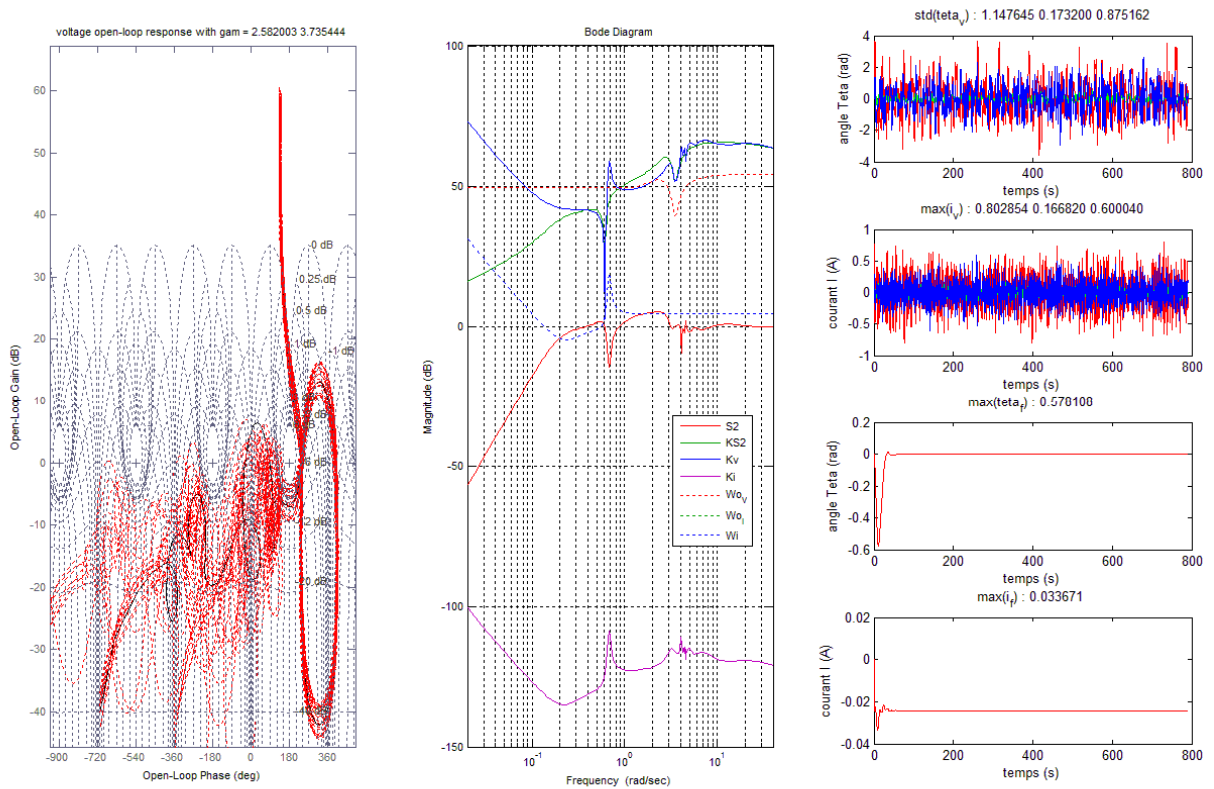


Figure 3-24: Problème de la robustesse de la stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.4.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	2,58 ; 3,73
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,58
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left i_f(t) \right \right) \leq i_{max}$	0,03
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,14
$\max_{t>0} \left(\left i_{vx}(t) \right \right) \leq i_{max}$	0,80
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,17
$\max_{t>0} \left(\left i_{vy}(t) \right \right) \leq i_{max}$	0,17
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,87
$\max_{t>0} \left(\left i_{vz}(t) \right \right) \leq i_{max}$	0,61

Tableau 3.4 Evaluation des exigences – robustesse de la stabilité

On constate là aussi que toutes les exigences sauf la quatrième sont satisfaites (un peu dégradées toutefois, l'augmentation de la robustesse s'accompagne d'une diminution de performances qui restent très acceptables). La valeur de γ pire cas est tombée à 3,73.

- problème de robustesse des marges de stabilité :

$$\begin{aligned} \min_{W_o(s), W_i(s) \in RH_\infty} \gamma, \quad \gamma = \max_{j=1, \dots, N} \gamma_j \\ \text{sous contrainte de } \chi \\ \text{et } K(s) \text{ stabilise tout } \varnothing \end{aligned} \quad (3-96)$$

On obtient le résultat du Tableau 3.5.

<i>IT / run = 10000</i>	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>Rnd2Bst_DE</i>	-0,21	-0,19	0,009	385,13

Tableau 3.5 Optimisation des pondérations – robustesse de la stabilité

On constate que le temps de calcul du problème de robustesse des marges de stabilité est bien plus grand que l'approche sur modèle nominal : ceci est dû à un nombre d'évaluations N fois plus grand de la norme H_∞ de la boucle fermée.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-25.

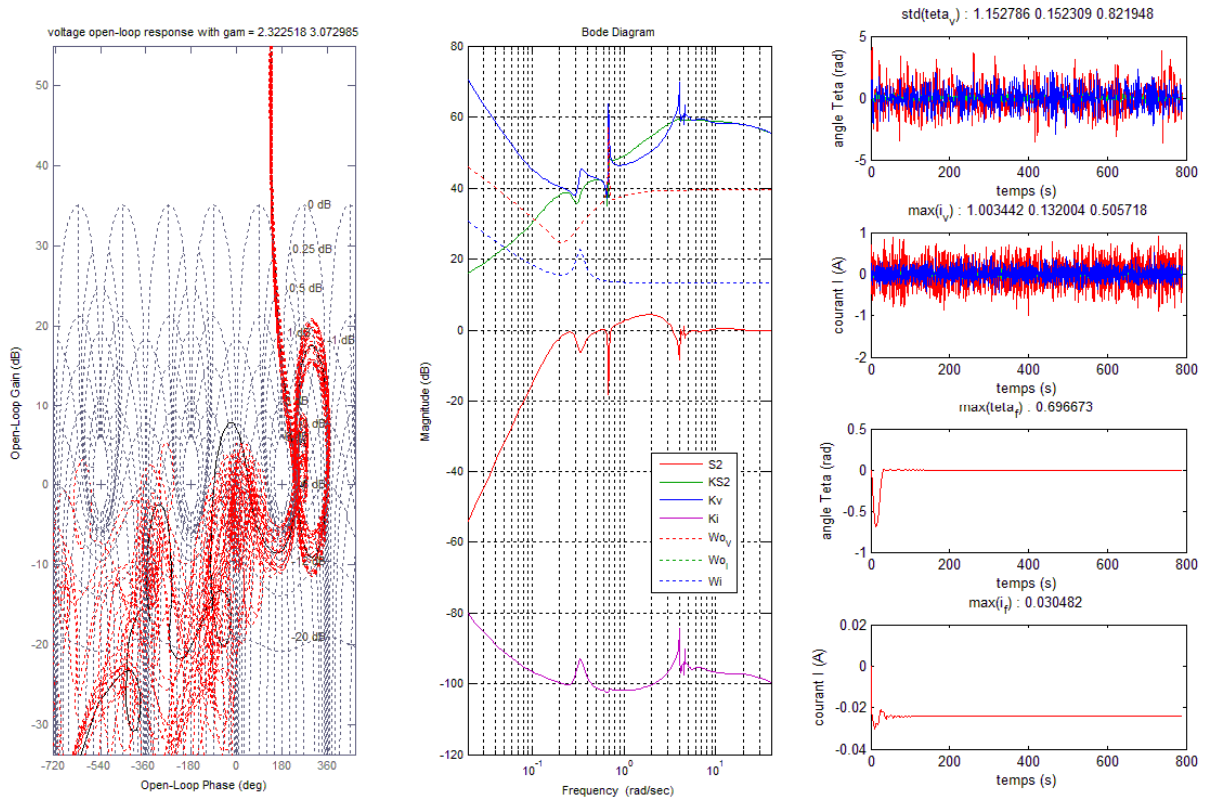


Figure 3-25: Problème de la robustesse des marges de stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.6.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	2,32 ; 3,07
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,69
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left i_f(t) \right \right) \leq i_{max}$	0,03
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,15
$\max_{t>0} \left(\left i_{vx}(t) \right \right) \leq i_{max}$	1,00
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,15
$\max_{t>0} \left(\left i_{vy}(t) \right \right) \leq i_{max}$	0,13
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,82
$\max_{t>0} \left(\left i_{vz}(t) \right \right) \leq i_{max}$	0,50

Tableau 3.6 Evaluation des exigences – robustesse de la stabilité

3.4.3.2. Le viseur de tir

On considère maintenant le viseur de tir dont la représentation fréquentielle de la fonction de transfert entre la tension de commande et la vitesse gyrométrique est donnée Figure 3-26. Ce système est d'ordre 20.

Pour des raisons de confidentialité, les fréquences et les bases temporelles des signaux mis en jeu dans cette étude ont été normalisées.

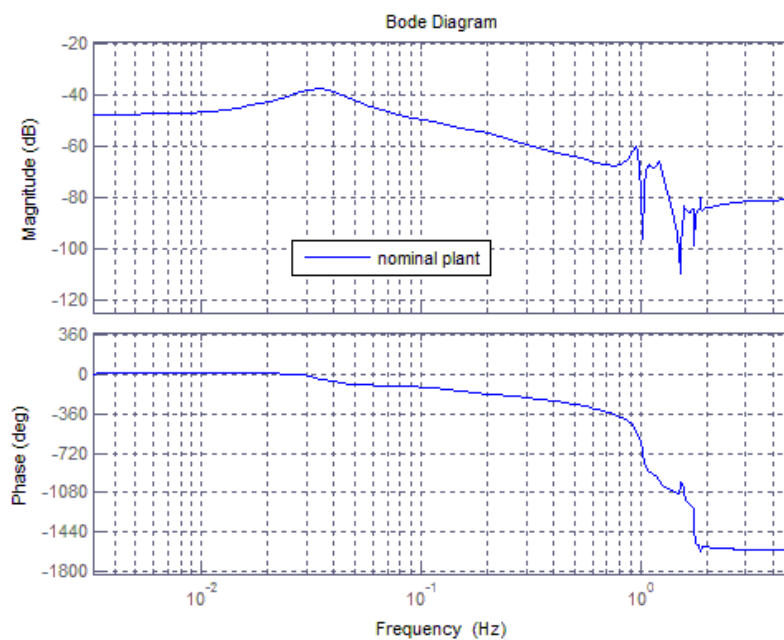


Figure 3-26: Réponse fréquentielle du viseur de tir

On considère cette fois-ci le schéma de commande de la Figure 3-27.

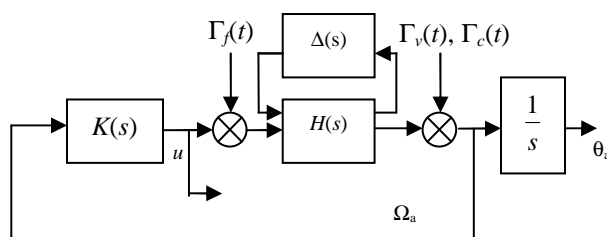


Figure 3-27: Schéma de commande du viseur

On dispose d'exigences de résiduel de stabilisation vis-à-vis des excitations vibratoires suivant les directions x et y , dont on peut disposer d'une acquisition temporelle $\Gamma_{vx}(t)$ et $\Gamma_{vy}(t)$ induite sur la vitesse (Figure 3-28).

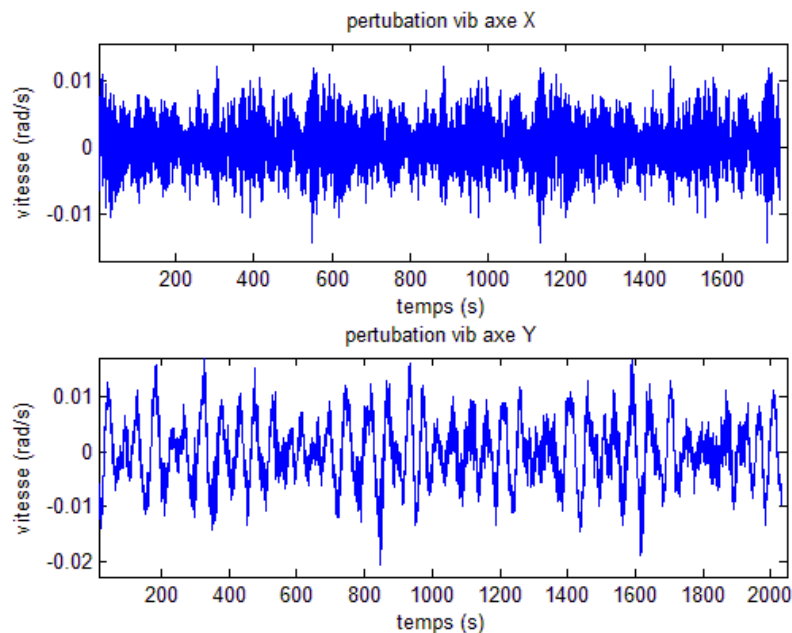


Figure 3-28: Acquisition temporelle de Γ_{vx} et Γ_{vy} .

On dispose aussi d'une exigence de rejet du frottement. Notons que l'on peut rendre le frottement homogène à une tension si l'on connaît la résistance et la constante de couple de la motorisation, ce qui permet de ramener le frottement $\Gamma_f(t)$ en entrée de modèle. Enfin plutôt que de limiter le courant, on va chercher à limiter la tension de commande u de sortie du correcteur à une valeur u_{max} .

En plus de ces exigences « nominales » de fonctionnement, il arrive qu'on dispose d'exigences « dégradées » en fonction de fonctionnalités particulières au viseur. Un exemple courant est le viseur de tir, où chaque tir peut occasionner « un choc » que le bouclage de la ligne de visée subit. Dans ce cas, pour éviter le dépointage occasionné par la mise en conditionnement du correcteur, on va chercher à limiter la tension de commande en réponse à ces chocs. Comme pour le fonctionnement nominal, il est possible de disposer d'acquisitions temporelles de la réponse de la ligne de visée (non bouclée) vis-à-vis des chocs, et ce suivant les 2 axes, que l'on nommera ici $\Gamma_{cx}(t)$ et $\Gamma_{cy}(t)$. Ces perturbations en sortie de modèle sont du type pseudo-amorti à la fréquence du premier mode fréquentiel de la structure du viseur comme le montre l'acquisition de la Figure 3-29.

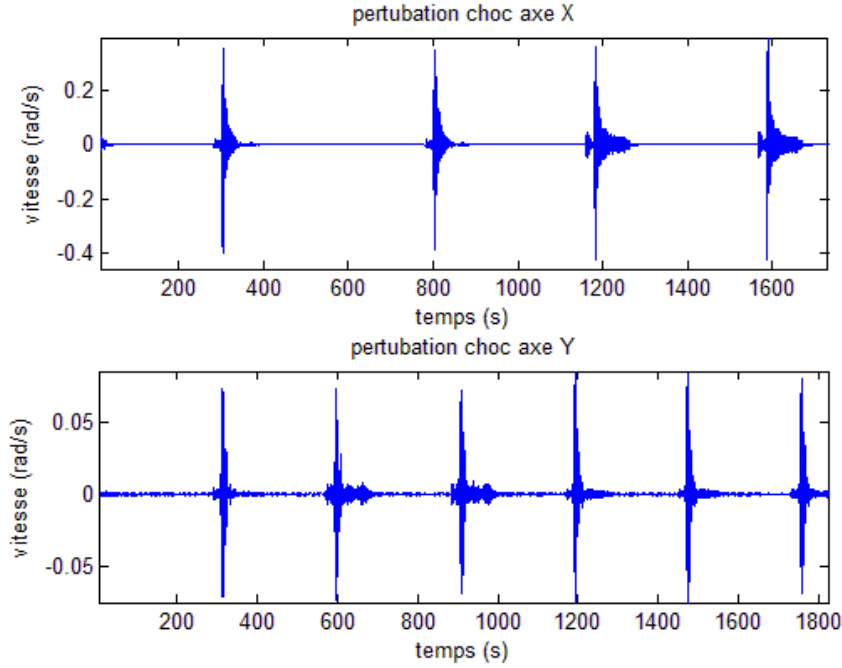


Figure 3-29: Acquisition des perturbations de type chocs

Comme il s'agit d'un mode dégradé, ces exigences supplémentaires ne disposent pas de la même priorité que les exigences nominales ce qui permet d'illustrer le mécanisme de gestion de priorité des exigences du §3.3.2. La spécification système considérée ici est donc la suivante :

$$\left. \begin{array}{l}
 \left. \begin{array}{l}
 |\theta_a(t)| \leq \bar{\theta}_f \text{ en réponse à l'échelon } \Gamma_f(t) \\
 \max_{t > T_f} |\theta_a(t)| \leq \varepsilon_f \text{ en réponse à l'échelon } \Gamma_f(t) \\
 |u(t)| \leq u_{\max} \text{ en réponse à l'échelon } \Gamma_f(t) \\
 \sigma_{\theta_x} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vx}(t) \\
 |u(t)| \leq u_{\max} \text{ en réponse à l'excitation } \Gamma_{vx}(t) \\
 \sigma_{\theta_y} \leq \bar{\theta}_v \text{ en réponse à l'excitation } \Gamma_{vy}(t) \\
 |u(t)| \leq u_{\max} \text{ en réponse à l'excitation } \Gamma_{vy}(t)
 \end{array} \right\} \text{ priorité forte} \\
 \\
 \left. \begin{array}{l}
 |u(t)| \leq u_{\max} \text{ en réponse à l'excitation } \Gamma_{cx}(t) \\
 |u(t)| \leq u_{\max} \text{ en réponse à l'excitation } \Gamma_{cy}(t)
 \end{array} \right\} \text{ priorité faible}
 \end{array} \right\} := \chi' \quad (3-97)$$

En posant :

- $\theta_f(t)$ la réponse en position à l'excitation $\Gamma_f(t)$,
- $u_f(t)$ la réponse en tension à l'excitation $\Gamma_f(t)$,

- $\theta_{vx}(t)$ la réponse en position à l'excitation $\Gamma_{vx}(t)$,
- $\theta_{vy}(t)$ la réponse en position à l'excitation $\Gamma_{vy}(t)$,
- $u_{vx}(t)$ la réponse en tension à l'excitation $\Gamma_{vx}(t)$,
- $u_{vy}(t)$ la réponse en tension à l'excitation $\Gamma_{vy}(t)$,
- $u_{cx}(t)$ la réponse en tension à l'excitation $\Gamma_{cx}(t)$,
- $u_{cy}(t)$ la réponse en tension à l'excitation $\Gamma_{cy}(t)$,

Alors la spécification se reformule aisément comme suit :

$$\left. \begin{array}{l}
 \max_{t>0} \left(\left| \theta_f(t) \right| \right) \leq \bar{\theta}_f, \max_{t>T_f} \left(\left| \theta_f(t) \right| \right) \leq \epsilon_f, \\
 \max_{t>0} \left(\left| u_f(t) \right| \right) \leq u_{\max} \\
 \sigma_{\theta_{vx}} \leq \bar{\theta}_v, \max_{t>0} \left(\left| u_{vx}(t) \right| \right) \leq u_{\max} \\
 \sigma_{\theta_{vy}} \leq \bar{\theta}_v, \max_{t>0} \left(\left| u_{vy}(t) \right| \right) \leq u_{\max}
 \end{array} \right\} \text{priorité forte}$$

$$\left. \begin{array}{l}
 \max_{t>0} \left(\left| u_{cx}(t) \right| \right) \leq u_{\max} \\
 \max_{t>0} \left(\left| u_{cy}(t) \right| \right) \leq u_{\max}
 \end{array} \right\} \text{priorité faible}$$

$$\left. \begin{array}{l}
 \left. \begin{array}{l}
 \max_{t>0} \left(\left| \theta_f(t) \right| \right) \leq \bar{\theta}_f, \max_{t>T_f} \left(\left| \theta_f(t) \right| \right) \leq \epsilon_f, \\
 \max_{t>0} \left(\left| u_f(t) \right| \right) \leq u_{\max} \\
 \sigma_{\theta_{vx}} \leq \bar{\theta}_v, \max_{t>0} \left(\left| u_{vx}(t) \right| \right) \leq u_{\max} \\
 \sigma_{\theta_{vy}} \leq \bar{\theta}_v, \max_{t>0} \left(\left| u_{vy}(t) \right| \right) \leq u_{\max}
 \end{array} \right\} := \chi' \\
 \left. \begin{array}{l}
 \max_{t>0} \left(\left| u_{cx}(t) \right| \right) \leq u_{\max} \\
 \max_{t>0} \left(\left| u_{cy}(t) \right| \right) \leq u_{\max}
 \end{array} \right\}
 \end{array} \right\} (3-98)$$

Pour des raisons de confidentialité, les spécifications numériques sont occultées. On mesurera la tenue des exigences en relatif par rapport au spécifié.

3.4.3.2.1. Optimisation des pondérations sur modèle nominal

Dans un premier temps, on se propose de chercher par *loop-shaping* le correcteur H_∞ SISO solution du problème d'optimisation suivant :

$$\min_{W_o(s), W_i(s) \in \mathbf{RH}_\infty} \gamma$$

sous contrainte de χ'

$$(3-99)$$

C'est à nouveau une application directe du problème (3-62) avec :

$$f_c(H_0, K) = \frac{\gamma_{obj}}{9} \left[\begin{aligned} & e^{\frac{\ln(\gamma_{obj})}{0.2\theta_f}(\max_{t>0}(|\theta_f(t)|) - \bar{\theta}_f)} + e^{\frac{\ln(\gamma_{obj})}{0.2\varepsilon_f}(\max_{t>T_f}(|\theta_f(t)|) - \varepsilon_f)} \\ & + e^{\frac{\ln(\gamma_{obj})}{0.2u_{max}}(\max_{t>0}(|u_f(t)|) - u_{max})} \\ & + e^{\frac{\ln(\gamma_{obj})}{0.2\theta_v}(\sigma_{\theta_{vx}} - \bar{\theta}_v)} + e^{\frac{\ln(\gamma_{obj})}{0.2u_{max}}(\max_{t>0}(|u_{vx}(t)|) - u_{max})} \\ & + e^{\frac{\ln(\gamma_{obj})}{0.2\theta_v}(\sigma_{\theta_{vy}} - \bar{\theta}_v)} + e^{\frac{\ln(\gamma_{obj})}{0.2u_{max}}(\max_{t>0}(|u_{vy}(t)|) - u_{max})} \\ & + \max\left(0, 1 + \frac{\ln(\gamma_{obj})}{0.2u_{max}}(\max_{t>0}(|u_{vx}(t)|) - u_{max})\right) \\ & + \max\left(0, 1 + \frac{\ln(\gamma_{obj})}{0.2u_{max}}(\max_{t>0}(|u_{vy}(t)|) - u_{max})\right) \end{aligned} \right] \quad (3-100)$$

$$\gamma_{obj} = 3$$

Comme les exigences relatives aux chocs sont de priorité plus faible, on choisit de développer les fonctions barrière les concernant, à l'ordre 1 (voir § 3.3.2).

Le correcteur étant SISO, les pondérations $W_\Omega(s)$ et $W_u(s)$ respectivement en sortie et en entrée de modèle le sont aussi. Nous choisissons les ordres suivants :

$$\partial^\circ W_\Omega = 7, \partial^\circ W_u = 1 \quad (3-101)$$

Ce qui conduit à 38 paramètres à optimiser.

Nous employons l'algorithme *Rnd2Bst_DE* car l'exemple précédent a montré que c'était l'algorithme le plus efficace pour le problème d'optimisation de pondérations.

La totalité du test a été réalisée sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

Chaque *run* fait l'objet de 2000 itérations.

On obtient le résultat du Tableau 3.7.

<i>IT / run = 2000</i>	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>Rnd2Bst_DE</i>	-0,21	-0,20	0,015	18,42

Tableau 3.7 Optimisation des pondérations du viseur de tir

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-30.

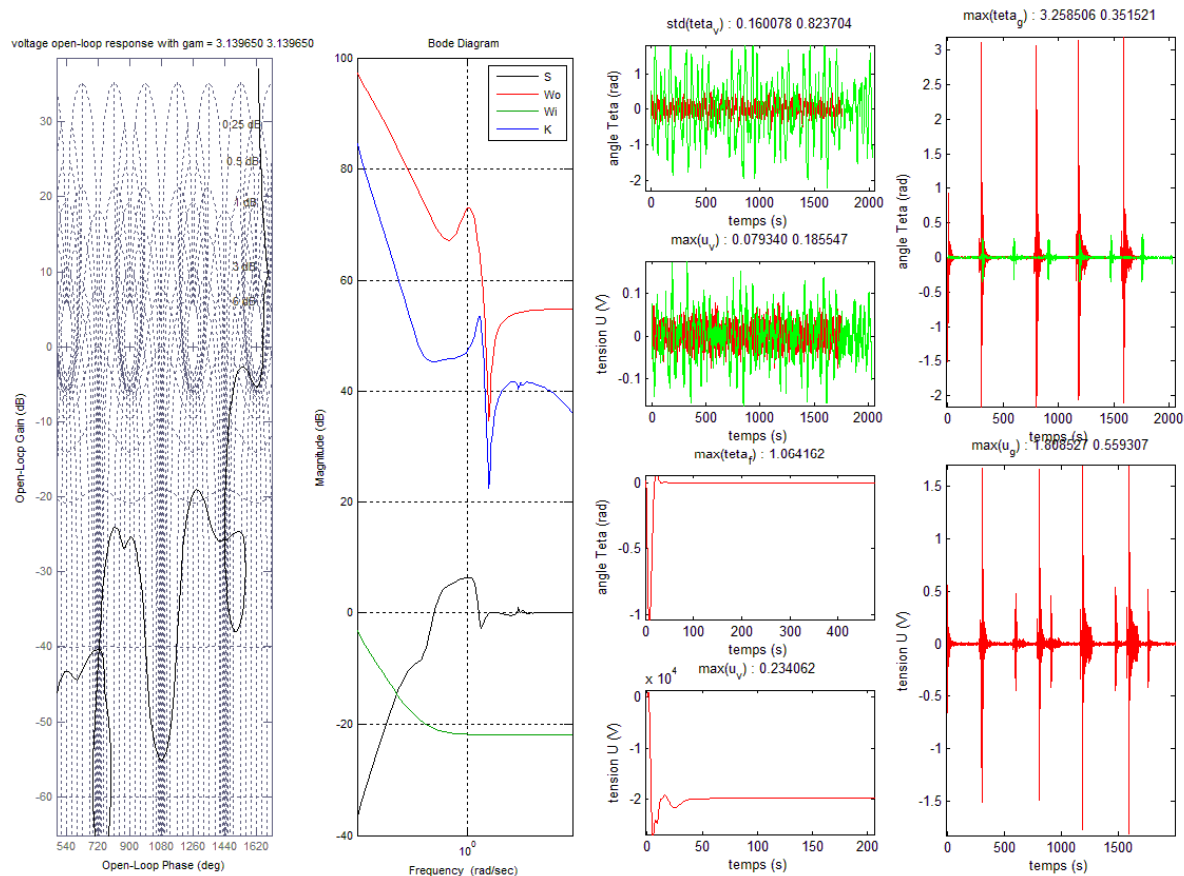


Figure 3-30: Optimisation des pondérations

Les résultats quant à l'évaluation finale des exigences temporelles sont rassemblés dans le Tableau 3.8. On rappelle que l'on présente pour des raisons de confidentialité les résultats relatifs aux valeurs maximales temporelles spécifiées (confidentielles), c'est-à-dire dire par rapport à 1.

