



**HAL**  
open science

# Méthodologies de réglage automatique temps-réel de lois de pilotage

Lilian Ronceray

► **To cite this version:**

Lilian Ronceray. Méthodologies de réglage automatique temps-réel de lois de pilotage. Automatique / Robotique. ISAE, 2009. Français. NNT: . tel-00430820

**HAL Id: tel-00430820**

**<https://theses.hal.science/tel-00430820>**

Submitted on 10 Nov 2009

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

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

# Méthodologies de réglage automatique temps-réel de lois de pilotage

Rapport de soutenance de thèse de Lilian Ronceray

le mardi 2 juin 2009



# Résumé

Cette thèse s'inscrit dans le cadre du réglage des lois de pilotage d'un avion de transport civil en phase de développement. De par la représentativité limitée des modèles utilisés pour leur réglage initial, certaines lois de pilotages requièrent plusieurs itérations en essais en vol avant d'atteindre un niveau de performance satisfaisant. Il s'agit d'un processus long et coûteux que l'industriel cherche à raccourcir autant que possible. En partant du constat que les données issues des essais en vol sont peu utilisées par rapport aux modèles, nous avons élaboré des méthodologies permettant une exploitation utile de ces données à des fins de réglage de lois de commande.

L'avion étant un système à paramètres variants, il faut alors tenir compte du fait que ces données ne sont valables que pour une petite région de son domaine de vol. Par conséquent, les modifications apportées sur les lois de commande, interpolées à l'aide des paramètres du domaine de vol, devront être répercutées uniquement dans cette petite région. Nous avons envisagé deux cas : celui d'un travail de retouche hors ligne (i.e. effectué en bureau d'études) et celui d'un réglage en ligne (i.e. directement sur l'avion pendant un vol d'essai).

Ces deux aspects d'interpolation et de données locales nous ont fait choisir une méthode d'apprentissage locale pour interpoler les lois de commande. Il s'agit d'une forme de réseau de neurones appelée *mélange d'experts* dont la structure permet de diviser le domaine de vol en plusieurs sous-domaines et d'affecter à chacun d'eux une modélisation locale.

Nous avons ensuite élaboré une méthodologie générique permettant de générer les données à interpoler, de définir la structure du mélange d'experts et d'effectuer son apprentissage à la fois dans un cadre hors ligne et en ligne. L'aspect génération de données constitue une part importante de la méthodologie. En effet, dans le cadre d'une loi de commande, l'obtention des données amenées à être interpolées n'est pas triviale. Nous proposons alors une méthode de synthèse basée sur l'inversion de données temporelles, pour le cas de lois feedforward. La structuration du mélange d'experts se compose du choix d'une subdivision optimale du domaine de vol, d'une sélection du type de la modélisation locale et de la gestion des pondérations affectées à chacun des modèles locaux.

La méthodologie a fait l'objet de deux applications identifiées par Airbus comme reflétant le problème posé. La première concerne l'estimation du dérapage aérodynamique de l'avion et la seconde une loi feedforward de compensation d'aérofreins. Ces exemples montrent comment la méthode a permis de systématiser un processus de réglage fonctionnant jusque là « à tâtons ».

**Mots clés :** apprentissage local, mélange d'experts, simulation inverse, commande adaptative, commandes de vol.



# Abstract

This thesis deals with the tuning of flight control laws for civilian carrier aircraft, during its development phase. Due to the limited representativeness of the models used for the initial tuning, flight control laws may require several flight test iterations to achieve satisfactory performance. It is a long and costly process and the aircraft manufacturer thus wants to shorten it. As we have noticed that flight test data are less used than models, we have designed methodologies allowing a better use of these data for a control law tuning purpose.

An aircraft being a parameter varying system, we must keep in mind that these data are valid in a small region of the flight domain only. As a consequence, the modifications made to the control laws, interpolated using parameters of the flight domain, must be passed on solely in this region. We considered two cases : an offline retuning work (i.e. performed in design office) and an online tuning (i.e. directly on the aircraft during a test flight).

Both the notions of interpolation and local data made us choose a local learning method in order to interpolate the control laws. This method is a form of neural networks known as “mixture of experts”. Its structure will allow us to divide the flight domain into several sub-domains, to which we will allocate a local model.

Therefore, we designed a generic methodology allowing to generate data to be interpolated, to define the mixture of experts structure and to perform a learning task on it. The data generation is a significant part of this methodology and considering the interpolation of control laws, it is not trivial. We thus propose an inversion based synthesis method for feedforward control laws. The mixture of experts structuring consists in choosing an optimal meshing of the flight domain, selecting the type of local model and managing the weightings applied to each local model.

The methodology has been tested on two applications, pointed out by Airbus as reflecting the issue at stake. The first one deals with the estimation of the aircraft aerodynamic sideslip and the second one deals with a feedforward control law compensating for the side-effects yielded by the airbrakes extension. These examples show how the proposed methodology helps systematizing a tuning process which has been until then tedious.

**Keywords :** local learning, mixture of experts, inverse simulation, adaptive control, flight control laws.



# Table des matières

Notations et glossaire	ix
Liste des publications	xi
Introduction	1
<b>I Modélisation locale pour la performance globale</b>	<b>7</b>
<b>1 Apprentissage</b>	<b>9</b>
1.1 Définition de l'apprentissage . . . . .	9
1.2 Formulations générales du problème d'apprentissage . . . . .	13
1.3 Conclusion . . . . .	15
<b>2 Mélanges d'experts</b>	<b>17</b>
2.1 Réseaux RBF . . . . .	17
2.2 Réseaux RBF généralisés . . . . .	19
2.3 Mélange d'experts . . . . .	23
2.4 Conclusion . . . . .	27
<b>II Méthodologies et outils pour la retouche</b>	<b>29</b>
<b>3 Outils pour la génération des données d'apprentissage</b>	<b>37</b>
3.1 Formalisation du problème . . . . .	37
3.2 Inversion . . . . .	38
3.3 Identification . . . . .	55
3.4 Conclusion . . . . .	66
<b>4 Outils pour l'apprentissage</b>	<b>67</b>
4.1 Stratégies d'apprentissage hors ligne . . . . .	67
4.2 Stratégies d'apprentissage en ligne . . . . .	74
4.3 Conclusion . . . . .	79
<b>5 Régularisation</b>	<b>81</b>
5.1 Notion de problème mal-posé . . . . .	81
5.2 Cas d'une solution discrète . . . . .	82
5.3 Cas d'une solution continue . . . . .	82



5.4	Détermination du paramètre de régularisation . . . . .	85
5.5	Conclusion . . . . .	89
<b>III</b>	<b>Applications</b>	<b>91</b>
<b>6</b>	<b>Présentation de l'avion</b>	<b>93</b>
6.1	Repères et notations . . . . .	93
6.2	Mécanique du vol . . . . .	97
6.3	Coefficients aérodynamiques . . . . .	101
<b>7</b>	<b>Estimateur de dérapage</b>	<b>105</b>
7.1	Génération de données . . . . .	107
7.2	Apprentissage . . . . .	112
7.3	Conclusion . . . . .	124
<b>8</b>	<b>Compensation de spoilers</b>	<b>125</b>
8.1	De l'utilisation des spoilers . . . . .	125
8.2	Expression du besoin . . . . .	128
8.3	Génération de données . . . . .	132
8.4	Apprentissage . . . . .	142
8.5	Conclusion . . . . .	148
<b>IV</b>	<b>Conclusion et perspectives</b>	<b>149</b>
<b>9</b>	<b>Conclusion et perspectives</b>	<b>151</b>
	<b>Annexes</b>	<b>157</b>
<b>A</b>	<b>Rappels mathématiques</b>	<b>157</b>
A.1	Algèbre . . . . .	157
A.2	Analyse . . . . .	158
<b>B</b>	<b>Perceptrons multi-couches</b>	<b>161</b>
B.1	Description générale . . . . .	161
B.2	Algorithme de rétro-propagation . . . . .	162
B.3	Généralisation . . . . .	165
B.4	Conclusion . . . . .	166
<b>C</b>	<b>Inversion et synthèse simultanées</b>	<b>167</b>
<b>D</b>	<b>Vérification de la propriété de la matrice d'observation du filtre en treillis normalisé</b>	<b>169</b>
	<b>Bibliographie</b>	<b>173</b>