<i>Exigences</i>	<i>Evaluations finales</i>
γ	3,13
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	1,06
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{\max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,16
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{\max}$	0,08
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,82
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{\max}$	0,18
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{\max}$	1,81
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{\max}$	0,56

Tableau 3.8 Evaluation des exigences à l'issue de l'optimisation des pondérations

On constate que les exigences de priorité haute sont quasiment toutes satisfaites. Une exigence de priorité basse n'est pas satisfaite mais elle reste néanmoins convenable : l'algorithme a cherché à optimiser au maximum les exigences de priorité faible une fois les exigences de priorité haute satisfaites (c'est ce qui explique le comportement de rejection du correcteur vers 2 rad/s, fréquence du mode propre du viseur sollicitée lors des chocs). En l'absence des exigences relatives à la tenue des chocs, cette réjection n'aurait pas été calculée.

3.4.3.2.2. Optimisation multi-modèles

A l'ensemble d'exigences χ' , on peut ajouter un besoin de robustesse en stabilité voire en performances sur un ensemble de N modèles centré sur le modèle nominal $H_0(s)$:

$$\mathcal{P} = \{H_i(s), i = 0, \dots, N\} \quad (3-102)$$

$N=25$ modèles sont obtenus par une technique strictement identique à celle du § 3.4.3.1.3 avec $\delta_{\omega} = \pm 15\%$, $\delta_{\xi} = \pm 15\%$, $\delta_R = \pm 10\%$. La représentation fréquentielle de la boule de modèle est tracée sur la Figure 3-31.

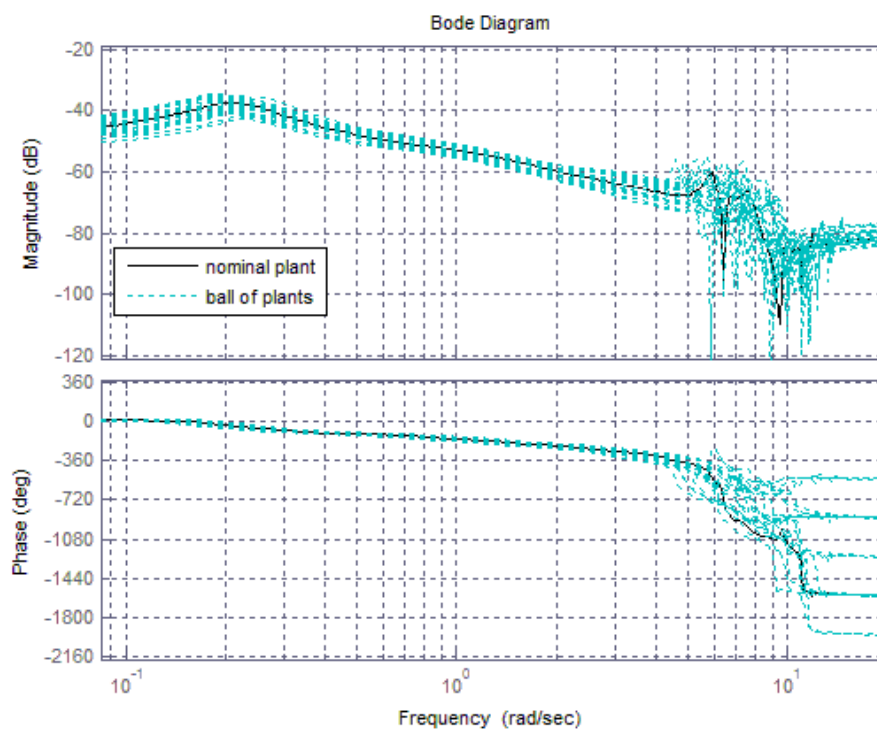


Figure 3-31: Boule de modèles

On constate sur la Figure 3-32 qu'en l'état, le correcteur obtenu précédemment est peu robuste vis-à-vis de la boule de modèles. De fait la valeur pire-cas de γ obtenu pour l'ensemble de la boule de modèle vaut 4,89. Aussi allons-nous chercher dans la suite à robustifier le correcteur par les techniques multi-modèles développées précédemment.

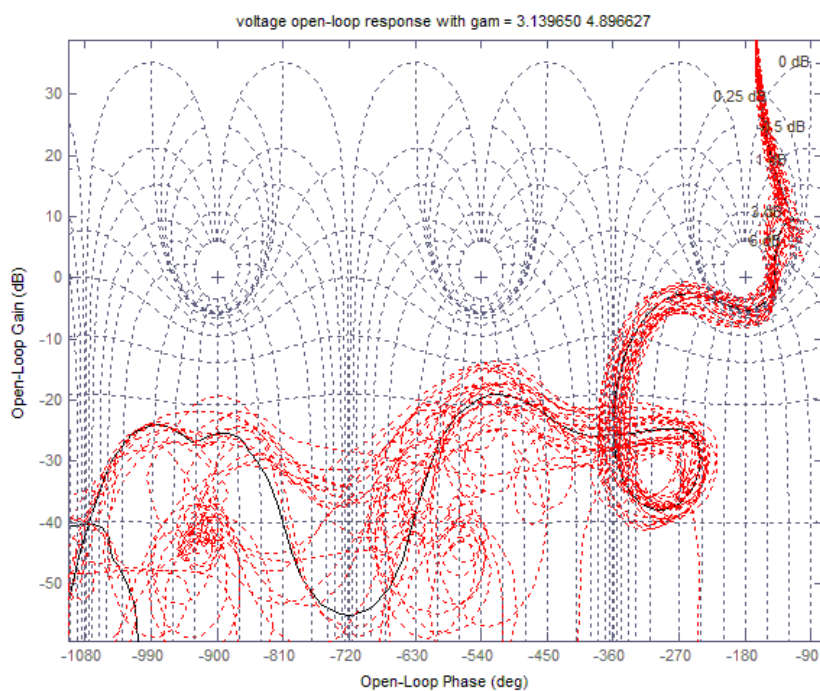


Figure 3-32: Examen de la robustesse des pondérations

On peut chercher (avec *Rnd_2_Bst*) un correcteur solution des problèmes suivants :

- problème de robustesse de la stabilité :

$$\min_{W_o(s), W_i(s) \in RH_\infty} \gamma$$

sous contrainte de χ'
et $K(s)$ stabilise tout φ

(3-103)

On obtient le résultat du Tableau 3.9.

<i>IT / run = 2000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,21	-0,20	-0,016	20,47

Tableau 3.9 Optimisation des pondérations – robustesse de la stabilité

On constate que les temps de calcul entre l’approche sur modèle nominal et le problème de robustesse en stabilité sont comparables.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-33.

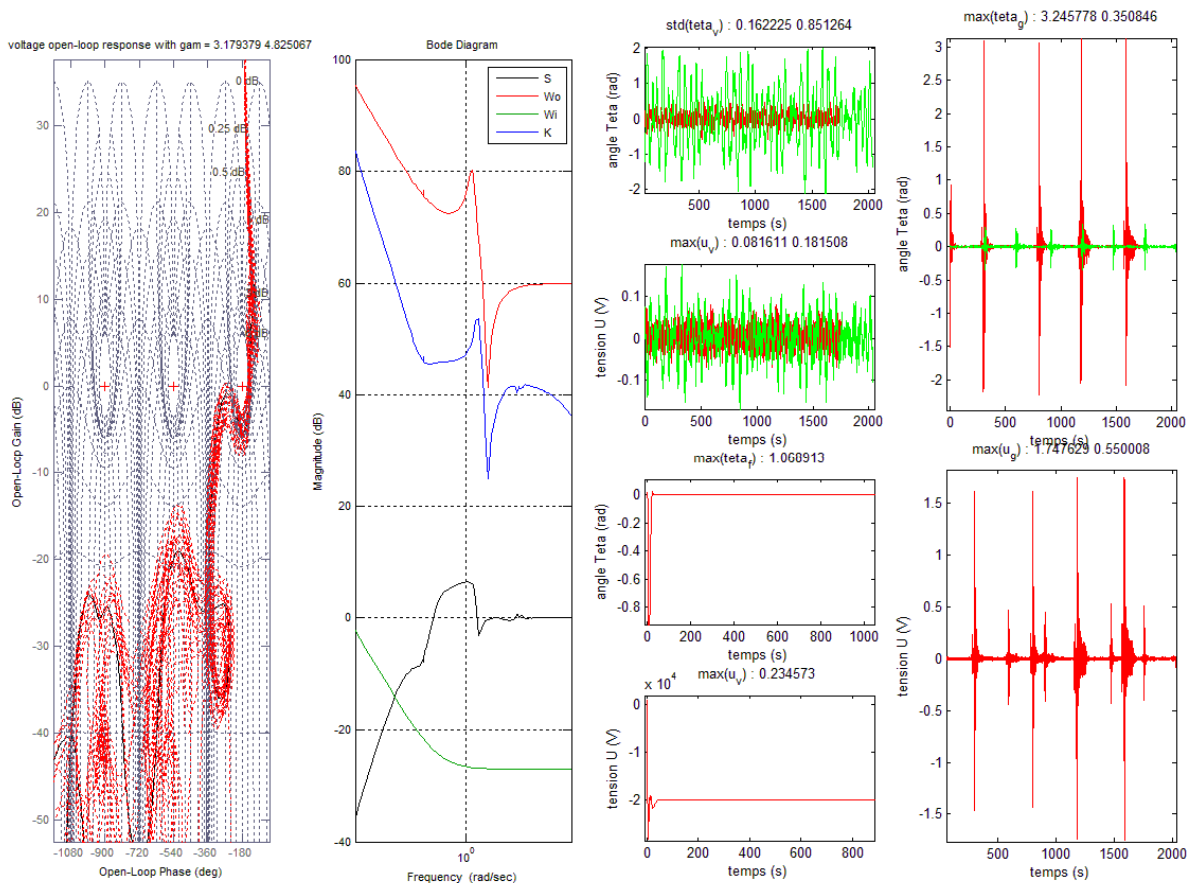


Figure 3-33: Problème de la robustesse de la stabilité

L’évaluation finale des exigences est reportée sur le Tableau 3.10.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	3,17 ; 4,82
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	1,07
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,16
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{max}$	0,08
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,85
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{max}$	0,18
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{max}$	1,74
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{max}$	0,55

Tableau 3.10 Evaluation des exigences – robustesse de la stabilité

On constate que la plupart des exigences sont satisfaites (un peu dégradées toutefois, l'augmentation de la robustesse s'accompagne d'une diminution de performances). La valeur de γ pire-cas est de 4,82.

- problème de robustesse des marges de stabilité :

$$\min_{W_o(s), W_f(s) \in RH_\infty} \gamma, \gamma = \max_{j=1, \dots, N} \gamma_j$$

sous contrainte de χ'
et $K(s)$ stabilise tout φ

(3-104)

On obtient le résultat du Tableau 3.11.

<i>IT / run = 2000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,16	-0,15	0,011	200,40

Tableau 3.11 Optimisation des pondérations – robustesse de la stabilité

On constate que le temps de calcul du problème de robustesse des marges de stabilité est bien plus grand que l'approche sur modèle nominal : ceci est dû à un nombre d'évaluations N fois plus grand de la norme H_∞ de la boucle fermée.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-34.

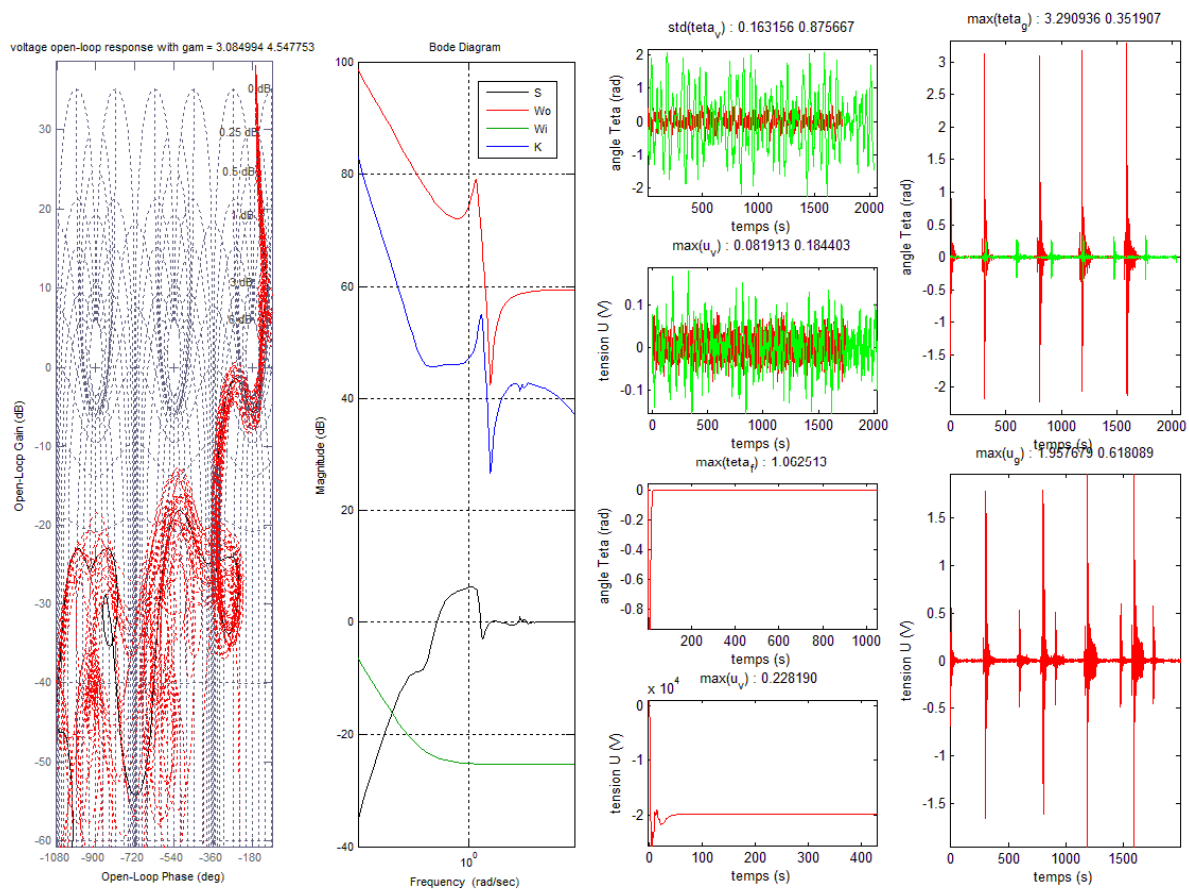


Figure 3-34: Problème de la robustesse des marges de stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.12.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	3,08 ; 4,54
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	1,06
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,16
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{max}$	0,082
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,88
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{max}$	0,18
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{max}$	1,95
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{max}$	0,62

Tableau 3.12 Evaluation des exigences – robustesse de la stabilité

3.4.3.3. Pour aller plus loin...

Les exigences système haut niveau d'un dispositif optronique inertielle portent sur sa Fonction de Transfert de Modulation (FTM) globale car la FTM fournit une présentation synthétique du contraste et de la résolution d'un système optique ou optronique [MAY 99], comme l'illustre la figure 1. Ainsi cette dernière permet de quantifier la qualité de l'image (flou) et donc la portée du dispositif considéré.

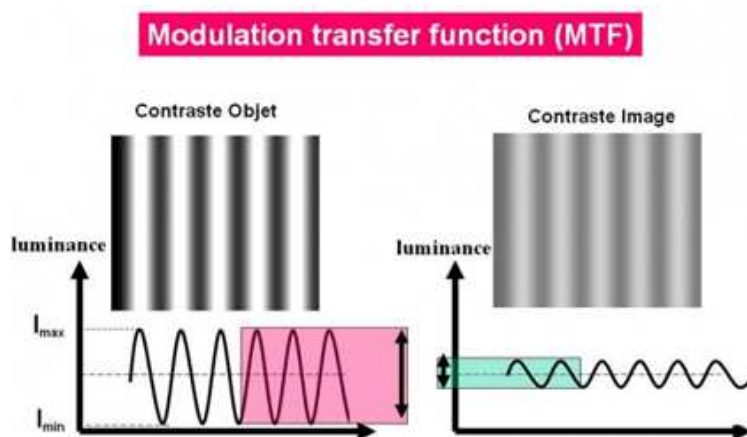


Figure 3-35: FTM et contraste

Dans le cas d'un système optronique complexe, la FTM globale du système est égale au produit des FTM des sous-systèmes le composant [MAY 99]-[HOL 06].

$$FTM_{\text{systeme}}(\nu) = \Pi(FTM_{\text{elements}}) \quad (3-105)$$

Or, un contributeur majoritaire de la dégradation des FTM est le résiduel de stabilisation du dispositif. Ainsi dans le cas d'un viseur, il est d'usage de décliner la spécification relative à la portée du viseur en une spécification sur l'écart-type du résiduel de stabilisation qui ne doit pas dépasser la taille d'un demi-pixel afin de limiter le flou engendré. En effet, en supposant que le résiduel de stabilisation est un bruit blanc gaussien, alors la FTM induite (qui dégrade la FTM globale du viseur, donc sa portée) s'écrit :

$$FTM_{\text{stab}}(\nu) = e^{-2\pi^2\sigma_{\theta\nu}^2\nu^2} \quad (3-106)$$

Où $\sigma_{\theta\nu}$ est l'écart-type du mouvement aléatoire (en rad) et ν la fréquence spatiale en (cycle/rad).

Cette spécification peut s'avérer d'autant plus conservatrice que les portées recherchées sont grandes car elle suppose que le résiduel de stabilisation est blanc et gaussien ce qui n'est jamais le cas de par la présence de composantes spectrales diverses en fonction de l'environnement mécanique du viseur (hélicoptère etc...) et du paysage modale de la structure mécanique du viseur : de ce fait, il est de plus en plus difficile expérimentalement d'approximer la FTM du résiduel de stabilisation par une gaussienne pour des résiduels de plus en plus petits. Se pose alors la question du conservatisme de la spécification envers l'étage de stabilisation du résiduel de stabilisation en écart-type.

Dans la suite, on suppose disposer d'une FTM spécifiée à l'étage de stabilisation notée $ftm_{obj}(\nu)$ sans préjuger de la nature des résiduels de stabilisation. Dès lors rien n'empêche l'emploi d'un critère sur l'écart de FTM entre l'étage de stabilisation bouclé par le correcteur et la FTM spécifiée. En effet, la spécification est satisfaite si la FTM du résiduel de stabilisation reste au-dessus du gabarit spécifié. Dans ce cas, il suffit de remplacer les exigences relatives aux écart-type de résiduel de stabilisation par les exigences relatives aux FTM suivantes¹⁸ :

¹⁸ La division $\left| \frac{ftm_{obj}(\nu)}{ftm_{\theta\nu}(\nu)} \right|$ est réalisée terme à terme.

$$\left. \begin{aligned}
& \max_{t>0} \left(\left| \theta_f(t) \right| \right) \leq \bar{\theta}_f, \max_{t>T_f} \left(\left| \theta_f(t) \right| \right) \leq \varepsilon_f, \max_{t>0} \left(\left| i_f(t) \right| \right) \leq i_{\max} \\
& \max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vx}}(v)} \right| \right) < 1, \max_{t>0} \left(\left| i_{vx}(t) \right| \right) \leq i_{\max} \\
& \max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vy}}(v)} \right| \right) < 1, \max_{t>0} \left(\left| i_{vy}(t) \right| \right) \leq i_{\max} \\
& \max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vz}}(v)} \right| \right) < 1, \max_{t>0} \left(\left| i_{vz}(t) \right| \right) \leq i_{\max}
\end{aligned} \right\} := \chi \quad (3-107)$$

Ainsi on peut reprendre le calcul du correcteur avec :

$$f_c(H_0, K) = \frac{\gamma_{obj}}{9} \left[\begin{aligned}
& e^{\frac{\ln(\gamma_{obj})}{0.2\bar{\theta}_f} \left(\max_{t>0} \left(\left| \theta_f(t) \right| \right) - \bar{\theta}_f \right)} + e^{\frac{\ln(\gamma_{obj})}{0.2\varepsilon_f} \left(\max_{t>T_f} \left(\left| \theta_f(t) \right| \right) - \varepsilon_f \right)} \\
& + e^{\frac{\ln(\gamma_{obj})}{0.2i_{\max}} \left(\max_{t>0} \left(\left| i_f(t) \right| \right) - i_{\max} \right)} \\
& + e^{\frac{\ln(\gamma_{obj})}{0.2} \left(\max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vx}}(v)} \right| \right) - 1 \right)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{\max}} \left(\max_{t>0} \left(\left| i_{vx}(t) \right| \right) - i_{\max} \right)} \\
& + e^{\frac{\ln(\gamma_{obj})}{0.2} \left(\max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vy}}(v)} \right| \right) - 1 \right)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{\max}} \left(\max_{t>0} \left(\left| i_{vy}(t) \right| \right) - i_{\max} \right)} \\
& + e^{\frac{\ln(\gamma_{obj})}{0.2} \left(\max_v \left(\left| \frac{ftm_{obj}(v)}{ftm_{\theta_{vz}}(v)} \right| \right) - 1 \right)} + e^{\frac{\ln(\gamma_{obj})}{0.2i_{\max}} \left(\max_{t>0} \left(\left| i_{vz}(t) \right| \right) - i_{\max} \right)}
\end{aligned} \right] \quad (3-108)$$

$\gamma_{obj} = 3$

On trouvera en Annexe B le moyen de calculer la FTM à partir d'un signal temporel.

L'inconvénient de cette approche ne réside pas au niveau de sa faisabilité car une solution sera trouvée mais plutôt au niveau du temps de calcul mis pour la trouver. Partant de ce constat, une thèse sur l'optimisation de fonctions coûteuses suivant ce type de critères est en cours, dans le cadre à nouveau d'une collaboration entre Sagem et CentraleSupélec.

3.5. Réglage optimal et simultané des pondérations et du correcteur final à ordre et structure fixés

3.5.1. Introduction

Un inconvénient inhérent aux méthodes modernes de synthèse de correcteurs est que l'ordre du correcteur obtenu est grand (il est directement relié à l'ordre des pondérations et du système). Ainsi l'ingénieur doit post-simplifier le correcteur obtenu : même si le correcteur d'ordre plein est satisfaisant, se pose alors la question de la robustesse et de la performance du correcteur réduit obtenu vis-à-vis du problème (3-4). L'ingénieur doit donc contraindre l'ordre du correcteur *a priori* et complète ainsi sa problématique comme suit :

$$\begin{aligned} & \min_{W_o(s), W_i(s)} \gamma \\ & \text{sous contraintes que :} \\ & K(s) \text{ d'ordre } n_k, \text{ stabilisant et assure la} \\ & \text{robustesse désirée,} \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m. \end{aligned} \quad (3-109)$$

Il est à remarquer que le problème H_∞ (3-58) est équivalent au problème (3-110) [MCF 92] :

$$K(s) = \arg \min \gamma, \left\| \begin{bmatrix} W_o(s) \\ W_i(s)^{-1} K(s) \end{bmatrix} (I + H(s)K(s))^{-1} \begin{bmatrix} W_o(s)^{-1} & H(s)W_i(s) \end{bmatrix} \right\|_\infty < \gamma \quad (3-110)$$

L'emploi de cette formulation « standard équivalente » impose à l'ordre plein l'emploi de la γ -itération [DOY 89] pour le calcul du correcteur qui est beaucoup plus longue que la simple résolution du problème de stabilisation robuste du loop-shape [MCF 90].

Par ailleurs elle nécessite la stabilité des pondérations et de leur inverse.

Remarquons que :

$$\begin{aligned} W_i(s) &= \begin{bmatrix} A_i & | & B_i \\ \hline C_i & | & D_i \end{bmatrix} \Rightarrow W_i(s)^{-1} = \begin{bmatrix} A_i - B_i D_i^{-1} C_i & | & B_i D_i^{-1} \\ \hline D_i^{-1} C_i & | & D_i^{-1} \end{bmatrix} \\ W_o(s) &= \begin{bmatrix} A_o & | & B_o \\ \hline C_o & | & D_o \end{bmatrix} \Rightarrow W_o(s)^{-1} = \begin{bmatrix} A_o - B_o D_o^{-1} C_o & | & B_o D_o^{-1} \\ \hline D_o^{-1} C_o & | & D_o^{-1} \end{bmatrix} \end{aligned} \quad (3-111)$$

En posant :

$$\begin{aligned}\bar{\lambda}_o &= \max_j \left(\operatorname{Re}(\lambda_j(A_o)) \right), \underline{\lambda}_o = \max_j \left(\operatorname{Re}(\lambda_j(A_o - B_o D_o^{-1} C_o)) \right) \\ \bar{\lambda}_i &= \max_j \left(\operatorname{Re}(\lambda_j(A_i)) \right), \underline{\lambda}_i = \max_j \left(\operatorname{Re}(\lambda_j(A_i - B_i D_i^{-1} C_i)) \right)\end{aligned}\quad (3-112)$$

Alors le cadre de résolution du problème (3-62) peut être étendu au problème (3-109), mais au prix de temps de calcul plus conséquent. Il suffit pour cela de déterminer à chaque itération le correcteur final $K(s)$ par les techniques de synthèse structurée présentées en première partie en considérant le problème d'optimisation suivant :

$$\begin{aligned}& \min_{\Theta_i, \Theta_o} f(\Theta_i, \Theta_o) \\ & \text{si } \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i) \geq 0 \\ & \quad f(\Theta_i, \Theta_o) = \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i) \\ & \text{sinon} \\ & \quad K(s) = \arg \min \gamma, \\ & \quad \left\| \begin{bmatrix} W_o(s) \\ W_i(s)^{-1} K(s) \end{bmatrix} (I + H(s)K(s))^{-1} \begin{bmatrix} W_o(s)^{-1} & H(s)W_i(s) \end{bmatrix} \right\|_{\infty} < \gamma \\ & \quad f(\Theta_i, \Theta_o) = -\frac{1}{\gamma + f_c(H, K)}\end{aligned}\quad (3-113)$$

Toutefois cette approche de la synthèse à ordre fixé présente des inconvénients majeurs :

- l'usage des techniques de synthèse structurée à chaque itération, telle que l'optimisation non lisse, présentées en première partie pour le calcul de $K(s)$ rend le processus d'optimisation très long car il convient à chaque fois de lancer plusieurs *run* pour le calcul de $K(s)$;
- utiliser de tels algorithmes pour le calcul de $K(s)$ rend la fonction $f(\Theta_i, \Theta_o)$ aléatoire diminuant ainsi l'efficacité de l'optimisation du problème global. En effet, partant de conditions initiales toujours différentes pour le calcul de $K(s)$, on n'obtient jamais le même résultat. Il faudrait imposer une condition initiale toujours identique à chaque itération, mais cela diminuerait alors l'efficacité du calcul de $K(s)$.