# Notations et glossaire

## Notations

$x, \theta$	scalaire
$\mathbf{x}, \boldsymbol{\theta}$	vecteur
$\mathcal{X}$	espace vectoriel
$X, \Theta$	matrice
$X^T$	matrice transposée
$x^*$	nombre complexe conjugué
$X^*$	matrice transposée conjuguée complexe
$X^{-1}$	matrice inverse
$X^\dagger$	matrice pseudo-inverse
$\mathbf{x}, X$	transformée de Fourier discrète d'un vecteur, d'une matrice
$\hat{\mathbf{x}}$	grandeur vectorielle estimée
$\tilde{X}$	grandeur matricielle augmentée
$\text{Im } X$	image d'une matrice
$\text{Ker } X$	noyau d'une matrice
$\text{vect}$	opérateur de vectorisation de matrice par concaténation des vecteurs colonnes
$\text{diag}(\mathbf{x})$	matrice diagonale dont les éléments sont ceux du vecteur $\mathbf{x}$
$*$	produit de convolution (temps continu)
$\otimes$	produit de convolution discret
$\otimes$	produit de Kronecker
$\langle \cdot, \cdot \rangle$	produit scalaire
$\mathcal{J}$	critère d'optimisation
$s$	variable de Laplace
$\mathcal{C}^\infty$	Ensemble des fonctions indéfiniment dérivables

## Glossaire

A/F	AéroFreins
ARMA	Auto-Regressive Moving Average
CAS	Calibrated AirSpeed
CFD	Computational Fluid Dynamics
IRS	Inertial Reference System
LMS	Least Mean Squares
LQG	Linéaire Quadratique Gaussien
LTI	Linéaire à Temps Invariant

LTV	Linéaire à Temps Variant
MC	Moindres Carrés
MCR	Moindres Carrés Récursifs
MEX	Mélange d'EXperts
NLMS	Normalized Least Mean Squares
PA	Pilote Automatique
PV	Paramètre Variant
RBF	Radial-Basis Functions
RI	Réponse Impulsionnelle
RIF	Réponse Impulsionnelle Finie
SIMO	Single Input Multiple Output
SPGL	Simplified Partial Gradient Lattice algorithm

# Liste des publications

- [Ronceray *et al.*, 2007a] L. Ronceray, M. Jeanneau, D. Alazard, P. Mouyon et S. Tebbani  
A Local learning approach to real-time parameter estimation –  
Application to an aircraft  
*4<sup>th</sup> INSTICC International Conference on Informatics in Control,  
Automation and Robotics*, 2007
- [Ronceray *et al.*, 2007b] L. Ronceray, M. Jeanneau, D. Alazard, P. Mouyon et S. Tebbani  
Real-time automatic tuning of an aircraft sideslip estimator  
*17<sup>th</sup> IFAC Symposium on Automatic Control in Aerospace*, 2007
- [Ronceray *et al.*, 2007c] L. Ronceray, M. Jeanneau, D. Alazard, P. Mouyon et S. Tebbani  
Réglage automatique temps-réel de l'estimateur de dérapage  
d'un avion de transport civil  
*2<sup>ème</sup> Journées Nationales des Doctorants MACS*, 2007
- [Ronceray *et al.*, 2008] L. Ronceray, D. Alazard, P. Mouyon, G. Puyou et S. Tebbani  
Aircraft spoilers compensation design using iterative inversion  
*17<sup>th</sup> IFAC World Congress*, 2008
- [Ronceray *et al.*, 2009] L. Ronceray, P. Mouyon, S. Tebbani, G. Puyou et D. Alazard  
Adaptive retuning of feedforward controller – Application to the  
airbrake compensation of an aircraft  
Accepté à l'*European Control Conference*, 2009



# Introduction

## Contexte

LORS du développement d'un nouvel avion, les essais en vol sont une phase critique qui court du premier vol à la livraison à la première compagnie cliente. L'objectif de cette phase est triple. Tout d'abord, il s'agit d'*identifier* l'avion en termes de coefficients aérodynamiques, son comportement dynamique ainsi que ses performances afin de mettre à jour les modèles prévisionnels, élaborés à partir de données d'essais de soufflerie et de calculs de CFD. Ensuite, il faut *régler* finement les commandes de vol manuelles et le pilote automatique afin de remplir au mieux un cahier des charges multi-objectifs (qualités de vol, charges, confort). Enfin, il s'agit de *démontrer* en vol, avant sa commercialisation et auprès des autorités de certification, les qualités de l'avion, telles que ses performances, son rayon d'action, la fiabilité de ses systèmes, sa robustesse à d'éventuelles pannes. On pourra noter que si l'enchaînement de ces tâches suit une logique certaine, il n'est pas pour autant réalisé de façon séquentielle.

S'il est vrai que les techniques et les moyens de modélisation sont en constante progression et permettent l'obtention de modèles de plus en plus précis, on constate toujours des différences de comportement notables entre ceux-ci et l'avion réel. Cela implique que les lois de pilotage initiales soient robustes en stabilité à ces écarts puis ajustées sur le comportement vrai de l'avion jusqu'à atteindre un niveau de performance satisfaisant conformément au cahier des charges. Cette façon de procéder, présentée sur la partie gauche de la figure 1, peut s'avérer longue et coûteuse puisqu'elle nécessite un certain nombre d'itérations : identification de l'avion en vol, recalage des modèles, réglage des lois en bureau d'étude sur la base des nouveaux modèles, validation sur simulateurs puis en vol, ce schéma étant itéré jusqu'à ce que la loi de pilotage soit jugée satisfaisante.

Ce processus est exclusivement construit sur une mise en série de l'*identification* et de la *commande*, ce qui s'avère pénalisant en terme de délais de mise au point des lois de pilotage. Il faut également souligner que les objectifs de l'*identification* ne sont pas uniquement liés à la *commande* : les modèles identifiés viennent également alimenter les simulateurs d'entraînement, peu avant la mise en service de l'avion. Les modèles les plus représentatifs ne sont donc pas disponibles pendant la première moitié des essais en vol. Or, il est important que les lois soient réglées le plus tôt possible afin de garantir une maturité suffisante à l'entrée en service de l'avion. A titre d'exemple, la figure 2 montre un profil type du nombre d'heures d'essais en vol consacrées à l'identification et au réglage des lois d'un avion en développement.

Nous pouvons constater qu'au début de la période d'essais en vol, la majeure partie est consacrée aux vols d'identification alors que l'activité de loi de pilotage est au repos. Durant cette période, l'avion vole à l'aide d'une loi spécifique en boucle ouverte. Une fois ces essais terminés, la phase dédiée au réglage des lois peut reprendre. Cet enchaînement

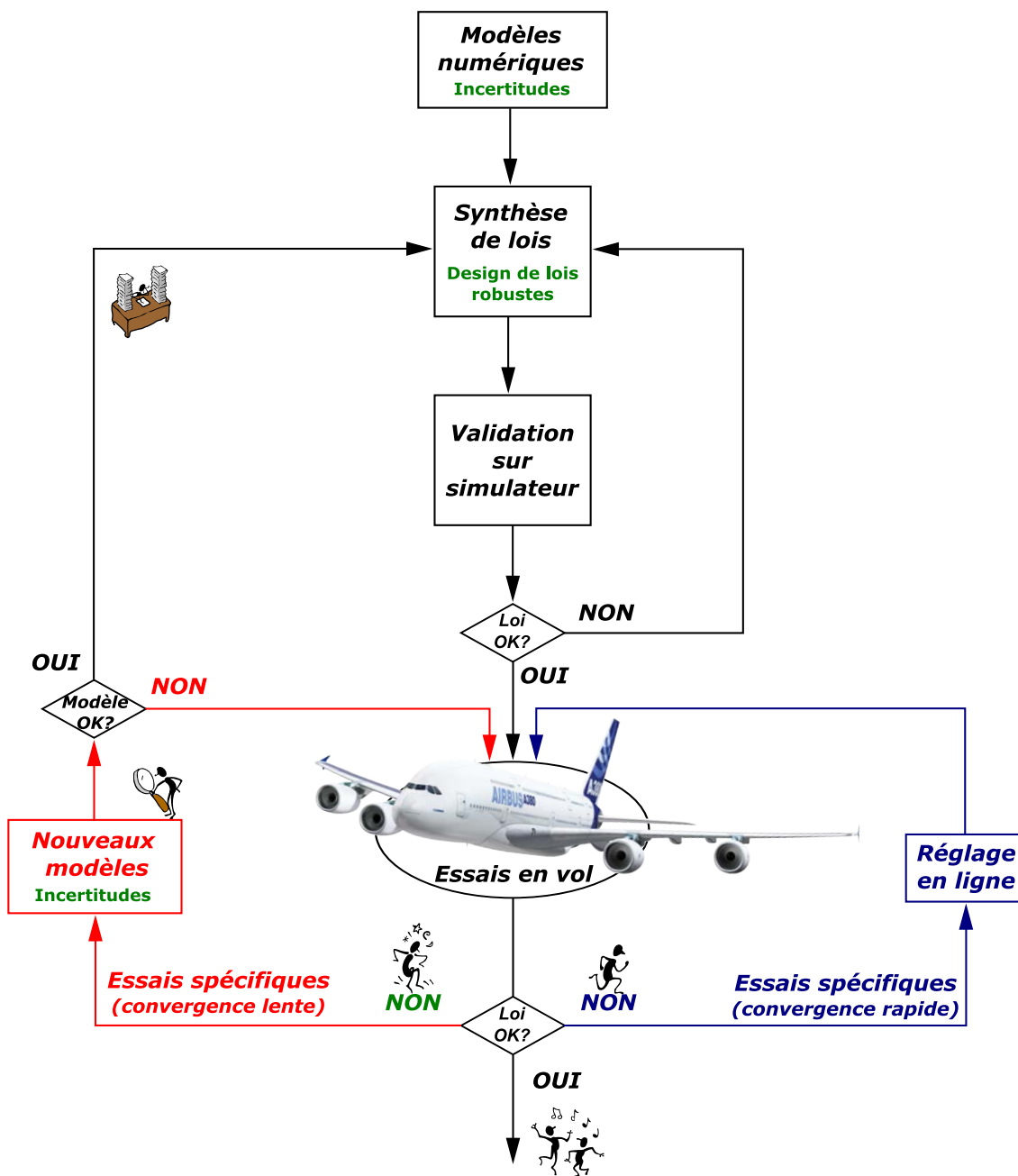


FIG. 1: Intérêt du réglage en ligne

séquentiel amène donc une période creuse pour le réglage des lois, qu'il serait souhaitable de combler.

Une première stratégie déjà mise en œuvre dans certains cas, consiste à retoucher les lois par l'intermédiaire d'une analyse des essais en vol [Gimonet, 2003] et par une évaluation des tendances obtenues par les modèles, sans attendre la fin de la campagne d'*identification*. Cette méthode donne toujours un poids fort aux modèles dans le réglage des lois mais peut réduire les délais de mise au point.

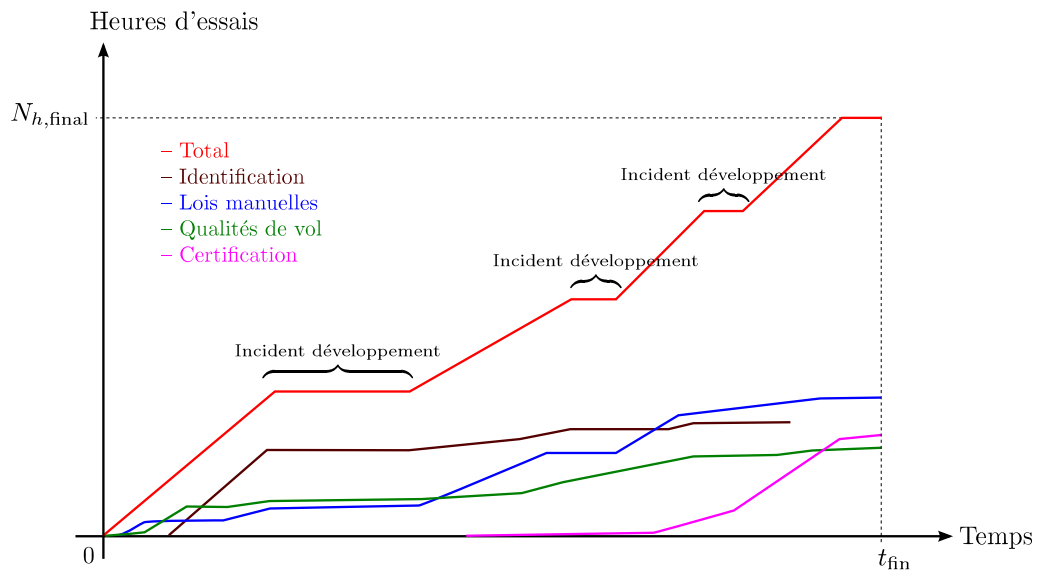


FIG. 2: Nombre d'heures d'essais en vol consacrées à l'identification et à la commande

## Enjeux et apports de la thèse

L'enjeu de cette thèse est de proposer des méthodes visant à découpler encore davantage ces phases d'*identification* et de *commande*. Pour cela, il faut donner un poids plus fort à l'avion, dont le comportement lors des premiers vols demeure une référence fixe durant tout le développement (à de rares exceptions près telles que l'ajout d'appendices aérodynamiques, modification de poussée, modifications de la répartition des ordres envoyés aux surfaces de contrôles, etc.). Il est dès lors envisageable de venir recalibrer les lois de pilotage *pendant* un essai en vol, comme il est suggéré sur la partie droite de la figure 1. Il est important de souligner qu'il s'agit de méthodes visant à *régler* les lois pendant une phase de développement et qu'il est exclu d'utiliser sur un avion de série : les lois seront ainsi optimisées puis « gelées ».

Toutefois, il n'est pas question de se dissocier intégralement du processus d'*identification* tant les simulations jouent un rôle central dans la mise au point, notamment dans les études statistiques de robustesse où un grand nombre de scénarios sont nécessaires à la validation. Le caractère fondamental de ces preuves de robustesse font qu'on ne peut se permettre un décalage prolongé entre des lois adaptées en vol et des lois compatibles des modèles identifiés.

Le fait de vouloir régler les lois en temps-réel soulève un problème majeur que nous allons traiter dans cette thèse. En effet, l'avion est un système notoirement à paramètres variants, ne serait-ce qu'à travers les modifications de l'environnement dans lequel il évolue (la densité de l'air évolue en fonction de l'altitude par exemple) et leur influence sur le comportement aérodynamique de l'avion. Il faut alors tenir compte du fait que lors d'un essai en vol, on n'observe le comportement de l'avion qu'autour d'un point de fonctionnement, aussi appelé point de vol. Par conséquent, le réglage ne pourra être que *local*. Il faut donc prévoir une structure capable de venir modifier la loi en conséquence, sans dégrader la performance en dehors du voisinage considéré. D'autre part, les lois de pilotage sont à la fois hybrides et à paramètres variants : elles peuvent être reconfigurées si un ensemble de conditions logiques est atteint et chacune des structures est fixée (i.e. chaque loi utilise un



objectif de contrôle et un ensemble de mesures différents). Leurs paramètres sont modifiés en fonction du point de vol à l'aide de fonctions d'*interpolation*. Pour des raisons de sûreté et de certification, il n'est pas envisageable de venir adapter en temps-réel la structure de la loi mais il semble plus judicieux de venir modifier les fonctions d'*interpolation*.