Or nous avons vu en première partie comment synthétiser un correcteur à ordre fixé par l'optimisation stochastique, approche dont nous avons montré la souplesse lors de la prise en compte de contraintes complexes. Ainsi les contraintes du problème (3-109) peuvent être prises en compte de la même manière, nous permettant ainsi d'aboutir à une méthode puissante permettant le calcul de correcteurs à ordre et structure fixés sur la base d'une spécification de forme quelconque. Nous verrons plus bas que dans le cadre de l'approche *loop-shaping*, la mise en œuvre de pondérations fréquentielles pour le calcul du correcteur (final) à ordre fixé devient même caduque ce qui soulage d'autant le processus d'optimisation.

Pour traiter la contrainte d'ordre, deux approches sont possibles :

- Soit l'optimisation de l'ordre porte sur le précorrecteur $K_p(s)$: cette approche est possible mais elle entraîne que l'ordre du correcteur final est au minimum égal à celui des pondérations :

$$\partial^\circ K = \partial^\circ K_p + \partial^\circ W_i + \partial^\circ W_o \quad (3-114)$$

Ainsi pour maîtriser l'ordre du correcteur final, on peut agir sur l'ordre du précorrecteur et sur celui des pondérations ce qui peut s'avérer complexe.

- Soit l'optimisation de l'ordre porte directement sur le correcteur final $K(s)$: c'est cette approche que nous envisageons ici.

3.5.2. Vers une élimination des pondérations

3.5.2.1. Optimisation sur modèle nominal

On reprend ici l'approche que nous avons proposée dans [FEY 13a] ou on cherche simultanément les pondérations et le correcteur final en résolvant le problème suivant :

$$\begin{aligned} & \min_{W_o(s), W_i(s), K(s)} \gamma \\ & \text{sous contraintes que :} \\ & W_o(s), W_i(s) \in \mathbf{RH}_\infty, \\ & W_o(s)^{-1}, W_i(s)^{-1} \in \mathbf{RH}_\infty, \\ & T(s) \in \mathbf{RH}_\infty, \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m. \end{aligned} \quad (3-115)$$

Avec :

$$T(s) = \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array} \right] = \left[\begin{array}{c} W_o(s) \\ W_i(s)^{-1} K(s) \end{array} \right] (I + H(s)K(s))^{-1} \left[\begin{array}{cc} W_o(s)^{-1} & H(s)W_i(s) \end{array} \right] \quad (3-116)$$

Et

$$\gamma = \|T(s)\|_\infty \quad (3-117)$$

On note pour simplifier :

$$\Theta_k = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix}, \quad \Theta_i = \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix}, \quad \Theta_o = \begin{pmatrix} A_o & B_o \\ C_o & D_o \end{pmatrix} \quad (3-118)$$

En posant :

$$\bar{\lambda}_{cl} = \max_j (\operatorname{Re}(\lambda_j(A_{cl}))) \quad (3-119)$$

Alors on résout le problème (3-115) en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{\Theta_i, \Theta_o, \Theta_k} f(\Theta_i, \Theta_o, \Theta_k) \\ \text{si } & \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i, \bar{\lambda}_{cl}) \geq 0 \\ & f(\Theta_i, \Theta_o, \Theta_k) = \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i, \bar{\lambda}_{cl}) \\ \text{sinon} & \\ & \gamma = \|T(s)\|_{\infty} \\ & f(\Theta_i, \Theta_o, \Theta_k) = -\frac{1}{\gamma + f_c(H, K)} \\ \text{fin si} & \end{aligned} \quad (3-120)$$

Cette structure est identique à celles des problèmes H_{∞} structurés avec contraintes présentées en première partie dont nous avons déjà montré l'efficacité. C'est la souplesse de l'approche proposée en première partie qui permet cette mise en œuvre relativement simple.

A l'usage, on se rend vite compte que le correcteur final dispose de toute l'information fréquentielle nécessaire pour garantir la spécification initiale (3-1) et que l'usage des pondérations fréquentielles devient caduc car ces dernières tendent à devenir constantes. On a donc tout intérêt à simplifier le problème (3-120) en une optimisation simultanée du correcteur et de pondérations statiques (*scaling*), c'est-à-dire :

$$\begin{aligned} & \min_{D_o, D_i, K(s)} \gamma \\ & \text{sous contraintes que :} \\ & T(s) \in \mathbf{RH}_{\infty}, \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m. \end{aligned} \quad (3-121)$$

Avec :

$$T(s) = \begin{bmatrix} A_{cl} & B_{cl} \\ C_{cl} & D_{cl} \end{bmatrix} = \begin{bmatrix} D_o \\ D_i^{-1} K(s) \end{bmatrix} (I + H(s)K(s))^{-1} \begin{bmatrix} D_o^{-1} & H(s)D_i \end{bmatrix} \quad (3-122)$$

Et

$$\gamma = \|T(s)\|_{\infty} \quad (3-123)$$

On résout alors le problème (3-121) en considérant le problème d'optimisation suivant :

$$\begin{aligned}
 & \min_{D_i, D_o, \Theta_k} f(D_i, D_o, \Theta_k) \\
 & \text{si } \bar{\lambda}_{cl} \geq 0 \\
 & \quad f(D_i, D_o, \Theta_k) = \bar{\lambda}_{cl} \\
 & \text{sinon} \\
 & \quad \gamma = \|T(s)\|_{\infty} \\
 & \quad f(D_i, D_o, \Theta_k) = -\frac{1}{\gamma + f_c(H, K)} \\
 & \text{fin si}
 \end{aligned} \tag{3-124}$$

Cette simplification est d'importance car :

- le nombre d'inconnues est grandement réduit puisque le « côté dynamique » des pondérations n'est plus à chercher, il s'en suit donc une augmentation de la rapidité du processus d'optimisation ;
- la phase de stabilisation des pondérations et de leur inverse n'existe plus ce qui raccourcit d'autant l'optimisation.

L'emploi des *scalings* statiques demeure par contre indispensable afin de pouvoir évaluer γ qui conserve toutes ses propriétés dues à la structure du problème H_{∞} particulier qu'est la synthèse H_{∞} par loop-shaping (c'est-à-dire un problème 4 blocs avec les pondérations et leur inverse deux à deux) : la fonction $f_c(H, K)$ garde alors tout son sens. Mais sans usage des scalings (par exemple en les imposant identitaires), on n'a plus aucune idée de la valeur de γ_{opt} , donc non plus du réglage des pénalités, et donc finalement du comportement du processus d'optimisation.

3.5.2.2. Optimisation multi-modèles

Dans ce paragraphe, on suppose disposer de la boule de modèles suivante centrée sur un modèle nominal $H_0(s)$:

$$\wp = \{H_i(s), i = 0, \dots, N\} \tag{3-125}$$

3.5.2.2.1. Optimisation robuste en stabilité

L'optimisation robuste en stabilité consiste à résoudre le problème suivant :

$$\begin{aligned}
 & \min_{D_o, D_i, K(s)} \gamma \\
 & \text{sous contraintes que :} \\
 & \quad T_j(s) \in \mathbf{RH}_{\infty}, j = 0, \dots, N \\
 & \quad Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
 \end{aligned} \tag{3-126}$$

Avec :

$$T_j(s) = \begin{bmatrix} A_{clj} & B_{clj} \\ C_{clj} & D_{clj} \end{bmatrix} = \begin{bmatrix} D_o \\ D_i^{-1}K(s) \end{bmatrix} (I + H_j(s)K(s))^{-1} \begin{bmatrix} D_o^{-1} & H_j(s)D_i \end{bmatrix} \quad (3-127)$$

Et :

$$\gamma = \|T_0(s)\|_{\infty} \quad (3-128)$$

En posant :

$$\bar{\lambda}_{cl} = \max_{j=0, \dots, N} \left[\max_i \left(\operatorname{Re}(\lambda_i(A_{clj})) \right) \right] \quad (3-129)$$

Alors on résout le problème (3-126) en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{D_i, D_o, \Theta_k} f(D_i, D_o, \Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq 0 \\ & \quad f(D_i, D_o, \Theta_k) = \bar{\lambda}_{cl} \\ & \text{sinon} \\ & \quad \gamma = \|T_0(s)\|_{\infty} \\ & \quad f(D_i, D_o, \Theta_k) = -\frac{1}{\gamma + f_c(H_0, K)} \\ & \text{fin si} \end{aligned} \quad (3-130)$$

3.5.2.2.2. Optimisation robuste en marge de stabilité

L'optimisation robuste en marge de stabilité consiste à résoudre le problème suivant :

$$\begin{aligned} & \min_{D_o, D_i, K(s)} \gamma, \gamma = \max_{i=0, \dots, N} \gamma_i \\ & \text{sous contraintes que :} \\ & T_j(s) \in \mathbf{RH}_{\infty}, j = 0, \dots, N \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \text{ pour } H_0(s) \end{aligned} \quad (3-131)$$

Avec :

$$T_j(s) = \begin{bmatrix} A_{clj} & B_{clj} \\ C_{clj} & D_{clj} \end{bmatrix} = \begin{bmatrix} D_o \\ D_i^{-1}K(s) \end{bmatrix} (I + H_j(s)K(s))^{-1} \begin{bmatrix} D_o^{-1} & H_j(s)D_i \end{bmatrix} \quad (3-132)$$

Et :

$$\gamma_j = \|T_j(s)\|_\infty \quad (3-133)$$

En posant :

$$\bar{\lambda}_{cl} = \max_{j=0,\dots,N} \left[\max_i \left(\operatorname{Re}(\lambda_i(A_{clj})) \right) \right] \quad (3-134)$$

Alors on résout le problème (3-126) en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{D_i, D_o, \Theta_k} f(D_i, D_o, \Theta_k) \\ & \text{si } \bar{\lambda}_{cl} \geq 0 \\ & \quad f(D_i, D_o, \Theta_k) = \bar{\lambda}_{cl} \\ & \text{sinon} \\ & \quad \gamma = \max_{j=0,\dots,N} \gamma_j, \quad \gamma_j = \|T_j(s)\|_\infty \\ & \quad f(D_i, D_o, \Theta_k) = -\frac{1}{\gamma + f_c(H_0, K)} \\ & \text{fin si} \end{aligned} \quad (3-135)$$

3.5.2.2.3. Optimisation robuste en performances

L'optimisation robuste en performance consiste à résoudre le problème suivant :

$$\begin{aligned} & \min_{D_o, D_i, K(s)} \gamma, \quad \gamma = \max_{i=0,\dots,N} \gamma_i \\ & \text{sous contraintes que} \\ & \left. \begin{aligned} & T(s) \in \mathbf{RH}_\infty, \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned} \right\} \forall H_j(s) \in \mathcal{H} \end{aligned} \quad (3-136)$$

Avec :

$$T_j(s) = \left[\begin{array}{c|c} A_{clj} & B_{clj} \\ \hline C_{clj} & D_{clj} \end{array} \right] = \left[\begin{array}{c} D_o \\ D_i^{-1} K(s) \end{array} \right] (I + H_j(s)K(s))^{-1} \left[\begin{array}{cc} D_o^{-1} & H_j(s)D_i \end{array} \right] \quad (3-137)$$

En posant :

$$\bar{\lambda}_{cl} = \max_{j=1,\dots,N} \left[\max_i \left(\operatorname{Re}(\lambda_i(A_{clj})) \right) \right] \quad (3-138)$$

Alors on résout le problème (3-126) en considérant le problème d'optimisation suivant :

$$\begin{aligned}
 & \min_{D_i, D_o, \Theta_k} f(D_i, D_o, \Theta_k) \\
 & \text{si } \bar{\lambda}_{cl} \geq 0 \\
 & \quad f(D_i, D_o, \Theta_k) = \bar{\lambda}_{cl} \\
 & \text{sinon} \\
 & \quad \gamma_j = \|T_j(s)\|_{\infty}, \quad j = 0, \dots, N \\
 & \quad f(D_i, D_o, \Theta_k) = -\frac{1}{\max_{j=0, \dots, N} [\gamma_j + f_c(H_j, K)]} \\
 & \text{fin si}
 \end{aligned} \tag{3-139}$$

3.5.2.3. Exemple applicatif (suite)

Dans ce paragraphe, on reprend les exemples données au §3.4.3. Les cahiers des charges des différents viseurs sont identiques et les problèmes que l’on cherche à résoudre aussi. Ainsi les fonctions $f(H,K)$ déjà définies restent inchangées. De même les modèles de synthèse considérés (nominal et boule de modèles) restent identiques.

La différence concerne le fait qu’on optimise directement le correcteur structuré sans passer par les pondérations fréquentielles. Il s’agit d’une application directe de l’algorithme (3-124) et des extensions multi-modèles.

3.5.2.3.1. Le viseur d’observation

Cette fois-ci on cherche directement le correcteur sous forme structurée en cascade comme sur la Figure 3-36.

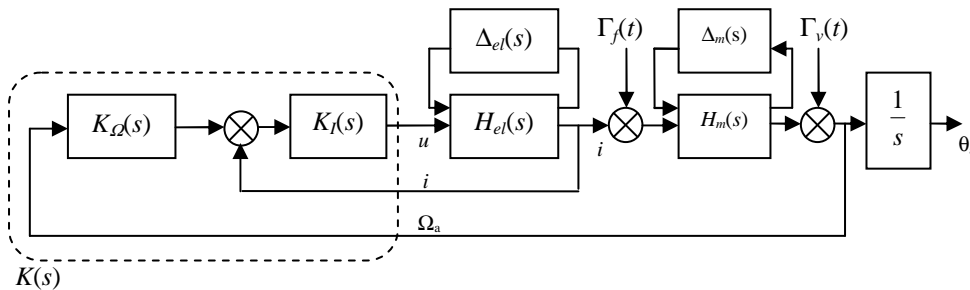


Figure 3-36: Schéma de commande du viseur

On choisit d’optimiser les correcteurs aux ordres suivants :

$$\partial K_I = 4, \quad \partial K_{\Omega} = 12 \tag{3-140}$$

Noter que :

$$K(s) = K_I(s)(1 - K_\Omega(s)) \quad (3-141)$$

Ainsi en posant :

$$K_I(s) := \begin{pmatrix} A_I & B_I \\ C_I & D_I \end{pmatrix} = \Theta_I, \quad K_\Omega(s) := \begin{pmatrix} A_\Omega & B_\Omega \\ C_\Omega & D_\Omega \end{pmatrix} = \Theta_\Omega \quad (3-142)$$

Alors :

$$K(s) := \begin{pmatrix} A_K & C_K \\ B_K & D_K \end{pmatrix} = \Theta_K = \begin{pmatrix} A_I & B_I C_\Omega & B_I D_\Omega & B_I \\ 0 & A_\Omega & B_\Omega & 0 \\ C_I & D_I C_\Omega & D_I D_\Omega & D_I \end{pmatrix} \quad (3-143)$$

Nous cherchons dans un premier temps à résoudre le problème suivant :

$$\begin{aligned} & \min_{D_o, D_I, (\Theta_I, \Theta_\Omega)} \gamma \\ & \text{sous contraintes de } \chi \text{ et} \\ & T(s) \in \mathbf{RH}_\infty. \end{aligned} \quad (3-144)$$

Les exigences χ sont définies en (3-85).

Comme pour l'approche par optimisation des pondérations fréquentielles, nous procédons à la résolution du problème avec plusieurs algorithmes et nous procédons à l'analyse statistique des résultats afin de faire ressortir l'algorithme le plus performant pour ce problème structuré. La logique de mesure de performance est identique à (2-89).

A titre informatif, le temps de calcul moyen est mesuré.

La totalité du test a été réalisée sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

Chaque test fait l'objet de 10 *runs* et chaque *run* fait l'objet de 20000 itérations.

<i>IT / run = 20000</i>	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>StaticCEC_QPSO</i>	-0,22	-0,07	0,10	78,09
<i>TVDecCEC_QPSO</i>	-0,19	-0,11	0,04	46,79
<i>Rnd_DE</i>	-0,21	-0,10	0,07	61,88
<i>Rnd2Bst_DE</i>	-0,23	-0,18	0,04	125,07
<i>Std_PSO</i>	-0,18	-0,06	0,07	143,05
<i>TVRndIW_PSO</i>	-0,18	-0,07	0,07	98,56
<i>ABC</i>	-0,15	-0,06	0,05	310,26
<i>CS</i>	-0,051	-0,01	0,02	22,62
<i>CS/DE</i>	-0,23	-0,13	0,07	565,37

Tableau 3.13 Comparatif d'algorithmes pour l'optimisation du correcteur structuré

On constate que le calcul évolutionnaire et particulièrement *Rnd2Bst_DE* conduit aux meilleurs résultats pour ce type de problème structuré.

Le meilleur des *runs* conduit au résultat de la Figure 3-37.

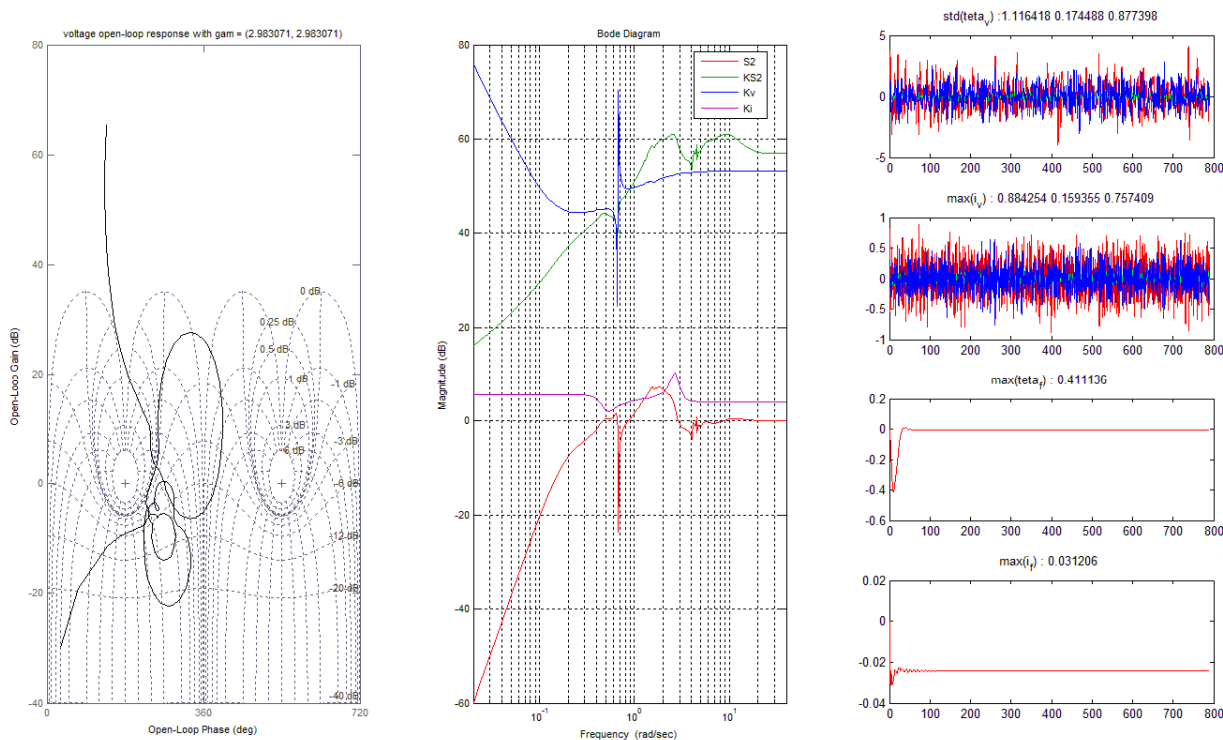


Figure 3-37: Régulation optimale du correcteur structuré

Les résultats quant à l'évaluation des exigences temporelles peuvent être rassemblés dans le Tableau 3.14. On rappelle que l'on présente pour des raisons de confidentialité les résultats relatifs aux valeurs maximales temporelles spécifiées (confidentielles), c'est-à-dire dire par rapport à 1.

<i>Exigences</i>	<i>Evaluations finales</i>
γ	2,98
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,41
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left i_f(t) \right \right) \leq i_{\max}$	0,03
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,11
$\max_{t>0} \left(\left i_{vx}(t) \right \right) \leq i_{\max}$	0,88
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,17
$\max_{t>0} \left(\left i_{vy}(t) \right \right) \leq i_{\max}$	0,16
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,88
$\max_{t>0} \left(\left i_{vz}(t) \right \right) \leq i_{\max}$	0,76

Tableau 3.14 Evaluation des exigences à l'issue de l'optimisation du correcteur structuré

On constate que la plupart des exigences sont satisfaites, seule la quatrième étant 11% au-dessus de la spécification ce qui reste acceptable.

3.5.2.3.2. Optimisation multi-modèles

La boule de modèles \wp employée a été définie au § 3.4.3.1.3.

On constate sur la Figure 3-38 qu'en l'état, le correcteur obtenu précédemment est peu robuste vis-à-vis de la boule de modèles. De fait la valeur pire de γ obtenu pour l'ensemble de la boule de modèles vaut 4,46. Aussi allons-nous chercher dans la suite à robustifier le correcteur par les techniques multi-modèles développées précédemment.

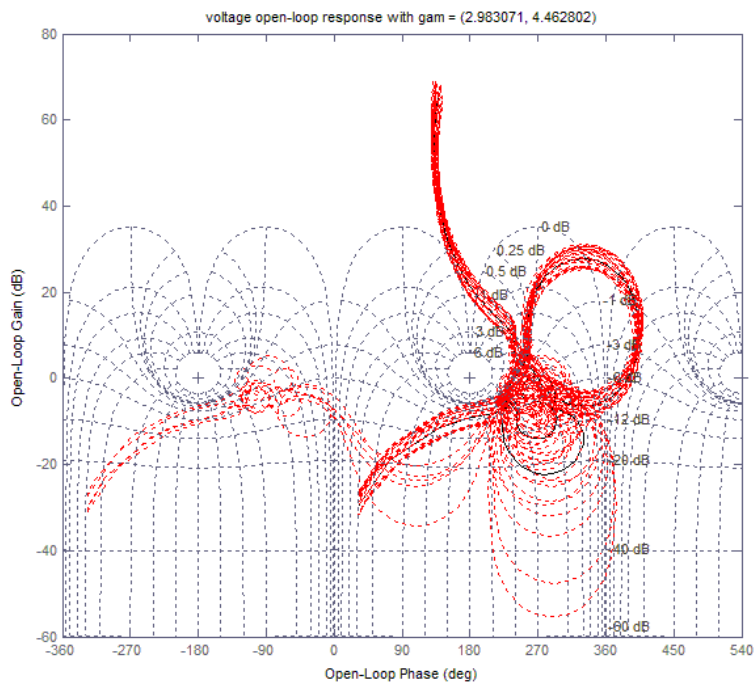


Figure 3-38: Examen de la robustesse du correcteur

On peut chercher (avec *Rnd_2_Bst*) un correcteur solution des problèmes suivants :

- problème de robustesse de la stabilité :

$$\begin{aligned}
 & \min_{D_o, D_i, K(s)} \gamma \\
 & \text{sous contraintes de } \chi \\
 & \text{et } K(s) \text{ stabilise tout } \varphi
 \end{aligned}
 \tag{3-145}$$

On obtient le résultat du Tableau 3.15.

<i>IT / run = 20000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,20	-0,13	0,05	181,39

Tableau 3.15 Optimisation du correcteur structuré – robustesse de la stabilité

On constate que les temps de calcul entre l’approche sur modèle nominal et le problème de robustesse en stabilité sont comparables.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-39.

L’évaluation finale des exigences est reportée sur le Tableau 3.4.

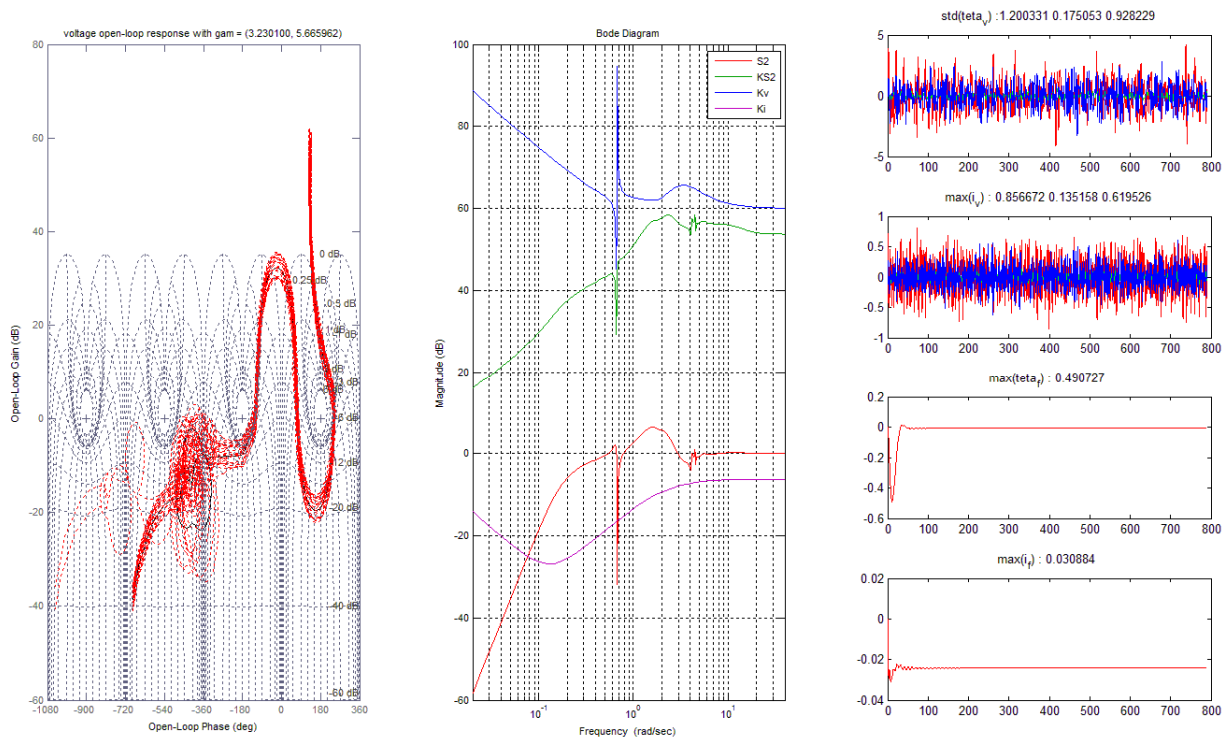


Figure 3-39: Problème de la robustesse de la stabilité

Exigences	Evaluations finales sur système nominal
$\gamma_{nom} ; \gamma_{pire}$	3,23 ; 5,66
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,49
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left i_f(t) \right \right) \leq i_{max}$	0,031
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,20
$\max_{t>0} \left(\left i_{vx}(t) \right \right) \leq i_{max}$	0,85
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,17
$\max_{t>0} \left(\left i_{vy}(t) \right \right) \leq i_{max}$	0,13
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,92
$\max_{t>0} \left(\left i_{vz}(t) \right \right) \leq i_{max}$	0,61

Tableau 3.16 Evaluation des exigences – robustesse de la stabilité

On constate que la plupart des exigences sont satisfaites (un peu dégradées toutefois, l'augmentation de la robustesse s'accompagne d'une diminution de performances).