Nous avons ici choisi d'appliquer les méthodes tirées de la théorie de l'apprentissage et des réseaux de neurones. En effet, les réseaux de neurones sont réputés pour leurs capacités d'interpolation et pour leurs garanties théoriques quant à la pertinence de l'interpolation (nous pensons notamment à la notion de généralisation). De plus, certaines classes de réseaux de neurones permettent, via la théorie de l'apprentissage, de prendre en compte le caractère local de l'information qui leur est proposée.

## Organisation du mémoire

Le présent mémoire est divisé en quatre parties :

- la première partie est intitulée « Modélisation locale pour la performance globale ». Dans le chapitre 1, nous introduisons brièvement la notion d'apprentissage en général, qui peut entre autres, être considéré comme un problème d'interpolation (exacte ou approchée). C'est sur cette vision de l'apprentissage que repose le travail proposé dans cette thèse. Plusieurs solutions existent à ce problème et nous les séparons en deux types : *globale* et *locale*. Ce dernier type de solution fait l'objet du chapitre 2, que nous consacrons à la notion de mélange d'experts. Il s'agit d'une méthode permettant de tenir compte de la localité de l'information à apprendre dans le but d'améliorer globalement la performance. Cette partie s'appuie en majorité sur [Haykin, 1999] ;
- la deuxième partie est intitulée « Méthodologies et outils pour la retouche ». Il s'agit du cœur de ce mémoire dans laquelle nous présentons notre apport. L'introduction de cette partie est essentielle à l'articulation des trois chapitres constituant cette partie : tout en établissant plus concrètement le lien entre la première partie et le problème posé par Airbus, elle formalise ce dernier et identifie plus précisément les sous-problèmes que nous devons résoudre, tant dans un contexte hors ligne (i.e. réalisable en bureau d'études) que dans un contexte en ligne (i.e. en temps réel pendant un essai en vol). Le chapitre 3 aborde la question de la génération des données d'apprentissage. En effet, dans le cadre du réglage d'une loi de commande, ces données sont les paramètres caractéristiques de loi. Un travail de synthèse préalable est donc nécessaire. Nous proposons deux méthodes basées sur l'inversion de données temporelle, l'une dans un contexte hors ligne l'autre dans un contexte en ligne. Le chapitre 4 traite des aspects « pratiques » liés à la paramétrisation des mélanges d'experts et à leur apprentissage. Enfin le chapitre 5 aborde la notion de régularisation, laquelle est mentionnée à la fois aux chapitre 2 et 3. Bien qu'elle soit utilisée dans des contextes différents, nous avons préféré regrouper ces divers éléments dans un seul chapitre par souci de clarté ;
- la troisième partie aborde les applications qui ont été traitées. Le chapitre 6 présente de façon générale un avion de transport civil en donnant quelques notions de mécanique du vol tout en définissant les notations spécifiques utilisées par la suite, afin de familiariser le lecteur non nécessairement initié. Le chapitre 7 présente les résultats obtenus sur un estimateur de dérapage. Il s'agit d'une première application relativement simple sur laquelle un travail important a été effectué sur la détermination d'une structure optimale pour les mélanges d'experts. Le chapitre 8 ceux obtenus

pour une loi de compensation des aérofreins laquelle a permis de mettre au point la synthèse par inversion ainsi que l'apprentissage en ligne d'un mélange d'experts interpolant les paramètres d'un correcteur boucle ouverte ;

- la dernière partie dresse un bilan méthodologique du travail effectué en évaluant l'apport de ces méthodes dans le processus actuel ainsi que l'applicabilité à d'autres structures équivalentes intervenant dans les commandes de vol. Nous donnons ensuite des perspectives à partir de quelques remarques d'ordre pratique soulevées au cours du mémoire afin d'élargir le périmètre d'application.



Première partie

Modélisation locale pour la  
performance globale



# Chapitre 1

## Apprentissage

### Sommaire

---

<b>1.1 Définition de l'apprentissage . . . . .</b>	<b>9</b>
1.1.1 Apprentissage non-supervisé . . . . .	10
1.1.2 Apprentissage supervisé . . . . .	11
1.1.3 Approximation de fonctions . . . . .	12
<b>1.2 Formulations générales du problème d'apprentissage . . . . .</b>	<b>13</b>
1.2.1 Formulation globale . . . . .	13
1.2.2 Formulation locale . . . . .	14
<b>1.3 Conclusion . . . . .</b>	<b>15</b>

---

LA notion d'apprentissage local est issue de la théorie de l'apprentissage, par opposition à la notion d'apprentissage global. Si ce dernier cherche une unique solution au problème posé dans l'ensemble des solutions, son alternative locale va chercher plusieurs jeux de solutions localement satisfaisantes qui seront ensuite combinées pour donner une solution globale. Nous allons tout d'abord définir chacune des notions et en souligner les différences notoires. Pour chacune des formulations, nous proposons différentes solutions de façon générale afin d'introduire les notations nécessaires aux développements des chapitres suivants.

### 1.1 Définition de l'apprentissage

Il est difficile de donner une unique définition de cette notion d'apprentissage de par la diversité des activités qui lui sont associées. Toutefois, dans le contexte de tentative de reproduction du comportement du cerveau humain par un objet mathématique paramétré, que nous appellerons *machine*, nous pouvons donner une définition tirée de [Haykin, 1999].

**DÉFINITION 1.1.1 :** *L'apprentissage est le procédé par lequel les paramètres libres de la machine sont adaptés par l'intermédiaire d'une stimulation de l'environnement dans lequel la machine est immergée, en rapport à une tâche spécifique attribuée à celle-ci. Le type d'apprentissage est déterminé par la façon dont les paramètres évoluent.*

A l'évidence, il n'existe pas d'algorithme d'apprentissage unique mais plutôt, une grande variété de méthodes adaptées au contexte du problème. Elles diffèrent notamment par leur

façon d'ajuster les paramètres. Il faut également considérer le mode d'interaction de la machine avec son environnement. Nous parlerons de paradigme d'apprentissage s'appliquant à un modèle de l'environnement de la machine. Nous citerons notamment l'apprentissage *non-supervisé* et *supervisé*, sur lequel nous allons davantage insister.

### 1.1.1 Apprentissage non-supervisé

Comme son nom l'implique, il n'y a pas d'élément permettant de surveiller le processus d'apprentissage. On distingue deux catégories : l'apprentissage par *renforcement* et l'apprentissage *auto-organisé* [Ghahramani, 2004].

Dans le premier cas, l'apprentissage d'une relation entrée/sortie se fait par une interaction continue avec l'environnement : la machine reçoit des entrées, exécute des actions qui modifient l'état de l'environnement et reçoit des récompenses (ou des punitions). La figure 1.1 montre un exemple de système d'apprentissage par renforcement, conçu autour d'un dispositif de *critique*. Le but de la machine est d'apprendre à générer des actions qui vont maximiser les récompenses (ou minimiser les punitions) au cours de son existence. Ce type d'apprentissage est très lié à la théorie de la décision (statistiques et management) et à celle de la commande. Ce type d'apprentissage peut être généralisé en ajoutant dans l'environnement d'autres machines apprenantes interagissant entre elles. Cette approche est très voisine de la théorie des jeux.