- problème de robustesse des marges de stabilité :

$$\min_{D_o, D_i, K(s)} \gamma, \gamma = \max_{i=0, \dots, N} \gamma_i$$

sous contrainte de χ
 et $K(s)$ stabilise tout φ

(3-146)

On obtient le résultat du Tableau 3.17.

<i>IT / run = 20000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,16	-0,11	0,05	778,92

Tableau 3.17 Optimisation des pondérations – robustesse des marges de stabilité

On constate que le temps de calcul du problème de robustesse des marges de stabilité est bien plus grand que l'approche sur modèle nominal : ceci est dû à un nombre d'évaluations *N* fois plus grand de la norme H_∞ de la boucle fermée.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-40.

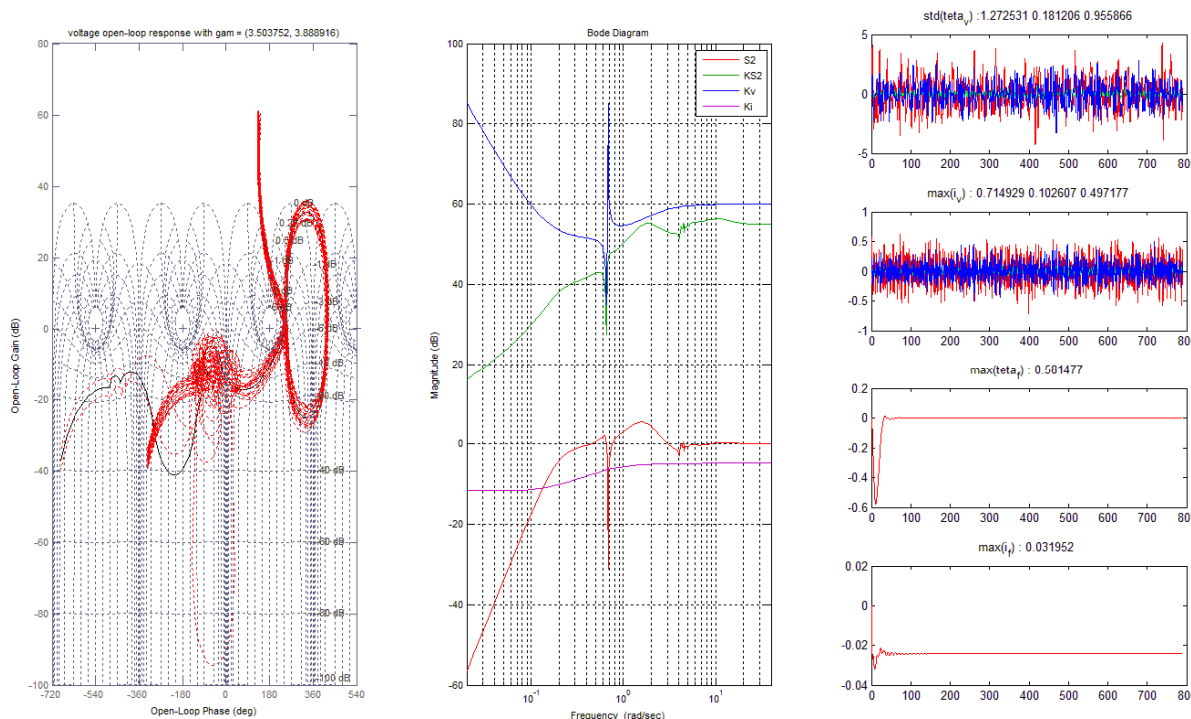


Figure 3-40: Problème de la robustesse des marges de stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.6.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	3,50 ; 3,89
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,58
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left i_f(t) \right \right) \leq i_{max}$	0,032
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	1,27
$\max_{t>0} \left(\left i_{vx}(t) \right \right) \leq i_{max}$	0,71
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,18
$\max_{t>0} \left(\left i_{vy}(t) \right \right) \leq i_{max}$	0,10
$\sigma_{\theta_{vz}} \leq \bar{\theta}_v$	0,96
$\max_{t>0} \left(\left i_{vz}(t) \right \right) \leq i_{max}$	0,50

Tableau 3.18 Evaluation des exigences – robustesse des marges de stabilité

On constate que la plupart des exigences sont satisfaites et que la valeur de γ pire est à présent de 3,89.

3.5.2.3.3. Le viseur de tir

On reprend l'exemple du viseur de tir et on cherche cette fois ci directement le correcteur de la Figure 3-41.

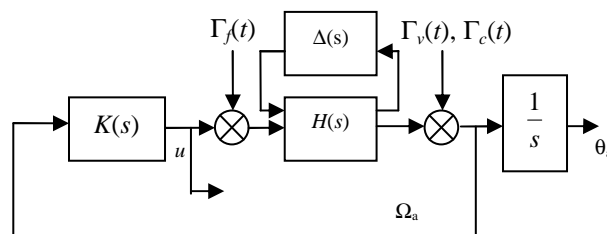


Figure 3-41: Schéma de commande du viseur

On choisit d'optimiser le correcteur à l'ordre 12.

Nous cherchons dans un premier temps à résoudre le problème suivant :

$$\min_{D_o, D_i, K(s)} \gamma$$

sous contraintes de χ' et

$$T(s) \in \mathbf{RH}_\infty \tag{3-147}$$

Les exigences χ' sont définies en (3-98).

De par son efficacité montrée précédemment, on choisit d'approcher le problème avec l'algorithme *Rnd2Bst_DE*.

La totalité du test a été réalisée sur une machine équipée d'un processeur Intel Core i7-3740QM (2.7 GHz) et de 8 Gb de RAM.

Chaque *run* fait l'objet de 10000 itérations.

On obtient le résultat du Tableau 3.19.

<i>IT / run = 10000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,21	-0,19	0,019	90,58

Tableau 3.19 Optimisation du correcteur du viseur de tir

Qui conduit à la Figure 3-42.

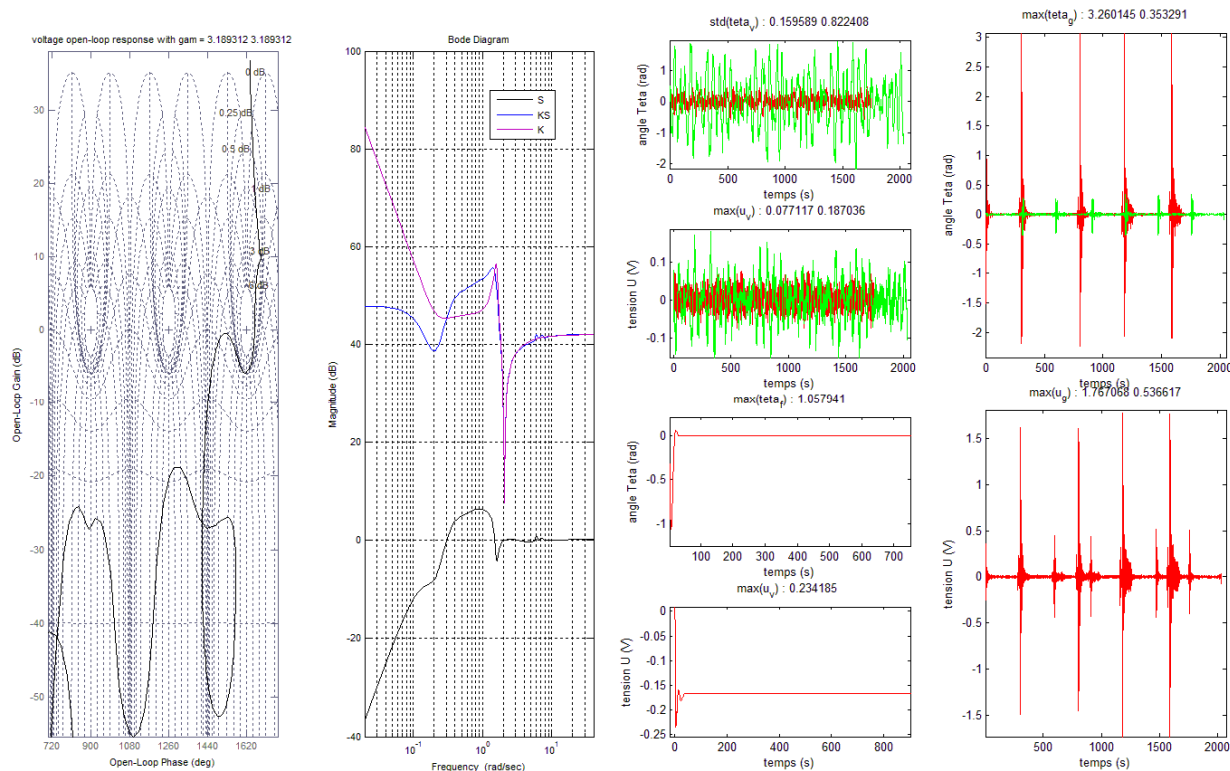


Figure 3-42: Optimisation du correcteur

Les résultats quant à l'évaluation finale des exigences temporelles peuvent être rassemblés dans le Tableau 3.20. On rappelle que l'on présente pour des raisons de

confidentialité les résultats relatifs aux valeurs maximales temporelles spécifiées (confidentielles), c'est-à-dire dire par rapport à 1.

<i>Exigences</i>	<i>Evaluations finales</i>
γ	3,19
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	1,06
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{\max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,16
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{\max}$	0,07
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,82
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{\max}$	0,19
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{\max}$	1,76
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{\max}$	0,54

Tableau 3.20 Evaluation des exigences à l'issue de l'optimisation du correcteur

On constate que les exigences de priorité haute sont quasiment toutes satisfaites. Une exigence de priorité basse n'est pas satisfaite mais elle reste néanmoins convenable : l'algorithme a cherché à optimiser au maximum les exigences de priorité faible une fois les exigences de priorité haute satisfaites (c'est ce qui explique le comportement de rejection du correcteur vers 2 rad/s, fréquence du mode propre du viseur sollicitée lors des chocs). En l'absence des exigences relatives à la tenue des chocs, cette rejection n'aurait pas été calculée.

3.5.2.3.4. Optimisation multi-modèles

La boule de modèles \wp employée a été définie au § 3.4.3.2.2.

On constate sur la Figure 3-43 qu'en l'état, le correcteur obtenu précédemment est peu robuste vis-à-vis de la boule de modèles. En effet, la valeur pire-cas de γ obtenu pour l'ensemble de la boule de modèles vaut 6,42. Aussi allons-nous chercher dans la suite à robustifier le correcteur par les techniques multi-modèles développées précédemment.

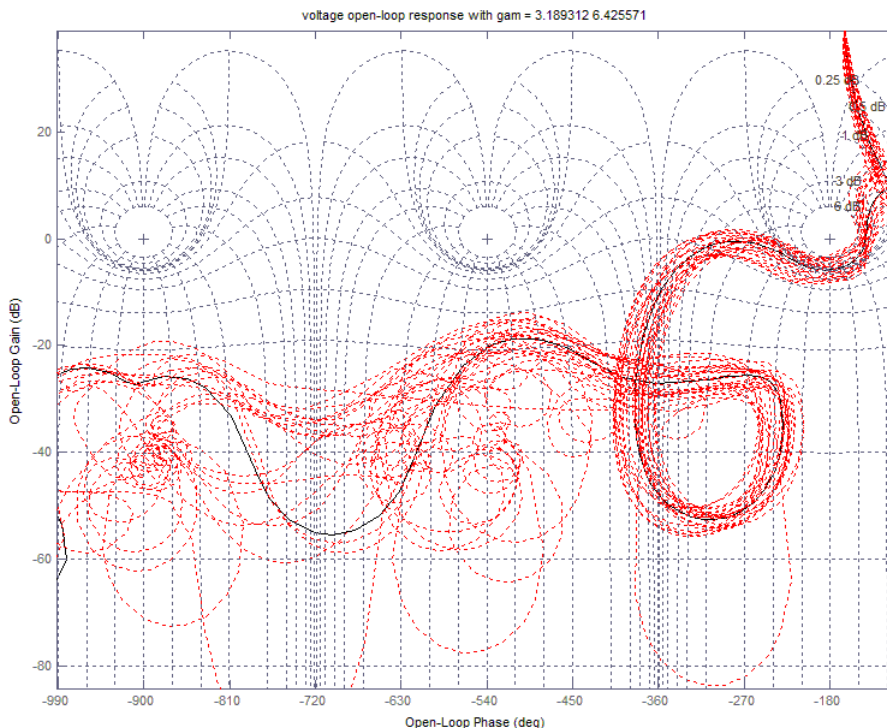


Figure 3-43: Examen de la robustesse du correcteur

On peut chercher (avec *Rnd_2_Bst*) un correcteur solution des problèmes suivants :

- problème de robustesse de la stabilité :

$$\min_{D_o, D_i, K(s)} \gamma$$

sous contraintes de χ' et $K(s)$ stabilise tout ϕ (3-148)

On obtient le résultat du Tableau 3.21.

<i>IT</i> / run = 10000	f_{min}	f_{mean}	f_{std}	$t_{cpu/run}$ (mn)
<i>Rnd2Bst_DE</i>	-0,20	-0,17	0,03	93,96

Tableau 3.21 Optimisation du correcteur – robustesse de la stabilité

On constate que les temps de calcul entre l’approche sur modèle nominal et le problème de robustesse en stabilité sont comparables.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-44.

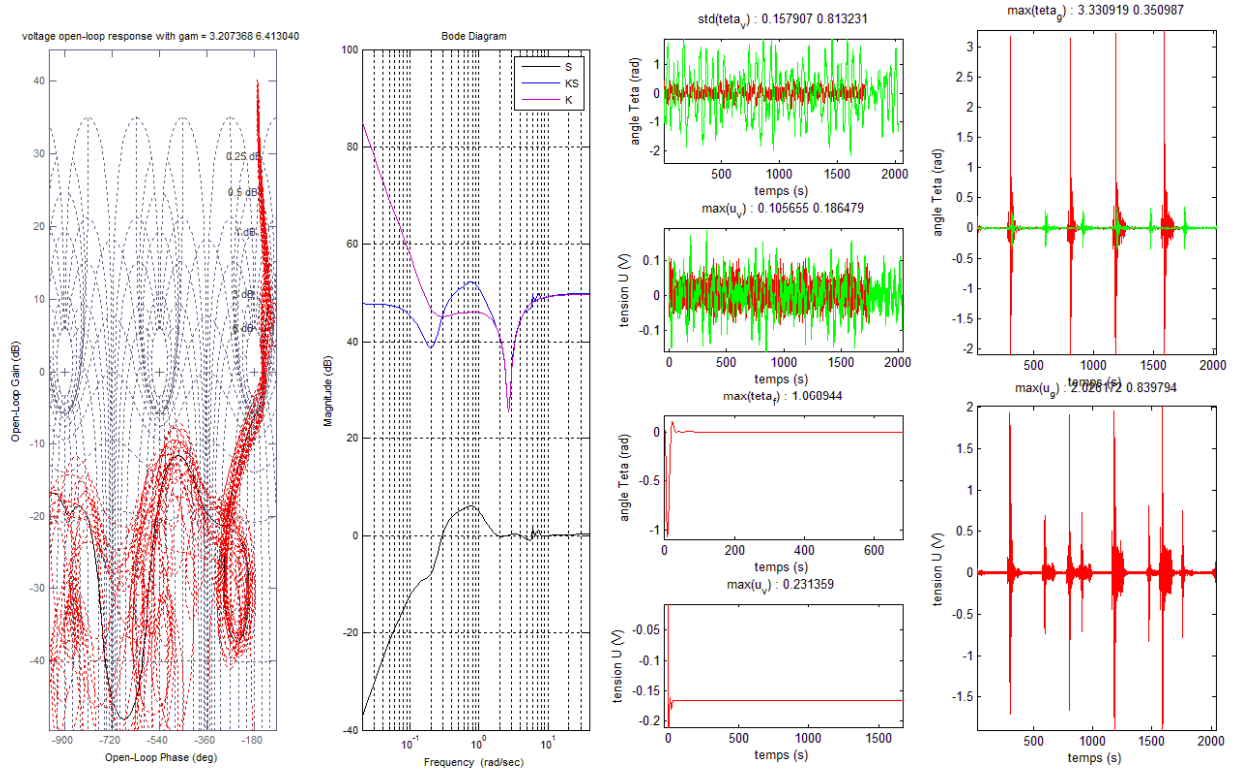


Figure 3-44: Problème de la robustesse de la stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.22.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	3,20 ; 6,41
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	1,07
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{\max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,16
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{\max}$	0,11
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,81
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{\max}$	0,19
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{\max}$	2,02
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{\max}$	0,84

Tableau 3.22 Evaluation des exigences – robustesse de la stabilité

On constate que la plupart des exigences sont satisfaites (un peu dégradés toute fois, l'augmentation de la robustesse s'accompagne d'une diminution de performances).

- problème de robustesse des marges de stabilité :

$$\min_{D_o, D_i, K(s)} \gamma, \gamma = \max_{j=0, \dots, N} \gamma_j$$

sous contrainte de χ' (3-149)
et $K(s)$ stabilise tout φ

On obtient le résultat du Tableau 3.23.

<i>IT / run = 10000</i>	<i>f_{min}</i>	<i>f_{mean}</i>	<i>f_{std}</i>	<i>t_{cpu/run} (mn)</i>
<i>Rnd2Bst_DE</i>	-0,12	-0,083	0,045	224,82

Tableau 3.23 Optimisation du correcteur – robustesse des marges de stabilité

On constate que le temps de calcul du problème de robustesse des marges de stabilité est bien plus grand que l'approche sur modèle nominal : ceci est dû à un nombre d'évaluations N fois plus grand de la norme H_∞ de la boucle fermée.

Le meilleur des 10 *runs* conduit aux résultats de la Figure 3-45.

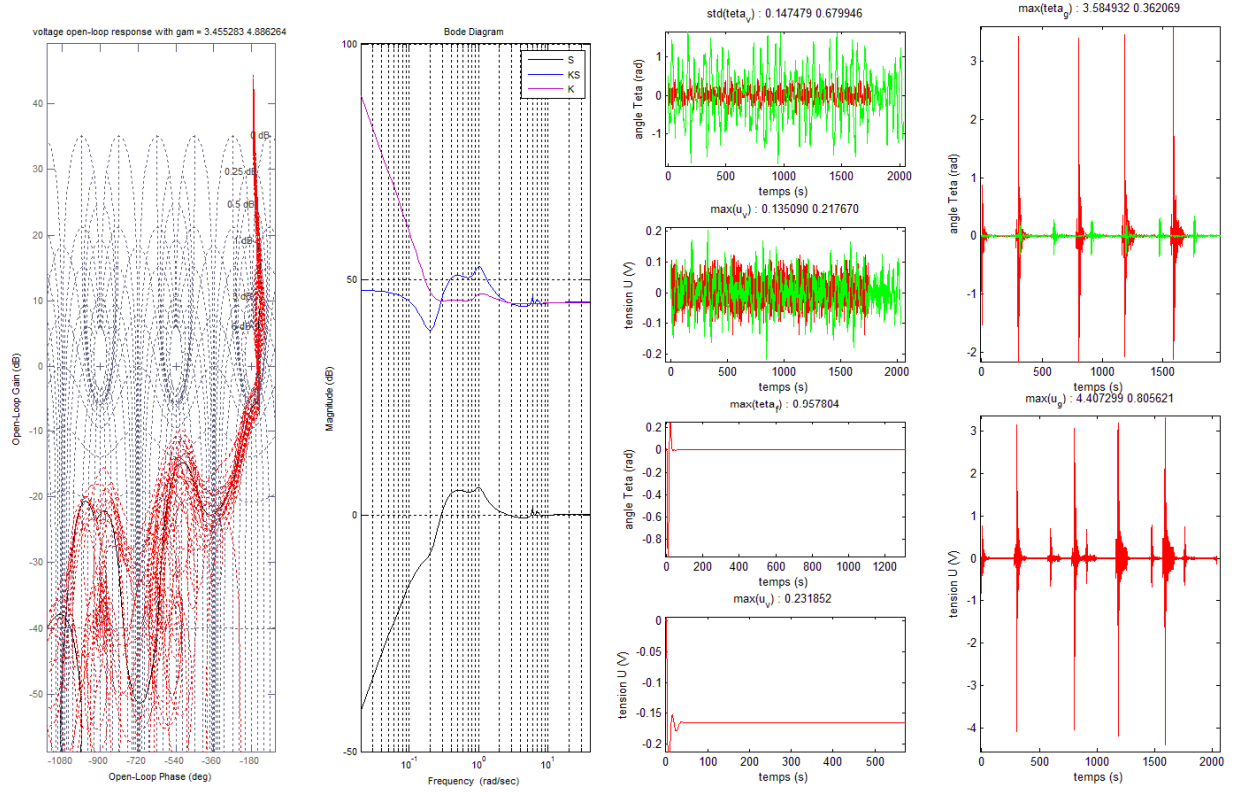


Figure 3-45: Problème de la robustesse des marges de stabilité

L'évaluation finale des exigences est reportée sur le Tableau 3.24.

<i>Exigences</i>	<i>Evaluations finales sur système nominal</i>
$\gamma_{nom} ; \gamma_{pire}$	3,45 ; 4,88
$\max_{t>0} \left(\left \theta_f(t) \right \right) \leq \bar{\theta}_f$	0,95
$\max_{t>T_f} \left(\left \theta_f(t) \right \right) \leq \varepsilon_f$	0
$\max_{t>0} \left(\left u_f(t) \right \right) \leq u_{\max}$	0,23
$\sigma_{\theta_{vx}} \leq \bar{\theta}_v$	0,15
$\max_{t>0} \left(\left u_{vx}(t) \right \right) \leq u_{\max}$	0,14
$\sigma_{\theta_{vy}} \leq \bar{\theta}_v$	0,68
$\max_{t>0} \left(\left u_{vy}(t) \right \right) \leq u_{\max}$	0,22
$\max_{t>0} \left(\left u_{cx}(t) \right \right) \leq u_{\max}$	4,41
$\max_{t>0} \left(\left u_{cy}(t) \right \right) \leq u_{\max}$	0,81

Tableau 3.24 Evaluation des exigences – robustesse des marges de stabilité

On constate que la valeur de γ pire-cas est à présent de 4,88. De plus les exigences à priorité faible sont moins satisfaites. On peut imaginer qu'elles le deviennent en augmentant l'ordre du correcteur cherché ou en augmentant le nombre d'itérations. Les exigences à forte priorité sont satisfaites.

3.5.3. Extension aux correcteurs à 2 degrés de liberté

3.5.3.1. Approche en une étape

3.5.3.1.1. Suivi de référence

Considérons le schéma de la Figure 3-46. On y retrouve le schéma 4-blocs équivalent de la synthèse H_∞ par loop-shaping auquel a été rajoutée une composante d'anticipation $K_{p1}(s)$ permettant le modelage d'un transfert de boucle (ici $S_p H_p$) vis-à-vis d'un modèle de référence $T_m(s)$ donné lorsque le transfert reliant ε_m à r_m est contraint. Le correcteur de boucle est noté $K_{p2}(s)$.

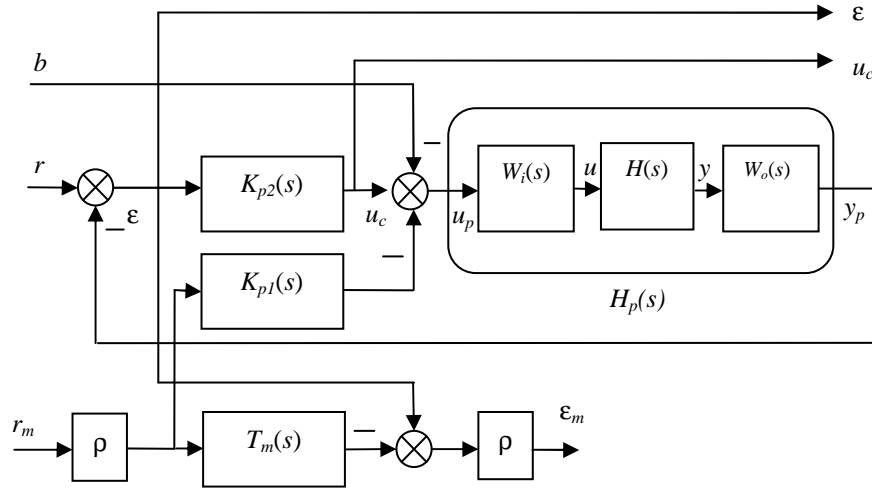


Figure 3-46: Correction à 2 degrés de liberté dans le cadre loop-shaping

On montre facilement que :

$$\begin{aligned}
 \varepsilon(s) &= S_p(s)r(s) + S_p(s)H_p(s)b(s) + S_p(s)H_p(s)K_{p1}(s)\rho r_m(s) \\
 u_c(s) &= K_{p2}(s)S_p(s)r(s) + K_{p2}(s)S_p(s)H_p(s)b(s) + \\
 &\quad K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho r_m(s) \\
 \varepsilon_m(s) &= \rho S_p(s)r(s) + \rho S_p(s)H_p(s)b(s) + \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho r_m(s) \\
 S_p(s) &= (I + H_p(s)K_{p2}(s))^{-1}
 \end{aligned} \tag{3-150}$$

C'est-à-dire sous forme matricielle :

$$\begin{pmatrix} \varepsilon(s) \\ u_c(s) \\ \varepsilon_m(s) \end{pmatrix} = \begin{pmatrix} S_p(s) & S_p(s)H_p(s) & S_p(s)H_p(s)K_{p1}(s)\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho \\ \rho S_p(s) & \rho S_p(s)H_p(s) & \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho \end{pmatrix} \begin{pmatrix} r(s) \\ b(s) \\ r_m(s) \end{pmatrix} \tag{3-151}$$

On peut alors chercher les correcteurs $K_{p2}(s)$ et $K_{p1}(s)$ qui stabilisent le transfert précédent et qui minimisent γ tel que :

$$\left\| \begin{pmatrix} S_p(s) & S_p(s)H_p(s) & S_p(s)H_p(s)K_{p1}(s)\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho \\ \rho S_p(s) & \rho S_p(s)H_p(s) & \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho \end{pmatrix} \right\|_{\infty} < \gamma \tag{3-152}$$

Lorsque $\rho=0$, on retrouve strictement la formulation de la synthèse H_{∞} par loop-shaping pour la détermination de $K_{p2}(s)$. Le paramètre ρ est ainsi un paramètre réglage permettant de découpler le problème du suivi de consigne par $K_{p1}(s)$ de celui de la

stabilisation du loop-shape par $K_{p2}(s)$. Le transfert $S_p(s)H_p(s)K_{p1}(s)$ sera donc modélisé par le modèle de référence $T_m(s)$ et ce d'autant plus que ρ est grand. Les autres transferts du critère dépendant de ρ permettent de limiter l'impact sur la détermination du correcteur de boucle $K_{p2}(s)$ (que l'on veut comme étant la solution au problème H_∞ loop-shaping standard).

On retrouvera un principe similaire employant le formalisme des factorisations premières dans [LIM 93, HOY 91, SKO 01].

Les précompensateurs K_{p1} et K_{p2} étant déterminés, les correcteurs finaux sont alors reconstruits par les relations (3-153).

$$\begin{aligned} K_1(s) &= W_i(s)K_{p1}(s)W_o(s) \\ K_2(s) &= W_i(s)K_{p2}(s)W_o(s) \end{aligned} \quad (3-153)$$

Justifions (3-153) en considérant la Figure 3-47.

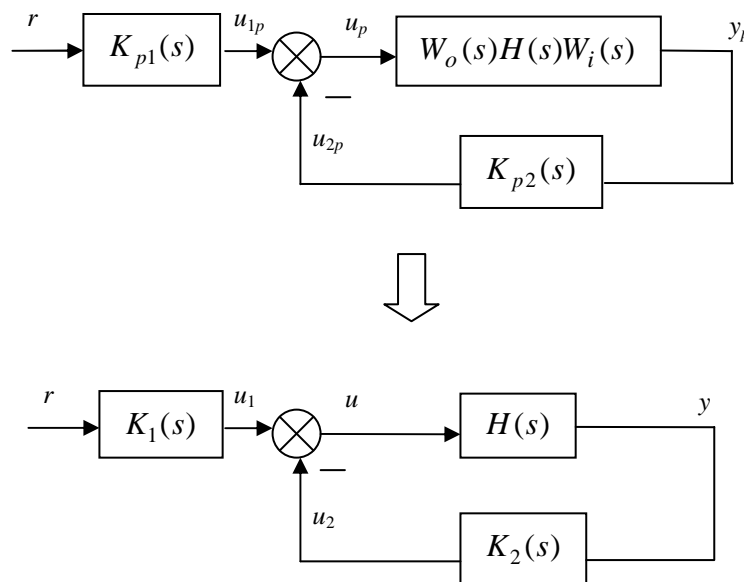


Figure 3-47: Reconstruction finale du correcteur à 2 degrés de liberté

On montre facilement que :

$$\begin{aligned} y(s) &= (I + H(s)K_2(s))^{-1} H(s)K_1(s)r(s) \\ &= S(s)H(s)K_1(s)r(s) \\ &= T_{r \rightarrow y}(s)r(s) \end{aligned} \quad (3-154)$$

D'autre part :

$$\begin{aligned}
 y_p(s) &= \left(I + H_p(s)K_{p2}(s) \right)^{-1} H_p(s)K_{p1}(s)r(s) \\
 &= \left(I + W_o(s)H(s)W_i(s)K_{p2}(s) \right)^{-1} W_o(s)H(s)W_i(s)K_{p1}(s)r(s)
 \end{aligned} \tag{3-155}$$

En supposant les pondérations inversibles, alors :

$$K_{p2}(s) = W_i(s)^{-1} K_2(s)W_o(s)^{-1} \tag{3-156}$$

Il vient donc en les supposant de plus diagonales :

$$\begin{aligned}
 y_p(s) &= \left(I + W_o(s)H(s)K_2(s)W_o(s)^{-1} \right)^{-1} W_o(s)H(s)W_i(s)K_{p1}(s)r(s) \\
 &= W_o(s) \left(I + H(s)K_2(s) \right)^{-1} H(s)W_i(s)K_{p1}(s)r(s) \\
 &= W_o(s)S(s)H(s)W_i(s)K_{p1}(s)r(s) \\
 &= T_{r \rightarrow y_p}(s)r(s)
 \end{aligned} \tag{3-157}$$

Ainsi :

$$\begin{aligned}
 \sigma(T_{r \rightarrow y_p}(s)) &= \sigma(W_o(s)S(s)H(s)W_i(s)K_{p1}(s)) \\
 &= \sigma(S(s)H(s)W_i(s)K_{p1}(s)W_o(s))
 \end{aligned} \tag{3-158}$$

Ainsi en écrivant :

$$K_1(s) = W_i(s)K_{p1}(s)W_o(s) \tag{3-159}$$

Alors :

$$\begin{aligned}
 \sigma(T_{r \rightarrow y_p}(s)) &= \sigma(S(s)H(s)K_1(s)) \\
 &= \sigma(T_{r \rightarrow y}(s))
 \end{aligned} \tag{3-160}$$

Ce qui justifie la reconstruction (3-153).

A noter que lorsqu'on ne souhaite pas imposer un modèle de référence à toutes les mesures mais à quelques composantes seulement, alors on introduira une matrice de sélection de canal lors de la construction de ε_m .

Reprenant notre problématique à m exigences, on peut alors chercher les pondérations et le paramètre ρ solutions du problème (3-161), simple extension de l'approche à 1 degré de liberté.

$$\begin{aligned}
& \min_{\rho, W_o(s), W_i(s)} \gamma \\
& \text{sous contraintes que :} \\
& W_o(s), W_i(s) \in \mathbf{RH}_\infty \\
& Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
\end{aligned} \tag{3-161}$$

Avec γ défini en (3-152). La formulation de ce problème est identique à celui à 1 degré de liberté. On calcule les correcteurs K_{p1} et K_{p2} H_∞ -optimaux, solution du problème (3-161), en considérant le problème d'optimisation suivant :

$$\begin{aligned}
& \min_{\Theta_i, \Theta_o, \rho} f(\Theta_i, \Theta_o, \rho) \\
& \text{si } \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i) \geq 0 \\
& f(\Theta_i, \Theta_o, \rho) = \max(\bar{\lambda}_o, \underline{\lambda}_o, \bar{\lambda}_i, \underline{\lambda}_i) \\
& \text{sinon} \\
& H_p(s) = W_o(s)H(s)W_i(s) \\
& K(s) = [K_{p1}(s) \ K_{p2}(s)] = \arg \min \gamma, \\
& \left\| \begin{array}{cc|c} S_p(s) & S_p(s)H_p(s) & S_p(s)H_p(s)K_{p1}(s)\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho \\ \hline \rho S_p(s) & \rho S_p(s)H_p(s) & \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho \end{array} \right\|_\infty < \gamma \tag{3-162} \\
& K_1(s) = W_i(s)K_{p1}(s)W_o(s), \ K_2(s) = W_i(s)K_{p2}(s)W_o(s) \\
& f(\Theta_i, \Theta_o, \rho) = -\frac{1}{\gamma + f_c(H, K)} \\
& \text{fin si}
\end{aligned}$$

De même que nous l'avons fait pour l'approche à 1 degré de liberté, rien ne nous empêche de chercher les pondérations et les correcteurs finaux simultanément. En supposant les pondérations du loop-shape inversibles alors :

$$\left. \begin{aligned}
H_p(s) &= W_o(s)H(s)W_i(s) \\
K_{p2}(s) &= W_i(s)^{-1}K_2(s)W_o(s)^{-1} \\
K_{p1}(s) &= W_i(s)^{-1}K_1(s)W_o(s)^{-1}
\end{aligned} \right\} \Rightarrow S_p(s) = (I + W_o(s)H(s)K_2(s)W_o(s)^{-1})^{-1} \tag{3-163}$$

Notons que :

$$\begin{aligned}
& \left\| \begin{array}{cc|c} S_p(s) & S_p(s)H_p(s) & S_p(s)H_p(s)K_{p1}(s)\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho \\ \hline \rho S_p(s) & \rho S_p(s)H_p(s) & \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho \end{array} \right\|_\infty \tag{3-164} \\
& = \left\| \begin{pmatrix} I \\ K_{p2}(s) \\ \rho I \end{pmatrix} S_p(s) \begin{pmatrix} I & H_p(s) & H_p(s)K_{p1}(s)\rho \end{pmatrix} - \rho \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & T_m(s) \end{pmatrix} \right\|_\infty
\end{aligned}$$

Si les pondérations sont par ailleurs diagonales :

$$\begin{aligned}
 & \left(\begin{array}{c} I \\ K_{p2}(s) \\ \rho I \end{array} \right) S_p(s) \left(I \quad H_p(s) \mid H_p(s)K_{p1}(s)\rho \right) \\
 &= \left(\begin{array}{c} I \\ W_i(s)^{-1}K_2(s)W_o(s)^{-1} \\ \rho I \end{array} \right) S_p(s) \left(I \quad W_o(s)H(s)W_i(s) \mid W_o(s)H(s)K_1(s)W_o(s)^{-1}\rho \right) \\
 &= \left(\begin{array}{c} W_o(s) \\ W_i(s)^{-1}K_2(s) \\ \rho W_o(s) \end{array} \right) W_o(s)^{-1} S_p(s) W_o(s) \left(W_o(s)^{-1} \quad H(s)W_i(s) \mid H(s)K_1(s)W_o(s)^{-1}\rho \right) \quad (3-165) \\
 &= \left(\begin{array}{c} W_o(s) \\ W_i(s)^{-1}K_2(s) \\ \rho W_o(s) \end{array} \right) (I + H(s)K_2(s))^{-1} \left(W_o(s)^{-1} \quad H(s)W_i(s) \mid H(s)K_1(s)W_o(s)^{-1}\rho \right) \\
 &= \left(\begin{array}{c} W_o(s) \\ W_i(s)^{-1}K_2(s) \\ \rho W_o(s) \end{array} \right) S(s) \left(W_o(s)^{-1} \quad H(s)W_i(s) \mid H(s)K_1(s)W_o(s)^{-1}\rho \right)
 \end{aligned}$$

Ainsi :

$$\begin{aligned}
 & \left(\begin{array}{cc|c} S_p(s) & S_p(s)H_p(s) & S_p(s)H_p(s)K_{p1}(s)\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)H_p(s)K_{p1}(s)\rho \\ \rho S_p(s) & \rho S_p(s)H_p(s) & \rho(S_p(s)H_p(s)K_{p1}(s) - T_m(s))\rho \end{array} \right) \\
 &= \left(\begin{array}{c} \left(\begin{array}{c} W_o(s) \\ W_i(s)^{-1}K_2(s) \\ \rho W_o(s) \end{array} \right) S(s) \left(W_o(s)^{-1} \quad H(s)W_i(s) \mid H(s)K_1(s)W_o(s)^{-1}\rho \right) \\ -\rho \left(\begin{array}{c|c} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right) \rho \\ \left(\begin{array}{c} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right) T_m(s) \end{array} \right) \quad (3-166) \\
 &= T(s)
 \end{aligned}$$

Noter que le critère H_∞ (3-166) correspond au schéma de la Figure 3-48.

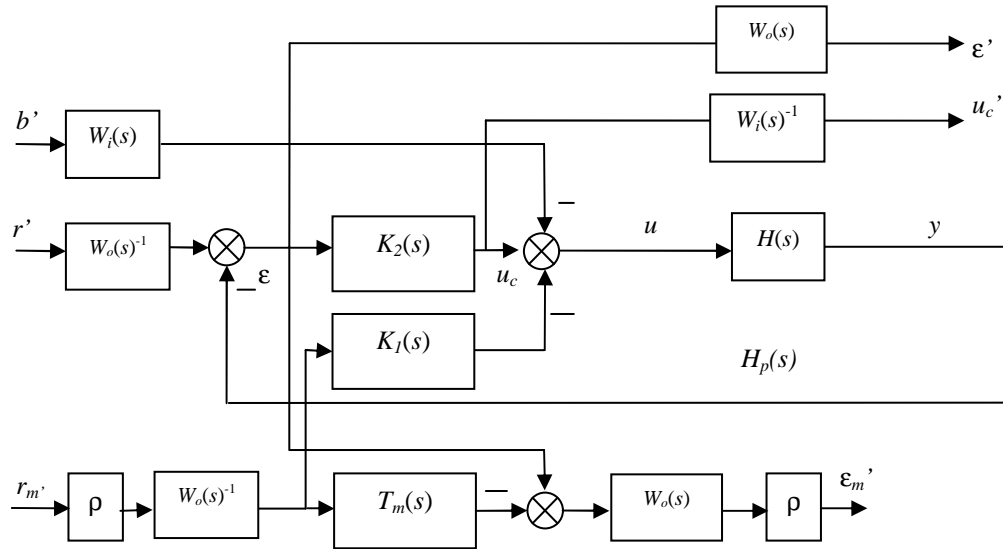


Figure 3-48: Correction à 2 degrés de liberté – pondérations externalisées

On peut alors chercher directement les correcteurs finaux structurés, les pondérations et le paramètre ρ , solutions du problème (3-167) :

$$\begin{aligned}
 & \min_{\rho, W_o(s), W_i(s), K_1(s), K_2(s)} \gamma \\
 & \text{sous contraintes que :} \\
 & W_o(s), W_i(s) \in \mathbf{RH}_\infty \\
 & T(s) \in \mathbf{RH}_\infty, \\
 & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m.
 \end{aligned} \tag{3-167}$$

Avec $\gamma = \|T(s)\|_\infty$.

De même que nous l'avons constaté pour l'approche à 1 degré de liberté, la recherche simultanée des pondérations et des correcteurs finaux conduit à des pondérations statiques si bien qu'on peut remplacer le problème (3-167) par le problème (3-168) avec la mise en œuvre de scalings statiques à la place des pondérations fréquentielles :

$$\begin{aligned}
 & \min_{\rho, D_o, D_i, K_1(s), K_2(s)} \gamma \\
 & \text{sous contrainte que :} \\
 & T(s) \in \mathbf{RH}_\infty \\
 & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
 \end{aligned} \tag{3-168}$$

Avec :

$$\begin{aligned}
 T(s) &= \begin{pmatrix} A_{cl} & B_{cl} \\ C_{cl} & D_{cl} \end{pmatrix} \\
 &= \left(\begin{pmatrix} D_o \\ D_i^{-1} K_2(s) \\ \rho D_o \end{pmatrix} S(s) \left(D_o^{-1} \quad H(s) D_i \quad \Big| \quad H(s) K_1(s) D_o^{-1} \rho \right) - \rho \begin{pmatrix} 0 & 0 & \Big| & 0 \\ 0 & 0 & \Big| & 0 \\ 0 & 0 & \Big| & T_m(s) \end{pmatrix} \rho \right) \quad (3-169) \\
 S &= (I + HK_2)^{-1}
 \end{aligned}$$

En posant :

$$\bar{\lambda}_{cl} = \max_j (\operatorname{Re}(\lambda_j(A_{cl}))) \quad (3-170)$$

Alors on résout le problème (3-168) en considérant le problème d'optimisation suivant :

$$\begin{aligned}
 &\min_{\rho, D_i, D_o, \Theta_{k1}, \Theta_{k2}} f(\rho, D_i, D_o, \Theta_{k1}, \Theta_{k2}) \\
 &\text{si } \bar{\lambda}_{cl} \geq 0 \\
 &\quad f(\rho, D_i, D_o, \Theta_{k1}, \Theta_{k2}) = \bar{\lambda}_{cl} \\
 &\text{sinon} \\
 &\quad \gamma = \|T(s)\|_{\infty} \\
 &\quad f(\rho, D_i, D_o, \Theta_{k1}, \Theta_{k2}) = -\frac{1}{\gamma + f_c(H, K)} \\
 &\text{fin si}
 \end{aligned} \quad (3-171)$$

On simplifie alors grandement la résolution du problème

3.5.3.1.2. Rejet de perturbations

On peut appliquer le principe précédent pour le calcul simultané d'un correcteur de boucle (structuré) et d'un anticipateur (structuré) vis-à-vis d'un problème de rejet de perturbations en sortie de modèle comme sur le schéma de la Figure 3-49.

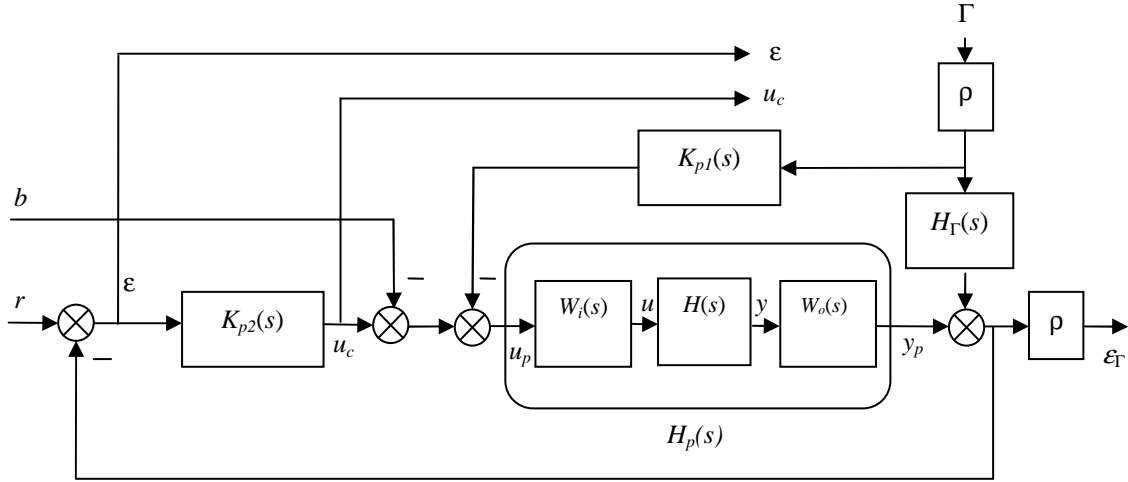


Figure 3-49: Correction à 2 degrés de liberté – rejet de perturbations

On montre que :

$$\begin{aligned}
 \varepsilon(s) &= S_p(s)r(s) + S_p(s)H_p(s)b(s) + S_p(s)(-H_\Gamma(s) + H_p(s)K_{p1}(s))\rho\Gamma(s) \\
 u_c(s) &= K_{p2}(s)S_p(s)r(s) + K_{p2}(s)S_p(s)H_p(s)b(s) + \\
 &\quad K_{p2}(s)S_p(s)(-H_\Gamma(s) + H_p(s)K_{p1}(s))\rho\Gamma(s) \\
 \varepsilon_\Gamma(s) &= \rho S_p(s)r(s) - \rho S_p(s)H_p(s)b(s) + \rho S_p(s)(H_\Gamma(s) - H_p(s)K_{p1}(s))\rho\Gamma(s) \\
 S_p(s) &= (I + H_p(s)K_{p2}(s))^{-1}
 \end{aligned} \tag{3-172}$$

C'est-à-dire sous forme matricielle :

$$\begin{pmatrix} \varepsilon(s) \\ u_c(s) \\ \varepsilon_\Gamma(s) \end{pmatrix} = \begin{pmatrix} S_p(s) & S_p(s)H_p(s) & S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \\ \rho S_p(s) & -\rho S_p(s)H_p(s) & -\rho S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \end{pmatrix} \begin{pmatrix} r(s) \\ b(s) \\ \Gamma(s) \end{pmatrix} \tag{3-173}$$

On peut alors chercher les correcteurs $K_{p2}(s)$ et $K_{p1}(s)$ qui stabilisent le transfert précédent et qui minimisent γ tel que :

$$\left\| \begin{pmatrix} S_p(s) & S_p(s)H_p(s) & S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \\ \rho S_p(s) & -\rho S_p(s)H_p(s) & -\rho S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\rho \end{pmatrix} \right\|_\infty < \gamma \tag{3-174}$$

Afin de déterminer directement les correcteurs finaux, notons que :

$$\begin{aligned}
 & \left(\begin{array}{cc|c} S_p(s) & S_p(s)H_p(s) & S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\mathfrak{p} \\ K_{p2}(s)S_p(s) & K_{p2}(s)S_p(s)H_p(s) & K_{p2}(s)S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\mathfrak{p} \\ \hline \rho S_p(s) & \rho S_p(s)H_p(s) & \rho S_p(s)(H_p(s)K_{p1}(s) - H_\Gamma(s))\mathfrak{p} \end{array} \right) \\
 & = \left(\begin{array}{c} I \\ \hline \frac{K_{p2}(s)}{\rho I} \end{array} \right) S_p(s) \left(\begin{array}{c|c} I & H_p(s) \\ \hline H_p(s)K_{p1}(s) & H_p(s)K_{p1}(s) - H_\Gamma(s) \end{array} \right) \mathfrak{p}
 \end{aligned} \quad (3-175)$$

Par un calcul similaire à (3-165), on montre facilement que :

$$\begin{aligned}
 & \left(\begin{array}{c} I \\ \hline \frac{K_{p2}(s)}{\rho I} \end{array} \right) S_p(s) \left(\begin{array}{c|c} I & H_p(s) \\ \hline H_p(s)K_{p1}(s) & H_p(s)K_{p1}(s) - H_\Gamma(s) \end{array} \right) \mathfrak{p} \\
 & = \left(\begin{array}{c} W_o(s) \\ \hline \frac{W_i(s)^{-1}K_2(s)}{\rho W_o(s)} \end{array} \right) S(s) \left(\begin{array}{c|c} W_o(s)^{-1} & H(s)W_i(s) \\ \hline H(s)K_1(s)W_o(s)^{-1} & H(s)K_1(s)W_o(s)^{-1} - W_o(s)^{-1}H_\Gamma(s) \end{array} \right) \mathfrak{p}
 \end{aligned} \quad (3-176)$$

$$S = (I + HK_2)^{-1}$$

Ainsi en remplaçant les pondérations fréquentielles par des scalings statiques, on peut alors résoudre le problème (3-177).

$$\begin{aligned}
 & \min_{\rho, D_o, D_i, K_1(s), K_2(s)} \gamma \\
 & \text{sous contrainte que :} \\
 & T(s) \in \mathbf{RH}_\infty \\
 & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
 \end{aligned} \quad (3-177)$$

Avec :

$$\begin{aligned}
 T(s) & = \begin{pmatrix} A_{cl} & B_{cl} \\ C_{cl} & D_{cl} \end{pmatrix} \\
 & = \left(\begin{array}{c} D_o \\ \hline \frac{D_i^{-1}K_2(s)}{\rho D_o} \end{array} \right) S(s) \left(\begin{array}{c|c} D_o^{-1} & H(s)D_i \\ \hline H(s)K_1(s)D_o^{-1} & H(s)K_1(s)D_o^{-1} - D_o^{-1}H_\Gamma(s) \end{array} \right) \mathfrak{p}
 \end{aligned} \quad (3-178)$$

$$S = (I + HK_2)^{-1}$$

La fonction à minimiser est identique à (3-171).

3.5.3.2. Approche en n étapes

Rien n'empêche d'étendre l'approche structurée précédente à la schématique générale de compensation par anticipation d'une perturbation Γ comme sur la Figure 3-50.

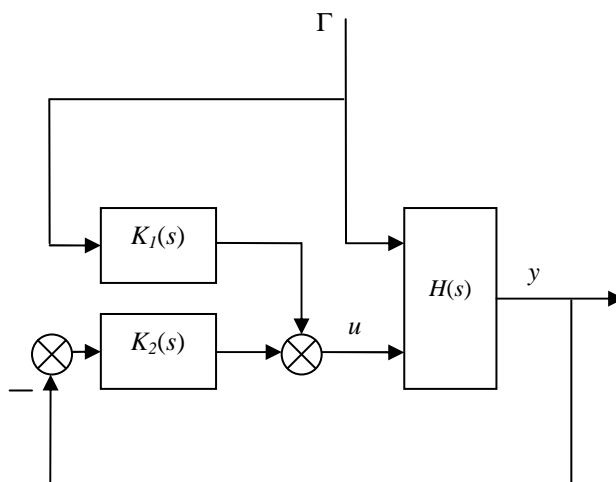


Figure 3-50: Correction à 2 degrés de liberté, formalisme générale

Dans ce cas la loi de commande a la forme (3-179).

$$u = K_1(s)\Gamma - K_2(s)y = \begin{bmatrix} K_1(s) & -K_2(s) \end{bmatrix} \begin{bmatrix} \Gamma \\ y \end{bmatrix} = K(s) \begin{bmatrix} \Gamma \\ y \end{bmatrix} \quad (3-179)$$

Dans la suite $K_1(s)$ désigne l'anticipateur et $K_2(s)$ désigne le correcteur de boucle. Le système $H(s)$ s'écrit :

$$y = H_{\Gamma y}(s)\Gamma + K_{uy}(s)u = \begin{bmatrix} H_{\Gamma y}(s) & H_{uy}(s) \end{bmatrix} \begin{bmatrix} \Gamma \\ u \end{bmatrix} = H(s) \begin{bmatrix} \Gamma \\ u \end{bmatrix} \quad (3-180)$$

On se propose alors de faire évoluer le problème (3-121) vers le problème (3-181).

$$\begin{aligned} & \min_{D_o, D_i, K_1(s), K_2(s)} \gamma \\ & \text{sous contraintes que :} \\ & T(s) \in \mathbf{RH}_\infty, \\ & K_1(s) \in \mathbf{RH}_\infty, \\ & Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \end{aligned} \quad (3-181)$$

Avec :

$$T(s) = \begin{bmatrix} A_{cl} & B_{cl} \\ C_{cl} & D_{cl} \end{bmatrix} = \begin{bmatrix} D_o \\ D_i^{-1} K_2(s) \end{bmatrix} \left(I + H_{uy}(s) K_2(s) \right)^{-1} \begin{bmatrix} H_{uy}(s) D_i & D_o^{-1} \end{bmatrix} \quad (3-182)$$

Notons que le calcul d'une correction par anticipation implique une contrainte supplémentaire, celle de la stabilité de l'anticipateur (qui est hors boucle).

On note pour simplifier :

$$\Theta_{k_1} = \begin{pmatrix} A_{k_1} & B_{k_1} \\ C_{k_1} & D_{k_1} \end{pmatrix}, \quad \Theta_{k_2} = \begin{pmatrix} A_{k_2} & B_{k_2} \\ C_{k_2} & D_{k_2} \end{pmatrix} \quad (3-183)$$

En posant :

$$\bar{\lambda} = \max \left[\max_l \left(\operatorname{Re}(\lambda_l(A_{k_1})) \right), \max_j \left(\operatorname{Re}(\lambda_j(A_{cl})) \right) \right] \quad (3-184)$$

Alors on résout le problème (3-181) en considérant le problème d'optimisation suivant :

$$\begin{aligned} & \min_{D_i, D_o, \Theta_{k_1}, \Theta_{k_2}} f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) \\ \text{si } & \bar{\lambda} \geq 0 \\ & f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = \bar{\lambda} \\ \text{sinon} & \\ & \gamma = \|T(s)\|_{\infty} \\ & f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = -\frac{1}{\gamma + f_c(H, K)} \\ \text{fin si} & \end{aligned} \quad (3-185)$$

L'extension multi-modèle est immédiate. Il suffit pour cela de définir une boule de N modèles \mathcal{D} centrée sur un modèle nominal $H_0(s)$ tel que :

$$\mathcal{D} = \left\{ H_i(s) = \left[H_{\Gamma y_i}(s) \quad H_{uy_i}(s) \right], i = 0, \dots, N \right\} \quad (3-186)$$

En posant :

$$\bar{\lambda} = \max \left[\max_l \left(\operatorname{Re}(\lambda_l(A_{k_1})) \right), \max_{j=0, \dots, N} \left[\max_i \left(\operatorname{Re}(\lambda_i(A_{cl_j})) \right) \right] \right] \quad (3-187)$$

On peut alors résoudre plusieurs types de problème de synthèse de robuste à 2 degrés de liberté. Par exemple :

- K_2 robuste en marge de stabilité, K_1 nominalement performant :

$$\begin{aligned}
& \min_{D_i, D_o, \Theta_{k_1}, \Theta_{k_2}} f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) \\
& \text{si } \bar{\lambda} \geq 0 \\
& \quad f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = \bar{\lambda} \\
& \text{sinon} \\
& \quad \gamma = \max_{j=0, \dots, N} \|T_j(s)\|_{\infty} \\
& \quad f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = -\frac{1}{\gamma + f_c(H_0, K)} \\
& \text{fin si}
\end{aligned} \tag{3-188}$$

- (K_2, K_1) robuste en performance :

$$\begin{aligned}
& \min_{D_i, D_o, \Theta_{k_1}, \Theta_{k_2}} f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) \\
& \text{si } \bar{\lambda} \geq 0 \\
& \quad f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = \bar{\lambda} \\
& \text{sinon} \\
& \quad \gamma_j = \|T_j(s)\|_{\infty} \\
& \quad f(D_i, D_o, \Theta_{k_1}, \Theta_{k_2}) = -\frac{1}{\max_{j=0, \dots, N} (\gamma_j + f_c(H_j, K))} \\
& \text{fin si}
\end{aligned} \tag{3-189}$$

- ...