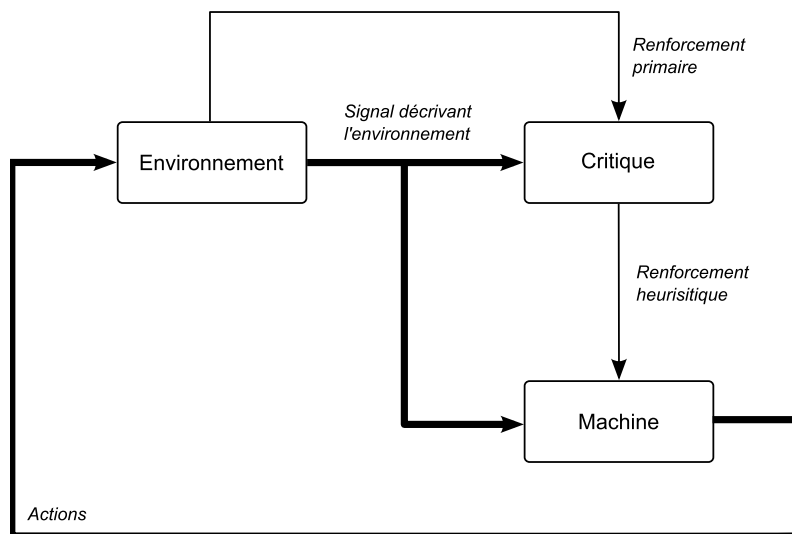


FIG. 1.1: Apprentissage par renforcement

Dans le deuxième cas de figure, il n'y a plus de système extérieur permettant d'aider l'apprentissage (cf. figure 1.2). La machine ne reçoit alors que des entrées et son but peut être alors de construire des représentations de ces entrées que l'on peut utiliser pour de la prise de décision, prédire des entrées futures, transmettre efficacement ces données à d'autres machines, etc. Il s'agit alors de dégager les caractéristiques intrinsèques des données reçues par la machine. Le partitionnement de données et la réduction de dimensionalité sont deux exemples classiques de ce type d'apprentissage.

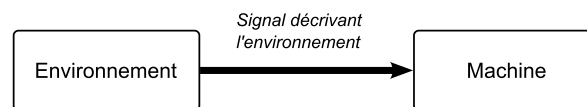


FIG. 1.2: Apprentissage auto-organisé

### 1.1.2 Apprentissage supervisé

L'apprentissage *supervisé* nécessite la présence d'un superviseur ayant connaissance de l'environnement, cette connaissance prenant la forme d'un ensemble de couples entrées/sorties. Par contre, la machine n'a pas cette connaissance. Lorsque le superviseur est soumis à une excitation de l'environnement (i.e. un exemple), il va fournir à la machine une réponse désirée pour cet exemple, qui en d'autres termes représente une action optimale à effectuer par la machine. L'effet de l'apprentissage se traduit alors par une modification des paramètres de la machine sous l'action combinée de l'exemple et d'un signal d'erreur, défini comme étant la différence entre la réponse désirée et la réponse de la machine (cf. figure 1.3). Cet ajustement est réalisé itérativement jusqu'à ce que la machine soit capable de reproduire le comportement du superviseur de façon optimale. Il faut alors voir ce procédé comme un « transfert » de la connaissance de l'environnement du superviseur à la machine. Une fois cette condition remplie, la machine peut se passer du superviseur et interagir par elle-même avec l'environnement.

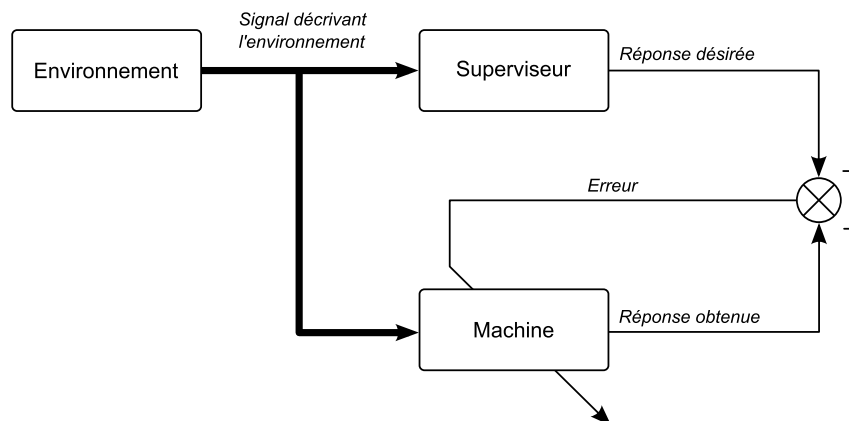


FIG. 1.3: Apprentissage supervisé

Dans ce contexte, les données disponibles générées par le superviseur sont divisées en deux sous-ensembles, statistiquement indépendants : un ensemble *d'apprentissage* et un ensemble de *validation*. Le premier sera utilisé afin d'ajuster au mieux les paramètres de la machine par rapport à la tâche qu'elle doit accomplir. Le second servira à valider le comportement de la machine sur des données sur lesquelles la machine n'a pas appris mais qui font partie de la connaissance du superviseur : on parle alors de capacité de *généralisation* de la machine. Pour illustrer simplement cette notion de généralisation, nous allons comparer le résultats de l'apprentissage de deux machines hypothétiques sur un même ensemble de données d'apprentissage. Nous supposons qu'elles ont une performance équivalente sur cet ensemble mais l'une (figure 1.4(a)) a une bonne capacité à généraliser l'autre non (figure 1.4(b)).

Cette notion de généralisation a d'ailleurs été étudiée de manière théorique, dans le



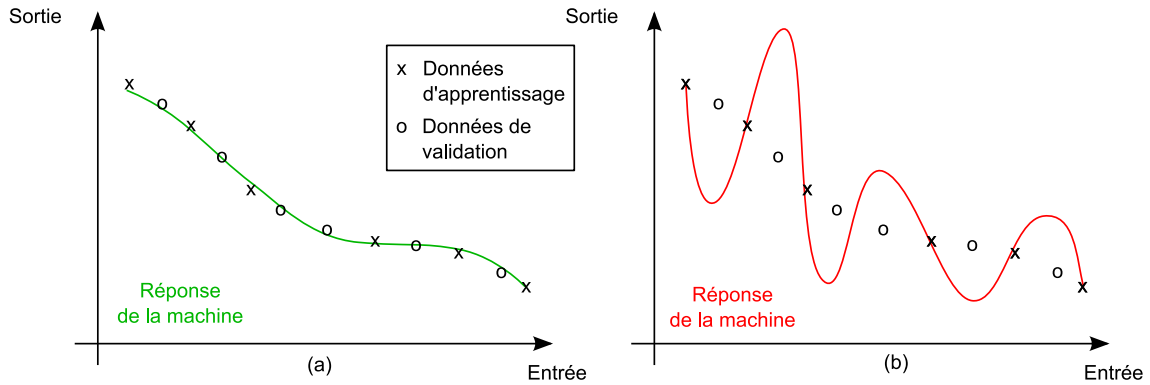


FIG. 1.4: (a) Bonne généralisation - (b) Mauvaise généralisation

cadre de l'apprentissage stochastique [Guermeur et Paugam-Moisy, 1999]. Les bases, initialement posées par Vapnik, font appel à la notion de dimension de Vapnik-Chervonenkis, ou VC-dimension [Vapnik et Chervonenkis, 1971], traduisant une *capacité* d'apprentissage de la machine et au principe de minimisation du risque structurel [Vapnik, 1991], [Vapnik, 1982]. De cette théorie ont découlé les *support vector machines* qui constituent une nouvelle classe de machines [Vapnik *et al.*, 1997].

Parmi les tâches pouvant être réalisées par un apprentissage supervisé, il y a notamment l'approximation de fonctions, la reconnaissance de formes, le filtrage et la commande des systèmes. Nous allons seulement développer la première de ces tâches car c'est celle qui nous intéresse pour le présent mémoire et qui met en pratique de façon la plus directe la notion d'apprentissage supervisé.

### 1.1.3 Approximation de fonctions

Considérons la relation entrée/sortie non-linéaire suivante :

$$\theta = f(\mathbf{a})$$

définie sur  $\mathcal{A}$  et dont seul un ensemble de  $K$  couples entrées/sorties est connu :

$$\mathcal{D} = \{(\mathbf{a}_k, \theta_k) \mid k = 1, \dots, K\}.$$

L'objectif est de concevoir une machine qui approche la fonction  $f$  de telle sorte que la fonction  $\hat{f}$  décrivant la relation entrée/sortie de la machine soit représentative de  $f$  au sens d'une distance (euclidienne par exemple), sur l'espace d'entrée :

$$\forall \mathbf{a} \in \mathcal{A}, \quad \|\hat{f}(\mathbf{a}) - f(\mathbf{a})\| < \epsilon,$$

où  $\epsilon$  est un nombre positif de faible valeur. Si  $K$  est assez grand pour que l'ensemble des  $\mathbf{a}_k$  recouvre l'espace d'entrée et que la machine dispose d'un nombre suffisant de paramètres,  $\epsilon$  peut-être rendu suffisamment petit.