Bien que l'approche soit efficace, à l'usage on se rend vite compte que la résolution du problème (3-185) prend énormément de temps car on cherche à déterminer un correcteur stabilisant en même temps qu'un correcteur dédié à la performance (hors boucle). Or il semble évident que la détermination du correcteur K_1 est conditionnée à celle du correcteur de boucle K_2 au préalable. Un cas particulièrement intéressant et fréquent est le cas où parmi les m exigences spécifiées, n d'entre elles (par exemple les n premières) ne peuvent pas servir à déterminer K_1 (par exemple, on ne peut pas déterminer K_1 en optimisant uniquement les exigences relatives au rejet de perturbations internes).

Posons :

$$\begin{aligned}
 Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m \Rightarrow f_c(H, K) &= \frac{\gamma_{obj}}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})}{0.2\bar{Z}_j}(Z_j - \bar{Z}_j)} \\
 Z_{n+1} < \bar{Z}_{n+1}, \dots, Z_m < \bar{Z}_m \Rightarrow h_c(H, K) &= \sum_{j=1}^{m-n} e^{\frac{\ln(\gamma_{obj})}{0.2\bar{Z}_j}(Z_j - \bar{Z}_j)}
 \end{aligned} \tag{3-190}$$

Basé sur (3-190), on peut imaginer une optimisation en n étapes successives :

- Etape 1 : à K_2 fixé, optimisation de K_1 par résolution du problème suivant :

$$\begin{aligned}
 &\text{Trouver } K_1(s) \text{ tel que :} \\
 &K_1(s) \in \mathbf{RH}_\infty, \\
 &Z_{n+1} < \bar{Z}_{n+1}, \dots, Z_m < \bar{Z}_m
 \end{aligned} \tag{3-191}$$

Les exigences doivent être évaluées avec K_2 fixé et le K_1 en cours.

En posant :

$$\bar{\lambda} = \max_l (\text{Re}(\lambda_l(A_{k_1}))) \tag{3-192}$$

Alors on détermine un anticipateur en résolvant le problème suivant :

$$\begin{aligned}
 &\min_{\Theta_{k_1}} f(\Theta_{k_1}) \\
 &\text{si } \bar{\lambda} \geq 0 \\
 &f(\Theta_{k_1}) = \bar{\lambda} \\
 &\text{sinon} \\
 &f(\Theta_{k_1}) = -\frac{1}{h_c(H, K)} \\
 &\text{fin si}
 \end{aligned} \tag{3-193}$$

A partir de là, soit on se satisfait du couple (K_1, K_2) soit on peut robustifier K_2 vis-à-vis d'une boule de modèles portant sur $H(s)$. Dans ce cas, on passera à l'étape 2.

- Etape 2 : à K_1 fixé optimisation de K_2 par résolution du problème suivant :

$$\begin{aligned}
 &\min_{D_o, D_i, K_2(s)} \gamma \\
 &\text{sous contraintes que :} \\
 &T(s) \in \mathbf{RH}_\infty, \\
 &Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m
 \end{aligned} \tag{3-194}$$

Les exigences doivent être évaluées avec K_1 fixé et K_2 en cours. Ce problème a déjà été développé en (3-124) et suivants. Il est résolu en considérant alors $f_c(H, K)$.

- Recommencer l'étape 1 avec le K_2 précédent etc...

L'approche s'initialise avec un correcteur K_2 déterminé par n'importe quelle méthode ; par exemple, on peut le calculer pour satisfaire quelques-unes des exigences.

3.5.4. Extension à la μ -synthèse

Nous avons vu en première partie comment la μ -synthèse structurée pouvait être reformulée pour l'optimisation stochastique.

De plus nous avons un exemple montrant comment la μ -synthèse pouvait être associée simplement avec la synthèse par loop-shaping comme sur la Figure 2-29.

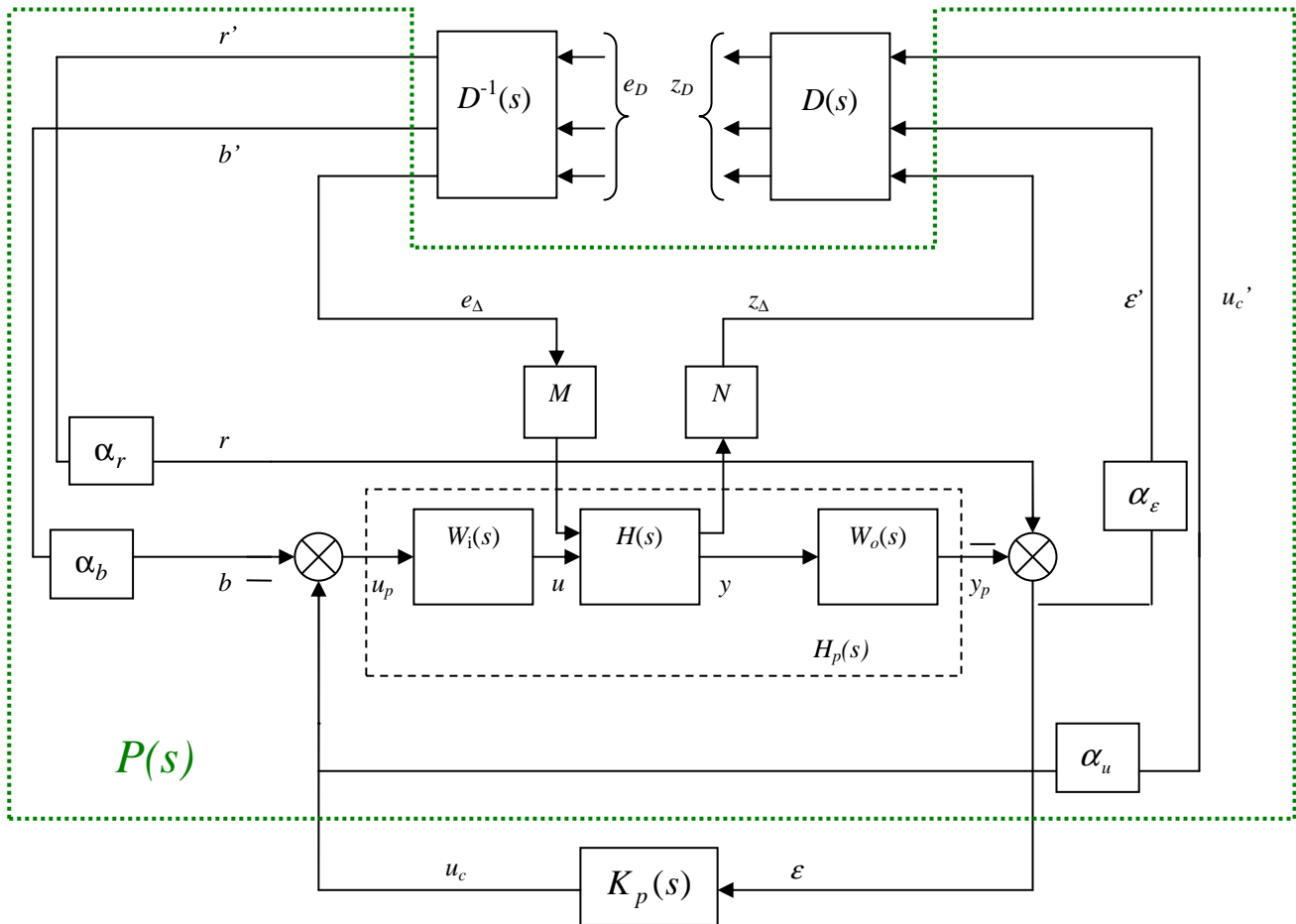


Figure 3-51: Schéma de commande pour la μ -synthèse

Nous avons vu qu'en choisissant $\alpha_\epsilon = \alpha_u = \alpha_r = \alpha_b = \gamma_{obj}^{-1/2}$, le problème de la μ -synthèse structurée peut-être approché par la résolution du problème (3-195).

$$\min_{D(s), K_p(s)} \left\| D(s)F_l(P, K_p)D(s)^{-1} \right\|_{\infty}$$

sous contraintes que

$$F_l(P, K_p) \in \mathbf{RH}_{\infty}$$
(3-195)

Ainsi le correcteur K_p sera robuste en performance lorsque $\left\| D(s)F_l(P, K_p)D(s)^{-1} \right\|_{\infty} \leq 1$.

En externalisant les pondérations fréquentielles (ce qui permet de travailler directement sur le correcteur final) et en les imposant statiques, rien ne nous empêche d'étendre les techniques précédentes à la μ -synthèse en rajoutant dans les variables de décision, les scalings fréquentiels (ceux de la μ -synthèse) à déterminer. Ainsi on peut chercher à résoudre le problème (3-196).

$$\min_{D_o, D_i, K(s), D(s)} \gamma$$

sous contraintes que :

$$F_l(P, K) \in \mathbf{RH}_{\infty},$$

$$Z_1 < \bar{Z}_1, Z_2 < \bar{Z}_2, \dots, Z_m < \bar{Z}_m$$
(3-196)

Avec :

$$\gamma = \left\| D(s)F_l(P, K)D(s)^{-1} \right\|_{\infty}$$
(3-197)

Comme pour le problème précédent, on peut approcher ce problème par le suivant :

$$\min_{\substack{D_o, D_i, K(s), D(s) \\ F_l(P, K) \in \mathbf{RH}_{\infty}}} \gamma + f_c(H, K)$$
(3-198)

Avec :

$$f_c(H, K) = \frac{\gamma_{obj}}{m} \sum_{j=1}^m e^{\frac{\ln(\gamma_{obj})(Z_j - \bar{Z}_j)}{\varepsilon_j}}$$
(3-199)

Le traitement de ce problème nécessite cependant une petite adaptation car la valeur de γ finale visée à l'issue de la μ -synthèse est de 1, alors que $f_c(H, K)$ est définie dans le cadre de la synthèse par loop-shaping avec $\gamma_{obj} = 3$ et ne peut être égale à 1. Afin de recouvrer la faisabilité du problème d'optimisation, il suffit de compléter les canaux d'incertitudes par des coefficients de normalisation $\alpha_M = \alpha_N = \gamma_{obj}^{1/2}$ et de choisir $\alpha_{\varepsilon} = \alpha_u = \alpha_r = \alpha_b = 1$ comme sur la Figure 3-52.

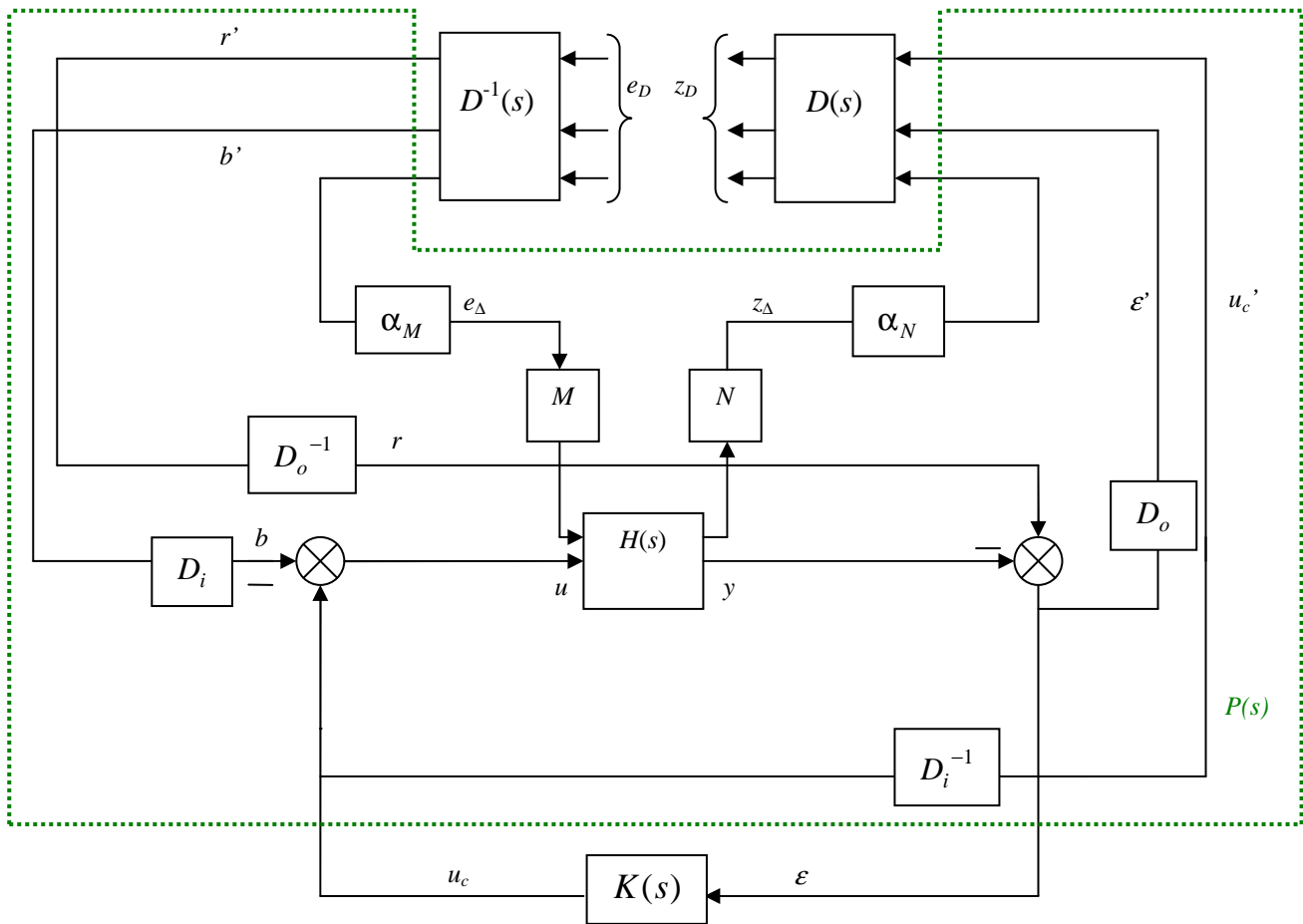


Figure 3-52: Schéma de commande pour la μ -synthèse

En effet, les incertitudes étant normalisées, on aura dans ce cas à la fin de l'optimisation $\|D(s)F_l(P, K)D(s)^{-1}\|_\infty \rightarrow \gamma_{obj}$ ce qui permet alors de résoudre le problème (3-198).

Conclusions et Perspectives

Dans cette thèse, nous avons montré l'efficacité des métaheuristiques à résoudre des problèmes de commande complexes dans le contexte industriel de la stabilisation inertielle de ligne de visée. Parce que ce type d'optimisation ne nécessite pas la formulation du gradient, nous avons pu synthétiser des correcteurs robustes et structurés en considérant des critères de toute forme. L'algorithme de l'Evolution Différentielle, dans une version *DE/Rand_to_Best/1* adaptée, s'avère très efficace pour les problèmes de commande : dans le cadre des problèmes structurés de synthèse H_∞ et de μ -synthèse, il a conduit à des résultats très compétitifs et dans le cadre de la synthèse structurée sur critère haut niveau, on a pu constater un gain en performance, en robustesse, et en temps de développement car il n'est plus nécessaire de procéder à des simplifications donnant lieu à des phases itératives lors du développement. Les techniques développées dans cette thèse pourront naturellement être employées dans d'autres domaines de l'industrie où les systèmes considérés et les spécifications systèmes haut niveau sont tout aussi complexes.

Dans ce travail nous avons contribué au développement d'un domaine de la recherche rassemblant l'optimisation et l'automatique. Ainsi, on pourra poursuivre dans l'amélioration des travaux en axant ses efforts dans l'un et/ou l'autre de ces domaines. Citons quelques pistes intéressantes (non exhaustives) :

- Dans le domaine de l'optimisation :
 - Un travail sur le temps de calcul est envisageable. Pour cela plusieurs approches sont possibles : techniques de métamodélisation de la fonction de coût, optimisation multi-niveaux, travail sur le critère d'arrêt, ...
 - L'approche multiobjectifs pourrait être une piste d'amélioration. Elle consiste à mettre chaque contrainte $Z_i \leq \bar{Z}_i \Leftrightarrow g_i(X) = Z_i - \bar{Z}_i \leq 0$ sous la forme d'un objectif à minimiser $h_i(X) = g_i(X) + |g_i(X)|$ ce qui permet de définir un critère multiobjectifs de la forme $F = (\gamma, h_1(X), h_2(X), \dots, h_m(X))$. Toutefois le problème multiobjectifs sera d'autant plus difficile à résoudre que le nombre de contraintes m est grand. Par ailleurs, cela implique de choisir un algorithme pouvant être employé dans le cadre multiobjectifs.
 - Nous avons vu dans cette thèse que finalement, à partir d'une spécification donnée, l'automaticien n'a plus qu'à choisir l'ordre des pondérations et/ou du correcteur final, ce qui en soit n'est pas une tâche aisée. On peut par contre envisager une couche d'optimisation supplémentaire pour choisir l'ordre des pondérations et/ou des correcteurs

le plus adéquat pour le problème donné : pour cela, l'emploi des techniques d'optimisation coûteuse bayésienne peut être intéressant, la fonction de coût à optimiser, bruitée par son caractère stochastique, étant évaluée à partir d'un nombre de *runs* donné du problème développé dans cette thèse.

- ...
- Dans le domaine de l'automatique :
 - Nous avons vu un moyen de reformuler pour l'optimisation stochastique les problèmes de commande robuste de type H^∞ et μ -synthèse. Rien n'empêche d'aller plus loin par exemple en abordant la problématique de la synthèse structurée dans le cadre des systèmes Linéaires à Paramètres Variant et d'étendre la technique développée dans cette thèse pour la synthèse de correcteur sous critère de toute forme dans ce cadre.
 - D'autres problématiques liées à la robustesse peuvent être abordées. Par exemple quel est le correcteur qui satisfait une spécification système haut niveau donné, et stabilisant la boule de modèles la plus grande possible.
 - Le problème inverse qui consiste en la détermination de pondérations de synthèse à partir d'un correcteur et d'une schématique donnée est une piste intéressante pour l'amélioration de correcteurs existants.
 - ...

Enfin la réunion de ces deux grandes thématiques pourra se faire avec le développement d'un outil numérique de synthèse de correcteurs structurés vis-à-vis de critères de toute forme. La puissance des outils numériques actuels tels que *Matlab/Simulink* permet d'envisager d'évaluer directement les exigences spécifiées sur des systèmes complexes non linéaires.

Annexe A

Une mise sous forme LFT d'un système électromécanique avec modes souples incertains

A.1. Modèle général et objectif

Le système est supposé disposer d'une entrée notée u (homogène à une tension de consigne ou un courant de consigne suivant les cas) et d'une sortie mesurée notée y (homogène à une position ou une vitesse suivant les cas).

Le modèle général employé pour l'étude est celui de la Figure A-1 . Il comprend :

- Un modèle mécanique de type « n modes souples »,

$$\frac{\theta}{C_{em}} = \left(\frac{1}{Js^2} + \sum_{i=1}^n \frac{a_i}{\frac{s^2}{\omega_i^2} + \frac{2\xi}{\omega_i}s + 1} \right) = H_m(s) \quad (\text{A-1})$$

- Associé à une motorisation de type moteur à courant continu,
 - soit piloté en courant via une boucle de courant, dans ce cas :

$$\frac{C_{em}}{u} = \frac{K_c}{1 + \tau_{el}s} = H_{el}(s) \quad (\text{A-2})$$

- soit attaqué directement en tension¹⁹, dans ce cas :

¹⁹ La f.c.e.m est supposée négligeable ici

$$\frac{C_{em}}{u} \approx \frac{K_c}{R + Ls} = H_{el}(s) \quad (\text{A-3})$$

– Associé à une chaîne de mesure,

- soit de position, dans ce cas :

$$\frac{y}{\theta} = H_r(s) = H_{mes}(s) \quad (\text{A-4})$$

- soit de vitesse, dans ce cas :

$$\frac{y}{\theta} = sH_r(s) = H_{mes}(s) \quad (\text{A-5})$$

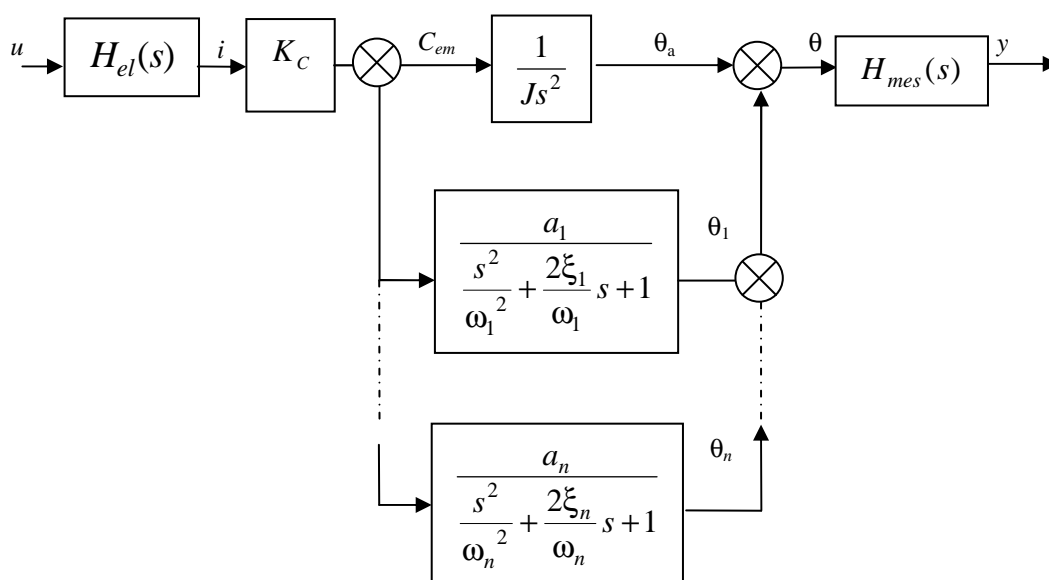


Figure A-1: système électromécanique à modes souples

Pour de tels systèmes, l'expérience montre que les paramètres suivants varient majoritairement :

- la constante de temps τ_{el} de la boucle de courant ou la résistance du moteur R suivant le mode de pilotage considéré dans $H_{el}(s)$,
- les facteurs a_i de contribution des modes souples,
- les amortissements des modes ξ_i souples,
- les pulsations propres ω_i de modes souples.

L'objectif est de mettre ce système et ses incertitudes sous forme LFT pour la μ -analyse (la matrice d'incertitude étant diagonale), c'est-à-dire sous la forme de la Figure 2-26

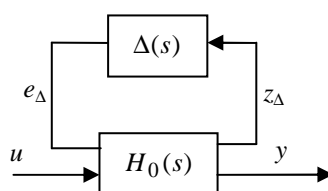


Figure A-2: forme LFT

A.2. Mise sous forme d'état du modèle nominal

Dans un premier temps, on établit une représentation d'état paramétrique du modèle nominal.

A.2.1. Mise sous forme d'état des blocs élémentaires

A.2.1.1. Motorisation

- cas du moteur piloté en courant (boucle de courant) :

$$\frac{C_{em}}{u} = \frac{K_c}{1 + \tau_{el}s} = H_{el}(s)$$

$$\Leftrightarrow$$

$$\frac{di}{dt} = -\frac{i(t)}{\tau_{el}} + \frac{i_c}{\tau_{el}} = -\omega_{el}i(t) + \omega_{el}u$$

$$C_{em} = K_c i(t)$$

$$\Leftrightarrow$$

$$\frac{dX_{el}}{dt} = A_{el}X_{el} + B_{el}u$$

$$C_{em} = C_{el}X_{el}$$
(A-6)

- cas du moteur piloté directement en tension :

$$\begin{aligned}
\frac{C_{em}}{u} &= \frac{K_c}{R + Ls} = H_{el}(s) \\
&\Leftrightarrow \\
\frac{di}{dt} &= -\frac{R}{L}i(t) + \frac{u}{L} \\
C_{em} &= K_c i(t) \\
&\Leftrightarrow \\
\frac{dX_{el}}{dt} &= A_{el}X_{el} + B_{el}u \\
C_{em} &= C_{el}X_{el}
\end{aligned} \tag{A-7}$$

A.2.1.2. Mécanique

- Inertie pure :

$$\begin{aligned}
\theta_a &= \frac{C_{em}}{Js^2} \\
&\Leftrightarrow \\
\frac{d}{dt} \begin{pmatrix} \theta_a \\ \dot{\theta}_a \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_a \\ \dot{\theta}_a \end{pmatrix} + \begin{pmatrix} 0 \\ J^{-1} \end{pmatrix} C_{em} \\
\theta_a &= (1 \quad 0) \begin{pmatrix} \theta_a \\ \dot{\theta}_a \end{pmatrix} \\
&\Leftrightarrow \\
\frac{dX_a}{dt} &= A_a X_a + B_a C_{em} \\
\theta_a &= C_a X_a
\end{aligned} \tag{A-8}$$

- $i^{\text{ème}}$ mode souple :

$$\begin{aligned}
 \theta_i &= \frac{a_i}{\frac{s^2}{\omega_i^2} + \frac{2\xi_i}{\omega_i}s + 1} C_{em} \\
 &\Leftrightarrow \\
 \frac{d}{dt} \begin{pmatrix} \theta_i \\ \dot{\theta}_i \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ -\omega_i^2 & -2\xi_i\omega_i \end{pmatrix} \begin{pmatrix} \theta_i \\ \dot{\theta}_i \end{pmatrix} + \begin{pmatrix} 0 \\ \omega_i^2 a_i \end{pmatrix} C_{em} \\
 &\Leftrightarrow \\
 \theta_i &= (1 \ 0) \begin{pmatrix} \theta_i \\ \dot{\theta}_i \end{pmatrix} \\
 &\Leftrightarrow \\
 \frac{dX_i}{dt} &= A_i X_i + B_i C_{em} \\
 \theta_i &= C_i X_i
 \end{aligned} \tag{A-9}$$

- Equation d'état du modèle mécanique n modes souples et inertie pure :

$$\left. \begin{aligned}
 \frac{dX_a}{dt} &= A_a X_a + B_a C_{em} \\
 \frac{dX_1}{dt} &= A_1 X_1 + B_1 C_{em} \\
 &\dots \\
 \frac{dX_i}{dt} &= A_i X_i + B_i C_{em} \\
 &\dots \\
 \frac{dX_n}{dt} &= A_n X_n + B_n C_{em}
 \end{aligned} \right\} \tag{A-10}$$

$$\Leftrightarrow$$

$$\frac{d}{dt} \begin{pmatrix} X_a \\ X_1 \\ \dots \\ X_i \\ \dots \\ X_n \end{pmatrix} = \begin{pmatrix} A_a & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & A_i & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & A_n \end{pmatrix} \begin{pmatrix} X_a \\ X_1 \\ \dots \\ X_i \\ \dots \\ X_n \end{pmatrix} + \begin{pmatrix} B_a \\ B_1 \\ \dots \\ B_i \\ \dots \\ B_n \end{pmatrix} C_{em} \Leftrightarrow \frac{dX_m}{dt} = A_m X_m + B_m C_{em}$$

A.2.1.3. Chaîne d'acquisition

$$\begin{aligned} \frac{y}{\theta} &= H_r(s) \\ &\Leftrightarrow \\ \frac{dX_r}{dt} &= A_r X_r + B_r \theta = A_r X_r + B_r \left(\theta_a + \sum_i \theta_i \right) \\ y &= C_r X_r + D_r \theta \end{aligned} \quad (\text{A-11})$$

A.2.2. Forme d'état globale

On a :

$$\left. \begin{aligned} \frac{dX_{el}}{dt} &= A_{el} X_{el} + B_{el} U \\ \frac{dX_m}{dt} &= A_m X_m + B_m C_{em} = B_m C_{el} \cdot X_{el} + A_m X_m \\ \frac{dX_r}{dt} &= A_r X_r + B_r \theta = B_r C_m X_m + A_r X_r \\ y &= C_r \cdot X_r + D_r \theta = C_r \cdot X_r + D_r C_m X_m \end{aligned} \right\} \quad (\text{A-12})$$

A.3. Mise sous forme LFT des incertitudes

A.3.1. Mise sous forme LFT des blocs élémentaires

La représentation d'état nominale précédente est sujette à des variations paramétriques. Le but est de mettre le modèle incertain sous la forme standard pour la μ -analyse.

A.3.1.1. Motorisation

- cas du moteur piloté en courant (boucle de courant) :

On a :

$$\begin{aligned} \frac{di}{dt} &= -\frac{i(t)}{\tau_{el}} + \frac{i_c}{\tau_{el}} = -\omega_{el} i(t) + \omega_{el} u \\ C_{em} &= K_c i(t) \end{aligned} \quad (\text{A-13})$$

Pour ce bloc, le paramètre variant majoritairement est la constante de temps électrique du pilotage en courant de sorte que²⁰ :

$$\omega_{el} = \omega_{el0} + \delta\omega_{el}\omega_{el1}, \quad |\delta\omega_{el}| \leq 1, \quad \omega_{el1} = \frac{\omega_{el\max} - \omega_{el\min}}{2} \quad (\text{A-14})$$

De sorte que :

$$\begin{aligned} \frac{di}{dt} &= -\omega_{el}i(t) + \omega_{el}u \\ &= -(\omega_{el0} + \delta\omega_{el}\omega_{el1})i(t) + (\omega_{el0} + \delta\omega_{el}\omega_{el1})u \\ &= -\omega_{el0}i(t) + \omega_{el0}u - \delta\omega_{el}\omega_{el1}i(t) + \delta\omega_{el}\omega_{el1}u \end{aligned} \quad (\text{A-15})$$

Ou encore²¹ :

$$\frac{di}{dt} = -\omega_{el0}i(t) + \omega_{el0}u + \sqrt{\omega_{el1}}e_{\Delta el_i} + \sqrt{\omega_{el1}}e_{\Delta el_u} \quad (\text{A-16})$$

Avec :

$$\begin{aligned} e_{\Delta el_i} &= \delta\omega_{el}z_{\Delta el_i}, \quad |\delta\omega_{el}| \leq 1 \\ z_{\Delta el_i} &= -\sqrt{\omega_{el1}}i(t) \\ e_{\Delta el_u} &= \delta\omega_{el}z_{\Delta el_u}, \quad |\delta\omega_{el}| \leq 1 \\ z_{\Delta el_u} &= \sqrt{\omega_{el1}}u \end{aligned} \quad (\text{A-17})$$

C'est-à-dire :

²⁰ ω_{el0} est supposé être le milieu de l'intervalle de variation $\omega_{el0} = \frac{\omega_{el\min} + \omega_{el\max}}{2}$

²¹ On choisit de répartir équitablement les poids de l'incertitude paramétrique sur $z_{\Delta el}$ et sur $e_{\Delta el}$

$$\frac{dX_{el}}{dt} = A_{el}X_{el} + B_{el}u + \begin{pmatrix} \sqrt{\omega_{el1}} & \sqrt{\omega_{el1}} \end{pmatrix} \begin{pmatrix} e_{\Delta el_i} \\ e_{\Delta el_u} \end{pmatrix}$$

$$\begin{pmatrix} z_{\Delta el_i} \\ z_{\Delta el_u} \end{pmatrix} = \begin{pmatrix} -\sqrt{\omega_{el1}} \\ 0 \end{pmatrix} X_{el} + \begin{pmatrix} 0 \\ \sqrt{\omega_{el1}} \end{pmatrix} u \quad (\text{A-18})$$

$$C_{em} = C_{el}X_{el}$$

$$e_{\Delta el_i} = \delta\omega_{el}z_{\Delta el_i}, |\delta\omega_{el}| \leq 1$$

$$e_{\Delta el_u} = \delta\omega_{el}z_{\Delta el_u}, |\delta\omega_{el}| \leq 1$$

Ou encore noté :

$$\frac{dX_{el}}{dt} = A_{el}X_{el} + B_{el}u + M_{el}e_{\Delta el}$$

$$z_{\Delta el} = N_{el}X_{el} + O_{el}u$$

$$C_{em} = C_{el}X_{el} \quad (\text{A-19})$$

$$e_{\Delta el} = \Delta_{el}z_{\Delta el}, \|\Delta_{el}\|_{\infty} \leq 1$$

- cas du moteur piloté en tension :

On rappelle que :

$$\frac{di}{dt} = -\frac{R}{L}i(t) + \frac{u}{L} \quad (\text{A-20})$$

$$C_{em} = K_c i(t)$$

Pour ce bloc, le paramètre variant majoritairement est la résistance électrique du moteur de sorte que $R \in [R_{\min}; R_{\max}]$ et ainsi ²²

$$R = R_0 + \delta\omega_R R_1, |\delta R| \leq 1, R_1 = \frac{R_{\max} - R_{\min}}{2} \quad (\text{A-21})$$

Il vient :

²² R_0 est supposée être le milieu de l'intervalle de variation $R_0 = \frac{R_{\min} + R_{\max}}{2}$

$$\begin{aligned}
\frac{di}{dt} &= -\frac{R}{L}i(t) + \frac{u}{L} \\
&= -\frac{1}{L}(R_0 + \delta\omega_R R_1)i(t) + \frac{u}{L} \\
&= -\frac{R_0}{L}i(t) - \delta\omega_R \frac{R_1}{L}i(t) + \frac{u}{L}
\end{aligned} \tag{A-22}$$

Ou encore²³ :

$$\frac{di}{dt} = -\frac{R_0}{L}i(t) + \frac{u}{L} + \sqrt{\frac{R_1}{L}}e_{\Delta el_i} \tag{A-23}$$

Avec :

$$\begin{aligned}
e_{\Delta el_i} &= \delta R z_{\Delta el_i}, \quad |\delta R| \leq 1 \\
z_{\Delta el_i} &= -\sqrt{\frac{R_1}{L}}i(t)
\end{aligned} \tag{A-24}$$

C'est-à-dire :

$$\begin{aligned}
\frac{dX_{el}}{dt} &= A_{el}X_{el} + B_{el}u + \sqrt{\frac{R_1}{L}}e_{\Delta el_i} \\
z_{\Delta el_i} &= -\sqrt{\frac{R_1}{L}}X_{el}
\end{aligned} \tag{A-25}$$

$$C_{em} = C_{el}X_{el}$$

$$e_{\Delta el_i} = \delta R z_{\Delta el_i}, \quad |\delta R| \leq 1$$

Ou encore noté :

$$\begin{aligned}
\frac{dX_{el}}{dt} &= A_{el}X_{el} + B_{el}u + M_{el}e_{\Delta el} \\
z_{\Delta el} &= N_{el}X_{el} \\
C_{em} &= C_{el}X_{el}
\end{aligned} \tag{A-26}$$

$$e_{\Delta el} = \Delta_{el} z_{\Delta el}, \quad \|\Delta_{el}\|_{\infty} \leq 1$$

L'entrée u est homogène à une tension de commande

²³ On choisit de répartir équitablement les poids de l'incertitude paramétrique sur $w_{\delta el}$ et sur $v_{\delta el}$

A.3.1.2. Mécanique

- $i^{\text{ème}}$ mode souple :

Pour les modes souples, les paramètres variant majoritairement sont :

- ξ_i de sorte que $\xi_i \in [\xi_{i_{\min}}; \xi_{i_{\max}}]$ et : $\xi_i = \xi_{i_0} + \delta\xi_i \xi_{i_1}$, $|\delta\xi_i| \leq 1$, $\xi_{i_1} = \frac{\xi_{i_{\max}} - \xi_{i_{\min}}}{2}$
- a_i de sorte que $a_i \in [a_{i_{\min}}; a_{i_{\max}}]$ et : $a_i = a_{i_0} + \delta a_i a_{i_1}$, $|\delta a_i| \leq 1$, $a_{i_1} = \frac{a_{i_{\max}} - a_{i_{\min}}}{2}$
- ω_i de sorte que $\omega_i \in [\omega_{i_{\min}}; \omega_{i_{\max}}]$ et : $\omega_i = \omega_{i_0} + \delta\omega_i \omega_{i_1}$, $|\delta\omega_i| \leq 1$, $\omega_{i_1} = \frac{\omega_{i_{\max}} - \omega_{i_{\min}}}{2}$

D'après l'équation d'état :

$$\begin{aligned} \frac{d\dot{\theta}_i}{dt} &= -\omega_i^2 \theta_i - 2\xi_i \omega_i \dot{\theta}_i + a_i \omega_i^2 C_{em} \\ &= -(\omega_{i_0} + \delta\omega_i \omega_{i_1})^2 \theta_i - 2(\xi_{i_0} + \delta\xi_i \xi_{i_1})(\omega_{i_0} + \delta\omega_i \omega_{i_1}) \dot{\theta}_i \\ &\quad + (\omega_{i_0} + \delta\omega_i \omega_{i_1})^2 (a_{i_0} + \delta a_i a_{i_1}) C_{em} \tag{A-27} \\ &= -(\omega_{i_0}^2 + 2\delta\omega_i \omega_{i_0} \omega_{i_1} + (\delta\omega_i \omega_{i_1})^2) \theta_i - 2(\xi_{i_0} \omega_{i_0} + \xi_{i_0} \delta\omega_i \omega_{i_1} + \delta\xi_i \xi_{i_1} \omega_{i_0} + \delta\xi_i \xi_{i_1} \delta\omega_i \omega_{i_1}) \dot{\theta}_i \\ &\quad + (\omega_{i_0}^2 + 2\delta\omega_i \omega_{i_0} \omega_{i_1} + (\delta\omega_i \omega_{i_1})^2) a_{i_0} C_{em} + (\omega_{i_0}^2 + 2\delta\omega_i \omega_{i_0} \omega_{i_1} + (\delta\omega_i \omega_{i_1})^2) \delta a_i a_{i_1} C_{em} \end{aligned}$$

Or en supposant que les termes d'ordre 2 relatifs aux variations paramétriques sont négligeables :

$$\begin{cases} \delta\xi_i < 1 \\ \delta\omega_i < 1 \\ \delta a_i < 1 \end{cases} \Rightarrow \begin{cases} \delta\xi_i \delta\omega_i \ll 1 \\ \delta\omega_i^2 \ll 1 \\ \delta a_i \delta\omega_i \ll 1 \end{cases} \tag{A-28}$$

Alors on peut procéder à la simplification suivante :

$$\begin{aligned} \frac{d\dot{\theta}_i}{dt} &\approx -(\omega_{i_0}^2 + 2\delta\omega_i \omega_{i_0} \omega_{i_1}) \theta_i \\ &\quad - 2(\xi_{i_0} \omega_{i_0} + \xi_{i_0} \delta\omega_i \omega_{i_1} + \delta\xi_i \xi_{i_1} \omega_{i_0}) \dot{\theta}_i \\ &\quad + (\omega_{i_0}^2 a_{i_0} + 2a_{i_0} \delta\omega_i \omega_{i_0} \omega_{i_1} + \omega_{i_0} \delta a_i a_{i_1}) C_{em} \end{aligned} \tag{A-29}$$

C'est-à-dire :

$$\begin{aligned}
 \frac{d\dot{\theta}_i}{dt} &\approx -\omega_{i0}^2 \theta_i - 2\xi_{i0} \omega_{i0} \dot{\theta}_i + a_{i0} \omega_{i0}^2 C_{em} \\
 &\quad - \delta\omega_i 2\omega_{i1} \left(\omega_{i0} \theta_i + \xi_{i0} \dot{\theta}_i \right) - \delta\xi_i (2\xi_{i1} \omega_{i0}) \dot{\theta}_i \\
 &\quad + \delta\omega_i (a_{i0} \omega_{i0} 2\omega_{i1}) C_{em} + \delta a_i \omega_{i0} a_{i1} C_{em}
 \end{aligned} \tag{A-30}$$

Il est alors posé ²⁴ :

$$\begin{aligned}
 e_{\Delta\omega_i\Theta} &= \delta\omega_i z_{\Delta\omega_i\Theta} \\
 z_{\Delta\omega_i\Theta} &= -\sqrt{2\omega_{i1}} \left(\omega_{i0} \theta_i + \xi_{i0} \dot{\theta}_i \right) \\
 e_{\Delta\omega_i C} &= \delta\omega_i z_{\Delta\omega_i C} \\
 z_{\Delta\omega_i C} &= \sqrt{a_{i0} \omega_{i0} 2\omega_{i1}} C_{em} \\
 e_{\Delta\xi_i} &= \delta\xi_i z_{\Delta\xi_i} \\
 z_{\Delta\xi_i} &= -\sqrt{(2\xi_{i1} \omega_{i0})} \dot{\theta}_i \\
 e_{\Delta a_i} &= \delta a_i z_{\Delta a_i} \\
 z_{\Delta a_i} &= \sqrt{\omega_{i0} a_{i1}} C_{em}
 \end{aligned} \tag{A-31}$$

De sorte que :

$$\begin{aligned}
 \frac{d\dot{\theta}_i}{dt} &\approx -\omega_{i0}^2 \theta_i - 2\xi_{i0} \omega_{i0} \dot{\theta}_i + \omega_{i0}^2 C_{em} \\
 &\quad + \sqrt{2\omega_{i1}} e_{\Delta\omega_i\Theta} + \sqrt{(2\xi_{i1} \omega_{i0})} e_{\Delta\xi_i} \\
 &\quad + \sqrt{a_{i0} \omega_{i0} 2\omega_{i1}} e_{\Delta\omega_i C} + \sqrt{\omega_{i0} a_{i1}} e_{\Delta a_i}
 \end{aligned} \tag{A-32}$$

Au final sous forme matricielle :

²⁴ On choisit de répartir équitablement le poids des incertitudes paramétriques sur les z_{Δ} et sur e_{Δ}

$$\frac{dX_i}{dt} = A_i X_i + B_i C_{em} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sqrt{2\omega_{i1}} & \sqrt{a_{i0}\omega_{i0}2\omega_{i1}} & \sqrt{2\xi_{i1}\omega_{i0}} & \sqrt{\omega_{i0}a_{i1}} \end{pmatrix} \begin{pmatrix} e_{\Delta\omega_i\Theta} \\ e_{\Delta\omega_i C} \\ e_{\Delta\xi_i} \\ e_{\Delta a_i} \end{pmatrix}$$

$$\begin{pmatrix} z_{\Delta\omega_i\Theta} \\ z_{\Delta\omega_i C} \\ z_{\Delta\xi_i} \\ z_{\Delta a_i} \end{pmatrix} = \begin{pmatrix} -\sqrt{2\omega_{i1}\omega_{i0}} & -\sqrt{2\omega_{i1}\xi_{i0}} \\ 0 & 0 \\ 0 & -\sqrt{2\xi_{i1}\omega_{i0}} \\ 0 & 0 \end{pmatrix} X_i + \begin{pmatrix} 0 \\ \sqrt{a_{i0}\omega_{i0}2\omega_{i1}} \\ 0 \\ \sqrt{\omega_{i0}a_{i1}} \end{pmatrix} C_{em} \quad (\text{A-33})$$

$$\theta_i = C_i X_i$$

$$e_{\Delta\xi_i} = \delta\xi_i z_{\Delta\xi_i}, \quad |\delta\xi_i| \leq 1$$

$$e_{\Delta a_i} = \delta a_i z_{\Delta a_i}, \quad |\delta a_i| \leq 1$$

$$z_{\Delta\omega_i\Theta} = \delta\omega_i e_{\Delta\omega_i\Theta}, \quad |\delta\omega_i| \leq 1$$

$$z_{\Delta\omega_i C} = \delta\omega_i e_{\Delta\omega_i C}, \quad |\delta\omega_i| \leq 1$$

Soit encore :

$$\frac{dX_i}{dt} = A_i X_i + B_i C_{em} + M_i \begin{pmatrix} e_{\Delta\omega_i\Theta} \\ e_{\Delta\omega_i C} \\ e_{\Delta\xi_i} \\ e_{\Delta a_i} \end{pmatrix}$$

$$\theta_i = C_i X_i$$

$$\begin{pmatrix} z_{\Delta\omega_i\Theta} \\ z_{\Delta\omega_i C} \\ z_{\Delta\xi_i} \\ z_{\Delta a_i} \end{pmatrix} = N_i X_i + O_i C_{em}$$

(A-34)

$$e_{\Delta\xi_i} = \delta\xi_i z_{\Delta\xi_i}, \quad |\delta\xi_i| \leq 1$$

$$e_{\Delta a_i} = \delta a_i z_{\Delta a_i}, \quad |\delta a_i| \leq 1$$

$$z_{\Delta\omega_i\Theta} = \delta\omega_i e_{\Delta\omega_i\Theta}, \quad |\delta\omega_i| \leq 1$$

$$z_{\Delta\omega_i C} = \delta\omega_i e_{\Delta\omega_i C}, \quad |\delta\omega_i| \leq 1$$

$$N_i = \begin{pmatrix} -\sqrt{2\omega_{i1}}\omega_{i0} & -\sqrt{2\omega_{i1}}\xi_{i0} \\ 0 & 0 \\ 0 & -\sqrt{2\xi_{i1}}\omega_{i0} \\ 0 & 0 \end{pmatrix}, \quad O_i = \begin{pmatrix} 0 \\ \sqrt{a_{i0}\omega_{i0}2\omega_{i1}} \\ 0 \\ \sqrt{\omega_{i0}a_{i1}} \end{pmatrix}$$

$$M_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sqrt{2\omega_{i1}} & \sqrt{a_{i0}\omega_{i0}2\omega_{i1}} & \sqrt{2\xi_{i1}\omega_{i0}} & \sqrt{\omega_{i0}a_{i1}} \end{pmatrix}$$

- Equation d'état du modèle mécanique incertain avec n modes souples et inertie pure :

$$\frac{d}{dt} \begin{pmatrix} X_a \\ X_1 \\ X_i \\ X_n \end{pmatrix} = \begin{pmatrix} A_a & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 \\ 0 & 0 & A_i & 0 \\ 0 & 0 & 0 & A_n \end{pmatrix} \begin{pmatrix} X_a \\ X_1 \\ X_i \\ X_n \end{pmatrix} + \begin{pmatrix} B_a \\ B_1 \\ B_i \\ B_n \end{pmatrix} C_{em} + \begin{pmatrix} 0 & 0 & 0 \\ M_1 & 0 & 0 \\ 0 & M_i & 0 \\ 0 & 0 & M_n \end{pmatrix} \begin{pmatrix} e_{\Delta\omega_1\Theta} \\ e_{\Delta\omega_1C} \\ e_{\Delta\xi_1} \\ e_{\Delta a_1} \\ e_{\Delta\omega_1\Theta} \\ e_{\Delta\omega_1C} \\ e_{\Delta\xi_i} \\ e_{\Delta a_i} \\ e_{\Delta\omega_n\Theta} \\ e_{\Delta\omega_nC} \\ e_{\Delta\xi_n} \\ e_{\Delta a_n} \end{pmatrix}$$

$$\begin{pmatrix} z_{\Delta\omega_1\Theta} \\ z_{\Delta\omega_1C} \\ z_{\Delta\xi_1} \\ z_{\Delta a_1} \\ z_{\Delta\omega_1\Theta} \\ z_{\Delta\omega_1C} \\ z_{\Delta\xi_i} \\ z_{\Delta a_i} \\ z_{\Delta\omega_n\Theta} \\ z_{\Delta\omega_nC} \\ z_{\Delta\xi_n} \\ z_{\Delta a_n} \end{pmatrix} = \begin{pmatrix} 0 & N_1 & 0 & 0 \\ 0 & 0 & N_i & 0 \\ 0 & 0 & 0 & N_n \end{pmatrix} \begin{pmatrix} X_a \\ X_1 \\ X_i \\ X_n \end{pmatrix} + \begin{pmatrix} O_1 \\ O_i \\ O_n \end{pmatrix} C_{em}$$

(A-35)

$$\begin{pmatrix} e_{\Delta\omega_1\Theta} \\ e_{\Delta\omega_1C} \\ e_{\Delta\xi_1} \\ e_{\Delta a_1} \\ e_{\Delta\omega_1\Theta} \\ e_{\Delta\omega_1C} \\ e_{\Delta\xi_i} \\ e_{\Delta a_i} \\ e_{\Delta\omega_n\Theta} \\ e_{\Delta\omega_nC} \\ e_{\Delta\xi_n} \\ e_{\Delta a_n} \end{pmatrix} = \begin{pmatrix} \delta\omega_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta\omega_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta\xi_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta\omega_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta\omega_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \delta\xi_i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta a_i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta\omega_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta\omega_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta\xi_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta a_n \end{pmatrix} \begin{pmatrix} z_{\Delta\omega_1\Theta} \\ z_{\Delta\omega_1C} \\ z_{\Delta\xi_1} \\ z_{\Delta a_1} \\ z_{\Delta\omega_1\Theta} \\ z_{\Delta\omega_1C} \\ z_{\Delta\xi_i} \\ z_{\Delta a_i} \\ z_{\Delta\omega_n\Theta} \\ z_{\Delta\omega_nC} \\ z_{\Delta\xi_n} \\ z_{\Delta a_n} \end{pmatrix}$$

$$|\delta\xi_i| \leq 1, \quad i = 1..n$$

$$|\delta a_i| \leq 1, \quad i = 1..n$$

$$|\delta\omega_i| \leq 1, \quad i = 1..n$$

Que l'on note pour simplifier :

$$\begin{aligned}\frac{dX_m}{dt} &= A_m X_m + B_m C_{em} + M_m e_{\Delta m} \\ z_{\Delta m} &= N_m X_m + O_m C_{em} \\ e_{\Delta m} &= \Delta_m z_{\Delta m}, \quad \|\Delta_m\|_{\infty} \leq 1\end{aligned}\tag{A-36}$$

$$\begin{aligned}\theta &= \theta_a + \theta_1 + \dots + \theta_n \\ &= C_m X_m\end{aligned}$$

A.3.2. Mise sous forme LFT globale

En rassemblant tous les blocs et le bloc de la chaîne d'acquisition, on a dans le cas général (celui du moteur piloté en courant) :

$$\begin{aligned}\frac{dX_{el}}{dt} &= A_{el} X_{el} + B_{el} u + M_{el} e_{\Delta el} \\ \frac{dX_m}{dt} &= A_m X_m + B_m C_{el} X_{el} + M_m e_{\Delta m} \\ \frac{dX_r}{dt} &= A_r X_r + B_r C_m X_m\end{aligned}\tag{A-37}$$

$$\begin{aligned}z_{\Delta el} &= N_{el} X_{el} + O_{el} u \\ z_{\Delta m} &= N_m X_m + O_m C_{em} = N_m X_m + O_m C_{el} X_{el}\end{aligned}$$

$$y = C_r X_r + D_r \theta = C_r X_r + D_r C_m X_m$$

Pour le cas du moteur piloté directement en tension, il convient de prendre $O_{el} = 0$. C'est-à-dire finalement :

$$\frac{d}{dt} \begin{pmatrix} X_{el} \\ X_m \\ X_r \end{pmatrix} = \begin{pmatrix} A_{el} & 0 & 0 \\ B_m C_{el} & A_m & 0 \\ 0 & B_r C_m & A_r \end{pmatrix} \begin{pmatrix} X_{el} \\ X_m \\ X_r \end{pmatrix} + \begin{pmatrix} M_{el} & 0 \\ 0 & M_m \\ 0 & 0 \end{pmatrix} \begin{pmatrix} e_{\Delta el} \\ e_{\Delta m} \end{pmatrix} + \begin{pmatrix} B_{el} \\ 0 \\ 0 \end{pmatrix}$$

$$y = C_r X_r + D_r C_m X_m$$

$$\begin{pmatrix} z_{\Delta el} \\ z_{\Delta m} \end{pmatrix} = \begin{pmatrix} N_{el} & 0 & 0 \\ O_m C_{el} & N_m & 0 \end{pmatrix} \begin{pmatrix} X_{el} \\ X_m \\ X_r \end{pmatrix} + \begin{pmatrix} O_{el} \\ 0 \end{pmatrix} u \quad (\text{A-38})$$

$$\begin{pmatrix} e_{\Delta el} \\ e_{\Delta m} \end{pmatrix} = \begin{pmatrix} \Delta_{el} & 0 \\ 0 & \Delta_m \end{pmatrix} \begin{pmatrix} z_{\Delta el} \\ z_{\Delta m} \end{pmatrix}$$

Que l'on note pour simplifier :

$$\frac{dX}{dt} = AX + Bu + Me_{\Delta}$$

$$y = CX$$

$$z_{\Delta} = NX + Ou \quad (\text{A-39})$$

$$e_{\Delta} = \Delta z_{\Delta}, \quad \|\Delta\|_{\infty} \leq 1$$

On constate que la sortie incertaine w_{δ} peut être non strictement propre car dépendante de u . Pour rendre le système incertain strictement propre, il suffit d'ajouter une dynamique fictive haute fréquence en entrée du système de sorte que :

$$u = \frac{u'}{1 + \tau_F s} \Rightarrow u + \tau_F \frac{du}{dt} = u' \Rightarrow \frac{du}{dt} = \frac{u'}{\tau_F} - \frac{u}{\tau_F} \quad (\text{A-40})$$

Ce qui permet d'augmenter la représentation d'état de la nouvelle variable d'état u' :

$$\begin{aligned}
\frac{dX}{dt} &= AX + Bu + Me_{\Delta} \\
\frac{du}{dt} &= -\frac{u}{\tau_F} + \frac{u'}{\tau_F} \\
y &= CX
\end{aligned}
\tag{A-41}$$

$$z_{\Delta} = NX + Ou$$

$$e_{\Delta} = \Delta z_{\Delta}, \quad \|\Delta\|_{\infty} \leq 1$$

Soit encore :

$$\begin{aligned}
\frac{d}{dt} \begin{pmatrix} X \\ u \end{pmatrix} &= \begin{pmatrix} A & B \\ 0 & -\tau_F^{-1} \end{pmatrix} \begin{pmatrix} X \\ u \end{pmatrix} + \begin{pmatrix} M \\ 0 \end{pmatrix} e_{\Delta} + \begin{pmatrix} 0 \\ \tau_F^{-1} \end{pmatrix} u' \\
y &= (C \quad 0) \begin{pmatrix} X \\ u \end{pmatrix}
\end{aligned}
\tag{A-42}$$

$$z_{\Delta} = (N \quad 0) \begin{pmatrix} X \\ u \end{pmatrix}$$

$$e_{\Delta} = \Delta z_{\Delta}, \quad \|\Delta\|_{\infty} \leq 1$$

Annexe B

Calcul d'une FTM (1D) à partir d'un signal temporel

B.1. Principe du calcul

On considère disposer d'une acquisition temporelle d'un résiduel de stabilisation $\theta_v(t)$ produisant du flou. Le but est de déterminer une FTM (à une dimension²⁵) résultante de ce temporel pour un temps d'intégration T_i donné. Si les acquisitions temporelles ont une durée globale t_{max} , alors on peut segmenter temporellement en N_f segments à partir de chacun duquel une FTM peut être calculée.

$$N_f = \text{floor}\left(\frac{t_{max}}{T_i}\right) + 1 \quad (\text{B-1})$$

A partir de cet échantillon de N_f FTMs, on peut en déduire une FTM moyenne, une FTM majorant 70% des FTMs de l'échantillon, etc... en fonction des besoins optroniques.