D'un point de vue plus concret, nous pouvons considérer le graphe généré par la fonction  $\hat{f}$  plutôt que la fonction elle-même. Ainsi, les données issues de l'ensemble d'apprentissage et présentées à la machine représentent des points du graphe et l'apprentissage est alors l'optimisation de la procédure d'ajustement du graphe sur ces points. Tout comme la

figure 1.4 le laissait supposer, la notion de généralisation évoquée précédemment est alors synonyme d'*interpolation* [Broomhead et Lowe, 1988].

Le problème que nous venons de décrire est un candidat idéal pour l'utilisation de l'apprentissage supervisé où les  $\mathbf{a}_k$  serviraient d'exemples et les  $\theta_k$  de réponse désirée. Inversement, nous pouvons considérer que l'apprentissage supervisé comme un problème d'approximation de fonctions. C'est d'ailleurs ce point de vue que nous allons considérer par la suite.

## 1.2 Formulations générales du problème d'apprentissage

Nous allons ici distinguer deux types de formulations : l'apprentissage *global* et l'apprentissage *local*. La première formulation permet de trouver et d'optimiser un unique modèle (ou machine) sur l'ensemble des données d'apprentissage tandis que la seconde va permettre de gérer une combinaison de sous-modèles, performants localement dans des voisinages de l'espace d'entrée. Si le premier type est pour ainsi dire, le plus répandu, le second présente des propriétés intéressantes permettant d'intégrer la localisation de l'apprentissage dans l'espace d'entrée.

Le problème est posé comme un problème d'interpolation approchée. Notons au passage que le problème est formulé dans le cas de sorties scalaires pour simplifier la notation mais cela n'altère en rien le caractère général de l'énoncé.

**PROBLÈME 1.2.1 :** *Etant donné un ensemble de  $K$  points distincts  $\{\mathbf{a}_k \in \mathcal{A} \mid k = 1 \dots K\}$  et un ensemble correspondant de  $K$  valeurs de référence à estimer  $\{\theta_k \in \mathbb{R} \mid k = 1 \dots K\}$ , trouver une fonction  $\hat{f} : \mathcal{A} \rightarrow \mathbb{R}$  telle que :*

$$\hat{f} = \arg \min_h \frac{1}{2} \sum_{k=1}^K \mathcal{L}(\theta_k, h(\mathbf{a}_k)) \quad (1.1)$$

où  $\mathcal{L}(\cdot)$  est une fonction coût quelconque.

### 1.2.1 Formulation globale

Cette formulation va nous permettre de résoudre le problème 1.2.1 à l'aide d'une solution unique *globale* pour tout l'espace d'entrée. Etant donné que la machine est paramétrée, introduisons un espace paramétrique  $\mathcal{X}$ . La relation entrée/sortie de la machine devient

$$\hat{f}(\mathbf{a}, \boldsymbol{\chi}), \quad \forall (\mathbf{a}, \boldsymbol{\chi}) \in \mathcal{A} \times \mathcal{X},$$

où  $\mathbf{a}$  est le vecteur d'entrée et  $\boldsymbol{\chi}$  le vecteur de paramètres.

Résoudre le problème global va alors consister à trouver le vecteur  $\boldsymbol{\chi}^*$  qui minimise une certaine fonction de coût sur l'ensemble des points d'apprentissage :

$$\boldsymbol{\chi}^* = \arg \min_{\boldsymbol{\chi} \in \mathcal{X}} \frac{1}{2} \sum_{k=1}^K \mathcal{L}(\theta_k, \hat{f}(\mathbf{a}_k, \boldsymbol{\chi})), \quad (1.2)$$

Dans la pratique, un critère de type moindres carrés est souvent utilisé pour la fonction coût  $\mathcal{L}$ , ce qui conduit à l'expression suivante :

$$\boldsymbol{\chi}^* = \arg \min_{\boldsymbol{\chi} \in \mathcal{X}} \frac{1}{2} \sum_{k=1}^K (\theta_k - \hat{f}(\mathbf{a}_k, \boldsymbol{\chi}))^2. \quad (1.3)$$

Une telle formulation s'applique typiquement à l'utilisation des réseaux de neurones en tant que machines, l'exemple le plus classique étant les *perceptrons multi-couches* [Dreyfus *et al.*, 2004]. Des précisions sur ces réseaux de neurones sont données en annexe B.

L'inconvénient est que cette structure de réseau de neurones ne permet pas de prendre en compte la localisation des données d'apprentissage dans l'espace d'entrée, ce qui ne convient pas à notre problématique. De plus, les paramètres de ce type de machine n'ont aucune signification physique dans la mesure où il est difficile d'expliquer les variations de paramètres lors de l'apprentissage.

### 1.2.2 Formulation locale

Il existe deux façons d'intégrer cette composante locale à notre problème. Soit nous venons dans une première approche pondérer directement le critère global (1.2), soit nous intégrons des pondérations dans la structure du modèle, dans une seconde approche.

#### 1.2.2.1 Première approche : pondération du critère

Pour cet aspect, nous devons introduire un voisinage d'un point donné  $v(\mathbf{a}, \sigma)$  de l'espace d'entrée, que nous définissons comme suit

$$\forall \mathbf{a} \in \mathcal{A} \quad \forall \sigma > 0, \quad v(\mathbf{a}, \sigma) = \{\mathbf{a}' \in \mathcal{A} \mid d_{\mathcal{A}}(\mathbf{a}', \mathbf{a}) < \sigma\}$$

où  $d_{\mathcal{A}}(\cdot, \cdot)$  est une mesure de la distance entre deux éléments de l'espace  $\mathcal{A}$ . Pour chaque voisinage d'un point  $\mathbf{a}$ , on va chercher un vecteur de paramètres  $\boldsymbol{\chi}^*(\mathbf{a}, \sigma)$  qui minimise le critère (1.2), modifié comme suit :

$$\boldsymbol{\chi}^*(\mathbf{a}, \sigma) = \arg \min_{\boldsymbol{\chi} \in \mathcal{X}} \frac{1}{2} \sum_{k=1}^K \mathcal{L}(\theta_k, \hat{f}(\mathbf{a}_k, \boldsymbol{\chi})) \mathcal{K}(d_{\mathcal{A}}(\mathbf{a}_k, \mathbf{a}), \sigma), \quad (1.4)$$

où les pondérations sont définies par un noyau  $\mathcal{K}(d_{\mathcal{A}}(\mathbf{a}_k, \mathbf{a}), \sigma)$  (cf. définition donnée à l'annexe A), de taille  $\sigma$ , centré en  $\mathbf{a}_0$  [Bottou et Vapnik, 1996].

Cette formulation générale permet de retrouver des algorithmes classiques. Si nous choisissons par exemple un critère quadratique tel que présenté en (1.3), un modèle constant vis-à-vis des entrées pour la machine, nous pouvons retrouver par exemple la méthode des  $k$  plus proches voisins si le noyau est carré ou encore des réseaux à bases radiales si le noyau est gaussien [Broomhead et Lowe, 1988], [Moody et Darken, 1989]. Elle a également débouché sur l'élaboration des machines à vecteurs supports [Vapnik *et al.*, 1997].