Sachant que les acquisitions temporelles sont numérisées à l'échantillonnage T_e , chaque segment de durée T_i contiendra N_p points :

$$N_f = \text{floor}\left(\frac{T_i}{T_e}\right) + 1 \quad (\text{B-2})$$

Le calcul de la FTM (1D) de flou résultante consiste en la réalisation des 2 étapes suivantes :

- Calcul de la PSF 1D.

La PSF est pour l'optronique ce qu'est la réponse impulsionnelle pour l'automaticien.

On considère maintenant un des N_f segments de résiduel de stabilisation (donc de taille N_p). On pose:

²⁵ On peut montrer que le calcul d'une FTM 2D à partir de données de site et de gisement suivie d'une projection en x et y afin d'obtenir deux FTMs (une pour le site et une pour le gisement) est rigoureusement équivalent aux calculs d'une FTM 1D à partir des données d'axe site ou d'axe gisement.

$$\bar{\varepsilon} = \max(\theta_v(t)), \underline{\varepsilon} = \min(\theta_v(t)) \quad (\text{B-3})$$

On considère un vecteur de PSF de taille n . On construit n points de référence, de coordonnées spatiales uniformément espacées et comprenant l'intervalle $[\bar{\varepsilon}; \underline{\varepsilon}]$:

$$x_{ref} = [\underline{\varepsilon} - b(\bar{\varepsilon} - \underline{\varepsilon}), \dots, \bar{\varepsilon} - b(\bar{\varepsilon} - \underline{\varepsilon})] \quad (\text{B-4})$$

Où b est un coefficient entier permettant d'augmenter la résolution de la FTM (Principe d'incertitude temps-fréquence). On peut par exemple choisir $b = 4$. Usuellement on choisit $n = 100$ points.

On calcule alors la coordonnée de la matrice de PSF la plus proche de celle du point mesurée :

$$x_{ref}(i) < \theta_v[k] \leq x_{ref}(i+1), k \in [1, N_p] \Rightarrow PSF_{1D}(i) = PSF_{1D}(i) + 1, i \in [1, n] \quad (\text{B-5})$$

La PSF est finalement normalisée:

$$PSF_{1D}(i) = \frac{PSF_{1D}(i)}{\sum_{k=1}^n PSF_{1D}(k)}, i \in [1, n] \quad (\text{B-6})$$

- Calcul de la FTM 1D.

La FTM est pour l'optronique ce qu'est la transmittance pour l'automaticien.

La FTM s'obtient par le gain de la transformée de Fourier 1D de la PSF, c'est-à-dire :

$$FTM_{1D}(v) = \left| \sum_{k=1}^n e^{-2\pi j v} PSF_{1D}(k) \right| \quad (\text{B-7})$$

Annexe C

Choix du nombre d'itérations pour les tests de

Compleib

<i>nom du test</i>	<i>degré de K</i>	<i>nbre_itérations</i>	<i>nom du test</i>	<i>degré de K</i>	<i>nbre_itérations</i>	<i>nom du test</i>	<i>degré de K</i>	<i>nbre_itérations</i>
AC1	0	250	DIS1	0	500	WEC1	0	500
AC1	2	250	DIS1	2	1000	WEC1	2	1000
AC2	0	125	DIS1	4	2000	WEC1	4	2000
AC2	2	250	DIS2	0	1000	WEC1	9	4000
AC3	0	500	DIS2	2	2000	WEC2	0	500
AC3	2	1000	DIS3	0	1000	WEC2	2	1000
AC4	0	250	DIS3	2	2000	WEC2	4	2000
AC4	2	1000	DIS4	0	500	TG1	0	250
AC5	0	250	DIS4	2	1000	TG1	2	500
AC5	2	1000	DIS5	0	500	TG1	4	1000
AC6	0	500	DIS5	2	1000	AGS	0	250
AC6	2	2000	EB1	0	250	AGS	2	250
AC6	4	2000	EB1	2	250	AGS	4	250
AC7	0	250	EB1	4	250	PAS	0	500
AC7	3	1500	EB2	0	250	PAS	2	1000
AC7	5	1500	EB2	2	250	TMD	0	250
AC8	0	500	EB2	4	250	TMD	2	250
AC8	3	1500	EB3	0	250	TMD	4	500
AC8	5	1500	EB3	2	250	CM1	0	250
AC9	0	1000	EB3	4	250	CM1	2	250
AC9	3	1500	NN1	0	250	CM1	7	250
AC9	5	2000	NN1	2	250	IH	0	10000
AC9	8	3000	NN2	0	250	IH	2	10000
AC10	0	500	NN2	2	250	IH	8	10000
AC10	3	10000	NN3	0	10000	IH	10	10000
AC10	10	10000	NN3	2	10000	IH	15	10000
AC10	20	10000	NN4	0	250	TF1	0	250
AC11	0	500	NN4	2	250	TF1	2	1000

AC11	2	500	NN5	0	500	TF1	4	2000
AC12	0	500	NN5	2	1000	TF2	0	250
AC12	2	1000	NN5	4	2000	TF2	2	250
AC13	0	500	NN6	0	500	TF2	4	250
AC13	2	500	NN6	2	1000	TF3	0	500
AC13	7	1000	NN6	4	2000	TF3	2	1000
AC13	10	1500	NN7	0	500	TF3	4	2000
AC13	20	2000	NN7	3	1000	CSE1	0	250
AC14	0	500	NN7	5	2000	CSE1	2	250
AC14	2	500	NN8	0	250	CSE1	8	250
AC14	10	1000	NN8	3	250	CSE2	0	250
AC15	0	250	NN9	0	250	CSE2	2	250
AC15	2	500	NN9	3	2000	CSE2	10	250
AC16	0	250	NN11	0	250	UWV	0	250
AC16	2	250	NN11	2	1000	UWV	2	250
AC17	0	250	NN11	6	4000	UWV	4	250
AC17	2	250	NN12	0	1000	EB1	0	250
AC18	0	1000	NN12	2	2000	EB1	2	250
AC18	2	1500	NN13	0	500	EB1	4	250
AC18	5	500	NN13	2	1000	EB2	0	250
HE1	0	4000	NN14	0	1000	EB2	2	250
HE1	2	4000	NN14	2	2000	EB2	4	250
HE2	0	4000	NN15	0	250	EB3	0	250
HE2	2	4000	NN15	2	250	EB3	2	250
HE3	0	4000	NN16	0	250	EB3	4	250
HE3	2	4000	NN16	2	1500	CM1	0	250
HE3	4	4000	NN16	4	3000	CM1	3	250
HE4	0	4000	NN17	0	250	CM1	8	250
HE4	2	4000	NN17	2	250	CM2	0	250
HE4	4	4000	ROC1	1	500	CM2	4	250
HE5	0	4000	ROC1	3	1000	CM2	11	250
HE5	2	4000	ROC1	5	2000	CM3	0	250
HE5	4	4000	ROC2	1	500	CM3	5	250
HE6	0	4000	ROC2	3	1000	CM3	14	500
HE6	2	4000	ROC2	5	2000	CM4	0	250
HE6	8	4000	ROC3	1	10000	REA1	0	250
HE6	15	4000	ROC3	3	10000	REA1	2	250
HE7	0	4000	ROC3	5	10000	REA2	0	250
HE7	2	4000	ROC4	1	500	REA2	2	250
HE7	8	4000	ROC4	3	1000	REA3	0	250
HE7	10	4000	ROC4	5	2000	REA3	2	250
HE7	15	4000	ROC5	1	500	REA3	5	250
JE1	0	500	ROC5	3	1000	REA3	10	250
JE1	2	3000	ROC5	5	2000			
JE1	8	3000	HF2D10	0	250			
JE2	0	2000	HF2D10	2	500			

JE2	2	2000	HF2D11	0	250
JE2	8	2000	HF2D11	2	500
JE2	10	2000	HF2D12	0	250
JE2	15	2000	HF2D12	2	500
JE3	0	250	HF2D13	0	250
JE3	2	1000	HF2D13	2	500
JE3	8	2000	JE3	10	3000

Bibliographie

- [APK 02] APKARIAN P., NOLL D., « Nonsmooth H_∞ synthesis », *Journal of class files*, vol. 1, n° 11, novembre 2002.
- [APK 03] APKARIAN P., NOLL D., TUAN H.D., « Fixed-order H_∞ control design via a partially augmented lagrangian method », *International Journal of Robust and Nonlinear Control*, vol. 13, Issue 12, p. 1 137-1 148, octobre 2003.
- [APK 10] APKARIAN P., « Comparison of HINFSTRUCT Matlab Robust Control Toolbox R2010b and HIFOO 3.0 with HANSO 2.0 », <http://pierre.apkarian.free.fr>, 2010.
- [APK 11] APKARIAN P., « Non-smooth μ -synthesis », *Int. Journal of Robust and Nonlinear Control*, vol. 21, Issue 13, p. 1493-1508, sept. 2011
- [APK 12] APKARIAN P., « Tuning controllers for multiple models or operating conditions », <http://pierre.apkarian.free.fr>
- [BIR 03] BIRBIL S.I., FANG S.C., « An Electromagnetism-like Mechanism for Global Optimization », *Journal of Global Optimization*, Kluwer Academic Publishers, Vol. 25, p 263-282, 2003.
- [BOY 94] BOYD S., EL GHAOU L., FÉRON E., BALAKRISHNAN V., *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Publications, Philadelphia, 1994.
- [BRA 07] BRATON D., KENNEDY J., « Defining a standard for Particle Swarm Optimization », *IEEE Swarm Intelligence Symposium 2007*, p 120-127, Avril 2007.
- [BUR 06] BURKE V., HENRION D., LEWIS A.S., OVERTON M.L., « Hifoo - a matlab package for fixed-order controller design and H_∞ optimization », *Proceedings of the IFAC Symposium on Robust Control Design*, 2006.
- [CHI 96] CHIPPERFIELD A.J., DAKEV N.V., FLEMING P.J., WHIDBORNE J.F., « Multiobjective robust control using evolutionary algorithm », *IEEE Int. Conf. on Indus.Technology*, p. 269-273, Décembre 1996.
- [CLE 02] CLERC M., KENNEDY J., « The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space », *IEEE Transactions on Evolutionary Computation*, vol. 6, p 58-73, 2002.
- [CLE 04] CLERC M., Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particuliers », *J3eA*, Vol. 3 – 7, 2004.
- [CLE 06] CLERC M., *Particle Swarm Optimization*, ISTE Ltd, London, 2006.

- [CLE 12] CLERC M., « Standard Particle Swarm Optimization », In <http://clerc.maurice.free.fr/psa>, 2012
- [DAL 08] DALESSIO J., SHUSTER E., HUMPHREYS D.A., WALKER M.L., IN Y., KIM J.S., « Robust Control of Resistive Wall Modes in Tokamak Plasmas using μ -synthesis », *17th IFAC World Congress*, Seoul, Korea, 6-11 Juillet, 2008.
- [DOY 89] DOYLE J.C., GLOVER K., KHARGONEKAR P.K., FRANCIS B.A., « State-Space Solutions to Standard H_2 and H_∞ Control Problems », *IEEE Trans. Autom. Control*, AC 34, n° 8, p. 831-846, 1989.
- [DRE 06] DRÉO J., PÉTROWSKI A., SIARRY P., TAILLARD E., *Metaheuristics for Hard Optimization. Methods and Case Studies*, Springer-Verlag, Berlin, 2006.
- [EBE 01] EBERHART R.C, SHI Y., « Particle Swarm Optimization: Development Applications and Resources », *IEEE Proceedings Congress on Evolutionary Computation*, vol.1, p. 81-86, Mai 2001.
- [ELG 94] EL GHAOU L., BALAKRISHNAN V., « Synthesis of fixed-structure controllers via numerical optimization », *IEEE Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, p. 2678-2683, Décembre 1994.
- [ELG 97] EL GHAOU L., OUSTRY F., AITRAMI M., « A Cone Complementarity Linearization Algorithm for Static Output-Feedback and Related Problems », *IEEE Trans. Autom. Control*, vol. 42, n° 8, août 1997.
- [FEY 13a] FEYEL P., DUC G., SANDOU G., « Fixed-Order H_∞ Loop-shaping synthesis: a Time Domain Approach », *IEEE Symposium on Computational Intelligence in Control and Automation*, 2013.
- [FEY 13b] FEYEL P., DUC G., SANDOU G., « Enhancing Optimal weight tuning in H_∞ loop-shaping control with particle swarm optimization », *5th international conference on Evolutionary Computation Theory and Applications*, Vilamoura, Portugal, Septembre 2013.
- [FEY 13c] FEYEL P., *La commande robuste par loop-shaping*, Editions-Hermès Science, 2013.
- [FEY 14a] FEYEL P., DUC G., SANDOU G., « Evolutionary Fixed-Structure H_∞ Synthesis for Multiple Plants », *IEEE MultiConference on Systems and Control*, Antibes, France, Octobre 2014.
- [FEY 14b] FEYEL P., DUC G., SANDOU G., « μ -synthesis with dynamic D -Scalings using Quantum Particle Swarm Optimization », *IEEE 13th European control Conference*, Strasbourg, France, Juin 2014.
- [FEY 14c] FEYEL P., DUC G., SANDOU G., « Evolutionary Fixed-Structure μ -Synthesis », *CICA & SDE 2013, IEEE Symposium on Computational Intelligence in Control and Automation*, Orlando, Décembre 2014.
- [FIS 12] FISTER I. JR, YANG X. S, FISTER I., BREST J., FISTER D., «A Brief Review of Nature-Inspired Algorithms for Optimization », *Neural and Evolutionary Computing, Elektrotehniski Vestnik*, p. 1-7, 2013.
- [GAH 11] GAHINET P., APKARIAN P., « Decentralized and Fixed-Structure H_∞ Control in MATLAB », *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, Etats-Unis, 12-15 décembre 2011.

- [GAH 94] GAHINET P., APKARIAN P., « A linear matrix inequality approach to H_∞ control », *Int. J. of Robust & Nonlinear Contr.*, vol. 4, p. 421-448, 1994.
- [GLO 89] GLOVER K., MCFARLANE D., « Robust Stabilization of Normalized Coprime Factor Plant Description with H_∞ -Bounded Uncertainty », *IEEE Trans. Autom. Control*, AC 34 n° 8, p. 821-830, 1989.
- [HIL 08] HILKERT J.M., « Inertially Stabilized Platforms Technology », *IEEE Control Systems Magazine*, 28(1):26-46, 2008.
- [HIR 12] HIRWA S., FEYEL P., DUC G., SANDOU G., « Reduced Order Controllers for Inertial Line Of Sight Stabilization », *Proceedings of the 7th IFAC Symposium on Robust Control Design*, Aalborg, Danemark, 20-22 juin 2012.
- [HOL 06] HOLST G.C., *Electro-optical imaging system performance*, SPIE press, JCD Publishing, 2006
- [HOY 91] HOYLE D.J., HYDE R.A., LIMEBEER D.J.N., « An H_∞ approach to two degree of freedom design », *30th IEEE Conference on Decision & Control*, Brighton, Angleterre, décembre 1991.
- [ICH 08] MARUTA I., KIM T-H., SUGIE T., « Synthesis of fixed-structure H_∞ controllers via Constrained Particle Swarm Optimization », *17th IFAC Control Conference*, Séoul, Corée, 2008.
- [KAR 07] KARABOGA D., BASTURG B., « A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm », *Journal of Global Optimization*, Springer, Vol. 39, Issue 3, p. 459-471, Avril 2007.
- [KAR 08] KARABOGA D., BASTURG B., « On the performance of artificial bee colony (ABC) algorithm », *Journal Applied Soft Computing*, Elsevier, Vol. 8, Issue 1, p. 687-697, Janvier 2008.
- [KEN 95] KENNEDY J., EBERHART R.C., « Particle Swarm Optimization », *IEEE International Conference on Neural Networks*, Perth, Australia, p. 1942-1948, 1995.
- [KWA 72] KWAKERNAAK H., SIVAN R., *Linear Optimal Control Systems*, Wiley-Interscience, New-York, 1972.
- [LAR 12] LARY A., KHORSRAVI A., KARAMNEZHAD R., « A New Approach to Design a Simple Structure μ Synthesis Controller: Application to an Uncertain DC Motor », *7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, 20-22 juin, 2012
- [LEE 05] LEE K. S., GEEM Z.W., « A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice », *Computer Methods in Applied Mechanics and Engineering*, Elsevier, p. 3902–3933, 2005.
- [LEI 03] LEIBFRITZ F., « COMpleib: CONstrained Matrix–optimization Problem library – a collection of test examples for nonlinear semidefinite programs, control system design and related problems », *Technical report*, Universität Trier, 2003.
- [LEW 12] LEWIS A.S., OVERTON M.L., « Nonsmooth optimization via quasi-Newton methods », *Mathematical Programming: A publication of the mathematical Optimization Society*, Springer-Verlag, 2012.
- [LIM 93] LIMEBEER D.J.N., KASANELLY E.M., PERKINS J.D., « On the design of robust two degree of freedom controllers », *Automatica*, vol. 29, n° 1, p. 157-168, 1993.

- [LIN 05] LIN C.C., CHOU Y.S., « A Novel Algorithm for Real Mu Synthesis », *IEEE ICSS2005 International Conference On Systems & Signals Society*, p.734-739, 2005.
- [LUC 09] LUSZCZEK P., « Parallel programming in Matlab », www.mathworks.com, Juillet 2009.
- [MAS 08] MASTEN M.K., « Inertially Stabilized Platforms for Optical Imaging Systems », *IEEE Control Systems Magazine*, 28(1):47-64, 2008.
- [MAT 10] MATHWORKS, « Matlab Parallel Computing Toolbox », www.mathworks.com.
- [MAY 99] MAYZONETTE J.L., *Base de radiométrie optique*, Editions Cépaduès, 1999.
- [MCF 90] MCFARLANE D., GLOVER K., « Robust Controller Design Using Normalized Coprime Factor Plant Descriptions », *Lecture Notes in Control and Information Sciences*, Springer Verlag, Berlin, 1990.
- [MCF 92] MCFARLANE D., GLOVER K., « A Loop Shaping Design Procedure Using H_∞ Synthesis », *IEEE Trans. Autom. Control*, AC 37 n° 6, p. 759-769, 1992.
- [MCK 96] MCKELVEY T., HELMERSSON A., « State-space parametrizations of multivariable linear systems using tridiagonal matrix forms », *35th IEEE Decision and Control Conf.*, vol. 4, p. 3 654-3 659, décembre 1996.
- [MEZ 08] MEZURA-MONTES E., REYES-SIERRA M., COELLO COELLO C.A., « Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art », *Advances in Differential Evolution Studies in Computational Intelligence*, Vol. 143, p. 173-196, 2008.
- [MIY 97] MIYAMOTO S., VINNICOMBE G., « Fixed Structure H_∞ Control Design for Multiple Plant. A Coprime Factorisation LMI Based Approach », *European Control Conf.*, Bruxelles, 1997.
- [NEL 65] NELDER J.A., MEAD R., « A simplex method for function minimization », *Computer Journal*, Vol. 7, p. 308-313, 1965.
- [PRE 04] PREMPAIN E., « Static H_∞ loop-shaping control of a fly-by-wire helicopter », *Conference on Decision & Control*, p. 1 188-1 195, vol. 2, 2004.
- [PRE 05] PREMPAIN E., « A comparison of two algorithms for the static H_∞ loop-shaping problem », *16th IFAC World Congress*, vol. 16, part 1, République Tchèque, 2005.
- [PRI 05] PRICE K., STORM K., LANPINEN J.A., *Differential Evolution – A Practical Approach to Global Optimization*, Springer-Verlag, Berlin, 2005.
- [POP 10] POPOV A., « Methods for Fixed-Structure H_∞ Controller Synthesis – A Comparison », *Technical Report*, Institute of Control Systems, Hamburg, 2010.
- [ROC 94] ROCKAFELLAR T., « Nonsmooth Optimization », dans J.R. Birge, K.G. Murty (dir.), *Mathematical Programming: State of the Art 1994*, p. 248-258, University of Michigan Press, Ann Arbor, 1994.
- [SAN 10] SANDOU G., DUC G., BEAUVOIS D., « Optimisation par essaim particulière du réglage d'un correcteur H_∞ », *Conférence Internationale Francophone d'Automatique*, CIFA 2008, Bucarest, Roumanie, 2008

- [SCH 97] SCHERER C., GAHINET P., CHILALI M. « MultiObjective Output-Feedback Control via LMI Optimization », *IEEE Transactions on Automatic Control*, vol. 42, n° 7, p. 896-911, Juillet 1997.
- [SED 09] SEDIGHIZADEH D., MASEHIAN E., « Particle Swarm Optimization Methods, Taxonomy and Applications », *International Journal of Computer Theory and Engineering*, vol. 1, n° 5, p. 1 793-8 201, décembre 2009.
- [SCH 97] SCHERER C., GAHINET P., CHILALI M., « Multiobjective Output-feedback control via LMI optimisation », *IEEE Trans. Autom. Contr.*, AC 42 n° 7, p. 896-911, 1997.
- [SIA 02] SIARRY P., « Application des métaheuristiques d'optimisation en électronique », *Techniques de l'Ingénieur*, RE 8 - 1, octobre 2002.
- [SKO 01] SKOGSTAD S., POSTLETHWAITE I., *Multivariable feedback control. Analysis and design*, 2^e édition, John Wiley & Sons, New York, 2001.
- [STO 95] STORM R. M, PRICE K., « Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces », *Technical Report TR9 -012, International Computer Science Institute, Berkeley, California*, 1995.
- [STO 96] STORM R. M., « On the usage of Differential Evolution for Function Optimization », *North American Fuzzy Information Processing Society*, p. 519-523, 1996
- [SUN 12a] SUN J., LAI C.H., WU X.J, *Particle Swarm Optimization : Classical and Quantum Perspectives*, CRC Press, London, 2012
- [SUN 12b] SUN J., WU X.J, PALADE V., FANG W., LAI C.H., XU W., « Convergence analysis and improvements of quantum-behaved particle swarm optimization », *Information Sciences*, vol. 193, pp 81–103, 2012.
- [TOF 95] TOFFNER-CLAUSEN S., ANDERSEN P., SOUSTRUP J., NIEMANN H.H., « A new approach to μ -synthesis for mixed perturbation sets », *3rd European Control Conference*, Rome, p 147–152, 1995.
- [TSA 02] TSAI K. Y., HINDI H. A., « DQIT : μ -synthesis without D-scale fitting », *Proceedings of the American Control Conference*, Anchorage, p 493–198, 2002.
- [OZT 12] OZTURK C., KARABOGA D., GORKEMLI B., « Artificial bee colony algorithm for dynamic deployment of wireless sensor networks », *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 20, Issue 2, p. 255-262, Janvier 2012.
- [VID 11] VIDYASAGAR M., *Control system synthesis. A factorisation Approach*, MIT Press, Cambridge, MA, 1985.
- [VIN 97] VINNICOMBE G., MIYAMOTO S., « On Reduced Order H_∞ Loop-Shaping Controllers: A Design Method and Examples of Local Optima », *European Control Conference*, Bruxelles, 1-4 juillet 1997.
- [VIN 00] VINNICOMBE G., *Uncertainty and feedback - H_∞ loop-shaping and the v-gap metric*, Imperial College Press, London, 2000.
- [YAN 09a] YANG X.S., DEB S, « Cuckoo Search Via Levy Flight », *Proc. of World Congress on Nature & Biologically Inspired Computing*, India. IEEE Publications, p. 210-214, USA, Décembre 2009

- [YAN 09b] YANG X.S., « Firefly Algorithms for Multimodal Optimization », *Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Science*, Springer, Vol. 5792, p. 169-178, 2009
- [YAN 10a] YANG X.S., *Nature-Inspired Metaheuristic Algorithms*, Lunivers Press, Cambridge, 2010
- [YAN 10b] YANG X.S., « A new metaheuristic bat-inspired algorithm » *Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence*, Springer, Vol. 284, p. 65-74, 2010.
- [YOU 90] YOUNG P., DOYLE J., « Computation of μ with real and complex uncertainties », *29th IEEE Conference on Decision and Control*, Honolulu, Hawaii, p. 1230-1235, 1990.
- [YOU 93] YOUNG P., « Controller Design with Real Parametric Uncertainty », *Technical Memorandum*, Control and Dynamical Systems, California Institute of Technology, Pasadena, 1993.
- [WAN 12] WANG G., GUO L., DUAN H., WANG H., LIU L., SHAO M., « A Hybrid Metaheuristic DE/CS Algorithm for UCAV Three-Dimension Path Planning », *The Scientific World Journal (www.hindawin.com)*, 2012
- [ZHO 96] ZHOU K., DOYLE J.C., GLOVER K., *Robust and Optimal Control*, Prentice-Hall, Englewood Cliffs, NJ, 1996.