Prenons par exemple le cas des réseaux à bases radiales [Atkeson *et al.*, 1997]. Dans ce cas, nous avons un modèle local constant tel que :

$$\hat{f}(\mathbf{a}, \boldsymbol{\chi}) = \boldsymbol{\chi}$$

Nous choisissons ici la norme euclidienne (cf. définition en annexe A) pour  $d_{\mathcal{A}}$  :

$$d_{\mathcal{A}}(\mathbf{a}_k, \mathbf{a}) = d_k(\mathbf{a}) = \|\mathbf{a}_k - \mathbf{a}\|_2$$

et d'utiliser un noyau gaussien pour pondérer le critère :

$$\mathcal{K}(d_k(\mathbf{a}), \sigma) = \exp\left(-\frac{d_k(\mathbf{a})^2}{2\sigma^2}\right).$$

Réécrivons le critère à minimiser pour  $\hat{f}(\mathbf{a}, \chi) = \chi$  :

$$\chi^*(\mathbf{a}, \sigma) = \arg \min_{\chi \in \mathcal{X}} \frac{1}{2} \sum_{k=1}^K (\theta_k - \chi)^2 \mathcal{K}(\|\mathbf{a}_k - \mathbf{a}\|_2, \sigma),$$

qui donne après résolution

$$\chi^*(\mathbf{a}, \sigma) = \frac{\sum_{k=1}^K \theta_k \mathcal{K}(\|\mathbf{a}_k - \mathbf{a}\|_2, \sigma)}{\sum_{k=1}^K \mathcal{K}(\|\mathbf{a}_k - \mathbf{a}\|_2, \sigma)} \quad (1.5)$$

Ce résultat nous donne une moyenne pondérée, fonction de  $\mathbf{a}$  : on parle alors de *moyenne pondérée par distance*<sup>1</sup>. Nous pouvons choisir de conserver cette structure pour notre machine, auquel cas nous avons affaire à un apprentissage *paresseux*<sup>2</sup>, où le traitement de données d'apprentissage est reporté au moment où la machine doit répondre à une requête.

### 1.2.2.2 Deuxième approche : pondération structurelle

Toutefois, si la quantité de données est importante, la structure précédente peut devenir lourde en terme de calculs. Il est alors possible de choisir  $R$  voisinages de points de  $\mathcal{A}$ , centrés en  $\mathbf{c}_r$  et de taille  $\sigma_r$ . En calculant pour chacun de ces centres la moyenne pondérée localement selon (1.5) avec  $\mathbf{a} = \mathbf{c}_r$  et  $\sigma = \sigma_r$  :

$$\hat{\theta}_r = \chi^*(\mathbf{c}_r, \sigma_r), \quad \forall r = 1, \dots, R$$

nous définissons un estimateur global en posant :

$$\hat{f}(\mathbf{a}, \chi^*) = \sum_{r=1}^R \hat{\theta}_r \mathcal{K}(\|\mathbf{c}_r - \mathbf{a}\|_2, \sigma_r) \quad (1.6)$$

L'équation (1.6) définit un réseau de neurones à base radiale que l'on appellera par la suite *réseau RBF*. Cette structure nous permet d'introduire la notion de *modèle local* et de *mélange d'experts* que nous allons présenter par la suite et développer davantage dans le chapitre suivant. Précédemment, nous avons calculé pour chaque centre du réseau RBF, une estimée constante  $\hat{\theta}_r$  localement optimale de la fonction à approcher. Nous avons donc défini un modèle local constant de la fonction, valide ou encore *expert*, dans le voisinage d'un point  $\mathbf{c}_r$  de l'espace d'entrée. Lorsque la machine est sollicitée par une entrée, la sortie de chaque modèle local est pondérée en fonction de la distance entre le point courant et le centre du voisinage rattaché à l'expert local : plus le point courant sera près d'un centre, plus l'estimée locale associée aura de poids dans l'estimation globale (cf. figure 1.5).

Cette deuxième approche consiste alors à intégrer au préalable cette subdivision de l'espace d'entrée dans la structure même de la machine. Cette solution a été retenue pour répondre au problème et fera l'objet de plus amples développements dans le chapitre 2.

## 1.3 Conclusion

Ce bref aperçu de la notion d'apprentissage nous a permis de dessiner les contours de la solution envisagée pour répondre au problème posé pour cette thèse. Nous n'avons

<sup>1</sup> « distance weighted averaging » en anglais

<sup>2</sup> « lazy learning » en anglais

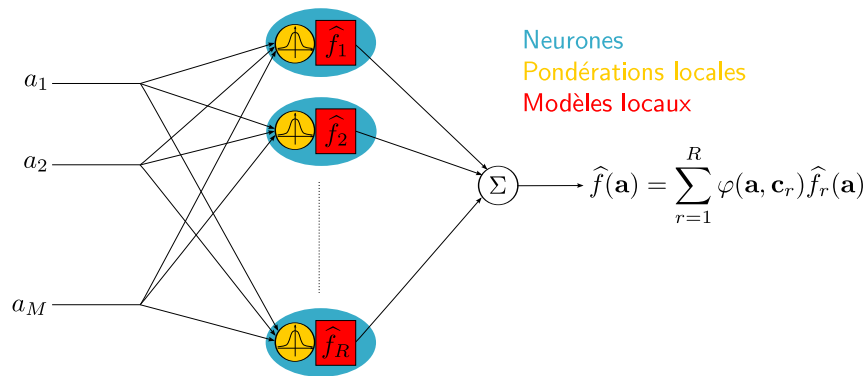


FIG. 1.5: Un exemple de machine localement structurée : un réseau RBF

ici considéré que l'aspect estimation de fonctions de l'apprentissage, dans la mesure où nous comptons utiliser ces méthodes pour interpoler les paramètres des lois de pilotage. Nous avons également présenté une formulation du problème d'apprentissage permettant de prendre en compte la localité des données d'apprentissage.

## Chapitre 2

# Mélanges d'experts

### Sommaire

---

<b>2.1 Réseaux RBF</b> . . . . .	<b>17</b>
<b>2.2 Réseaux RBF généralisés</b> . . . . .	<b>19</b>
2.2.1 Réseaux RBF régularisés . . . . .	19
2.2.2 Réseaux RBF généralisés . . . . .	20
2.2.3 Propriétés d'approximation . . . . .	22
2.2.4 Conclusion . . . . .	22
<b>2.3 Mélange d'experts</b> . . . . .	<b>23</b>
2.3.1 Description . . . . .	23
2.3.2 L'apprentissage . . . . .	23
2.3.3 Généralisation au cas matriciel . . . . .	26
<b>2.4 Conclusion</b> . . . . .	<b>27</b>

---

Nous détaillons dans ce chapitre les solutions au problème d'apprentissage local auxquelles nous allons donner une structure qui permet de prendre en compte cette localisation. Nous allons dans un premier temps passer en revue les réseaux RBF. Il s'agit d'une solution où les modèles locaux sont des constantes et où les fonctions de pondérations sont des fonctions à base radiale. Cela permettra d'alléger les notations afin d'introduire les notions de réseau RBF régularisés et généralisés. Enfin, nous présenterons les mélanges d'experts et détaillerons l'extension du concept des réseaux RBF à des modèles locaux affines, cette représentation étant suffisamment aisée à manipuler et relativement peu complexe pour les applications visées.

### 2.1 Réseaux RBF

Nous allons à présent revenir au problème initial 1.2.1 et décrire une autre façon d'aborder la technique RBF [Powell, 1988], [Haykin, 1999]. Contrairement à ce qui précède, nous allons donner directement à la machine une structure locale telle que :

$$\hat{f}(\mathbf{a}, \boldsymbol{\chi}) = \sum_{k=1}^K \chi_k \varphi(\|\mathbf{a} - \mathbf{a}_k\|) \quad (2.1)$$

où les  $\varphi(\cdot)$  sont des fonctions à base radiale. Nous notons que dans cette structure originelle de réseau RBF, il y a autant de centres que de données d'apprentissage et que ces centres sont les entrées des données.

Reprenons le problème 1.2.1 pour lequel nous utilisons une fonction de coût quadratique et une solution  $\hat{f}$  sous la forme proposée en (2.1). Nous devons donc minimiser le critère suivant :

$$\mathcal{J}(\hat{f}) = \frac{1}{2} \sum_{k=1}^K \left( \theta_k - \sum_{j=1}^K \chi_j \varphi(\|\mathbf{a}_k - \mathbf{a}_j\|) \right)^2 \quad (2.2)$$

$$= \frac{1}{2} \|\boldsymbol{\theta} - \Phi \boldsymbol{\chi}\|^2 \quad (2.3)$$

avec

$$\Phi = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1K} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{K1} & \varphi_{K2} & \cdots & \varphi_{KK} \end{pmatrix} \quad (2.4)$$

$$\boldsymbol{\chi} = (\chi_1 \ \chi_2 \ \cdots \ \chi_K)^\top \quad (2.5)$$

$$\boldsymbol{\theta} = (\theta_1 \ \theta_2 \ \cdots \ \theta_K)^\top \quad (2.6)$$

avec

$$\varphi_{kj} = \varphi(\|\mathbf{a}_k - \mathbf{a}_j\|), \quad \forall k, j = 1, \dots, K \quad (2.7)$$

La condition d'optimalité au premier ordre donne :

$$\Phi^\top (\Phi \boldsymbol{\chi} - \boldsymbol{\theta}) = 0.$$

Or, si  $\Phi$  est inversible, cette condition est équivalente à :

$$\Phi \boldsymbol{\chi} = \boldsymbol{\theta} \quad (2.8)$$

$$\iff \boldsymbol{\chi}^* = \Phi^{-1} \boldsymbol{\theta} \quad (2.9)$$

Nous pouvons donc conclure que si  $\Phi$  est inversible, nous trouvons une solution exacte au problème d'interpolation. L'inversibilité de la matrice  $\Phi$  est donnée par le théorème de Michelli [Michelli, 1986], pour une grande classe de fonctions à base radiale.

**THÉORÈME DE MICHELLI :** *Soit  $\varphi$  une fonction continue sur  $\mathbb{R}^+$  et positive sur  $\mathbb{R}^{+*}$ . Si sa dérivée première est monotone mais non constante sur  $\mathbb{R}^{+*}$ , alors pour des vecteurs distincts  $\{\mathbf{a}_k \in \mathcal{A} \mid k = 1, \dots, K\}$ , la matrice  $\Phi$  dont le  $ij^{\text{ème}}$  élément est  $\varphi(\|\mathbf{a}_i - \mathbf{a}_j\|^2)$ , est inversible.*

L'intérêt de ce théorème réside dans le fait que si la dérivée première de  $\varphi$  est monotone, elle peut être utilisée comme fonction à base radiale. Il en découle alors une liste de fonctions pouvant être utilisées lors d'une mise en pratique de cette méthode [Poggio et Girosi, 1989] :

– Fonction gaussienne

$$\varphi(d) = \exp\left(-\frac{d}{\sigma^2}\right)$$

– Multiquadrique

$$\varphi(d) = \sqrt{(d + \sigma^2)}$$

- Multiquadrique inverse

$$\varphi(d) = \frac{1}{\sqrt{(d + \sigma^2)}}$$

- Multiquadrique généralisée

$$\varphi(d) = (d + \sigma^2)^\alpha, \quad \alpha \in ]-\infty, 1[$$

## 2.2 Réseaux RBF généralisés

Toutefois, cette solution peut ne pas être toujours adéquate. Si par exemple  $K$  est beaucoup plus grand que le nombre de degrés de liberté sous-jacent du problème, ce dernier est alors sur-déterminé ce qui pourrait engendrer une mauvaise capacité de généralisation (cf. figure 1.4, p. 12). De plus, ce problème est *mal-posé* au sens d'Hadamard, par opposition à la notion de problème *bien-posé* dont nous donnons la définition au paragraphe §5.1 (p. 81). Afin de proposer une meilleure solution, nous allons régulariser le problème au sens de Tikhonov [Tikhonov et Arsenin, 1977]. Le lecteur pourra trouver davantage de détails au chapitre 5 (p. 81).

### 2.2.1 Réseaux RBF régularisés

Pour des raisons de clarté, nous ne donnerons dans ce paragraphe que les résultats du chapitre 5 (p. 81) appliqués aux réseaux RBF. La régularisation consiste à stabiliser la solution en ajoutant au critère un terme de pénalité sur les dérivées de la solution, à l'aide d'un opérateur différentiel linéaire  $\mathbf{D}$  et d'un scalaire positif  $\lambda$  permettant de pondérer cette pénalité par rapport au critère principal. Le critère ainsi régularisé devient :

$$\mathcal{J}_\lambda(\hat{f}) = \frac{1}{2} \sum_{k=1}^K (\theta_k - \hat{f}(\mathbf{a}_k))^2 + \frac{\lambda}{2} \|\mathbf{D}\hat{f}\|^2 \quad (2.10)$$

La solution qui minimise ce critère est la suivante :

$$\hat{f}_\lambda(\mathbf{a}) = \sum_{k=1}^K \chi_k^* g(\mathbf{a}, \mathbf{a}_k) \quad (2.11)$$

où

$$\boldsymbol{\chi}^* = (G + \lambda I)^{-1} \boldsymbol{\theta} \quad (2.12)$$

$$G = \begin{pmatrix} g(\mathbf{a}_1, \mathbf{a}_1) & g(\mathbf{a}_1, \mathbf{a}_2) & \cdots & g(\mathbf{a}_1, \mathbf{a}_K) \\ g(\mathbf{a}_2, \mathbf{a}_1) & g(\mathbf{a}_2, \mathbf{a}_2) & \cdots & g(\mathbf{a}_2, \mathbf{a}_K) \\ \vdots & \vdots & \ddots & \vdots \\ g(\mathbf{a}_K, \mathbf{a}_1) & g(\mathbf{a}_K, \mathbf{a}_2) & \cdots & g(\mathbf{a}_K, \mathbf{a}_K) \end{pmatrix} \quad (2.13)$$

La fonction  $g(.,.)$  ci-dessus est la fonction de Green de l'opérateur différentiel linéaire  $\mathbf{L} = \mathbf{D}^\sharp \mathbf{D}$  (cf. définition en annexe A.2, p. 160), où  $\mathbf{D}^\sharp$  est l'opérateur adjoint de  $\mathbf{D}$ . Le cas qui nous intéresse tout particulièrement ici est celui où  $\mathbf{D}$  est invariant par translation et rotation. La fonction de Green associée ne dépendra alors que de la distance entre  $\mathbf{a}$  et  $\mathbf{a}_k$  :

$$g(\mathbf{a}, \mathbf{a}_k) = g(d(\mathbf{a}, \mathbf{a}_k)) \quad (2.14)$$



Dans ce cas, la fonction de Green devra être une fonction à base radiale et la solution régularisée prendra la forme suivante [Poggio et Girosi, 1990] :

$$\widehat{f}_\lambda(\mathbf{a}) = \sum_{k=1}^K \chi_k^* g(d(\mathbf{a}, \mathbf{a}_k)) \quad (2.15)$$

Cette solution répond au problème 1.2.1 mais elle diffère de celle présentée en (2.1) dans la mesure où elle a été *régularisée*. Notons que la solution (2.9) est le cas particulier où  $\lambda = 0$ .

L'opérateur différentiel linéaire  $\mathbf{L}$  correspondant à une fonction de Green gaussienne multivariable est le suivant [Poggio et Girosi, 1990] :

$$\mathbf{L} = \sum_{n=0}^{\infty} (-1)^n \alpha_n \nabla^{2n} \quad (2.16)$$

avec

$$\alpha_n = \frac{\sigma_k^{2n}}{n!2^n} \quad (2.17)$$

$\nabla^{2n}$  est l'opérateur de Laplace itéré : si  $\dim \mathcal{A} = M$ , alors

$$\nabla^2 = \sum_{m=1}^M \frac{\partial^2}{\partial a_m^2}$$

Or par définition, nous avons

$$\mathbf{L} = \mathbf{D}^\sharp \mathbf{D}.$$

L'opérateur de régularisation  $\mathbf{D}$  correspondant à une fonction de Green gaussienne multivariable est alors

$$\mathbf{D} = \sum_n \sqrt{\alpha_n} \left( \sum_{m=1}^M \frac{\partial}{\partial a_m} \right)^n \quad (2.18)$$

$$= \sum_{n_1+n_2+\dots+n_M=n} \sqrt{\alpha_n} \frac{\partial^n}{\partial a_1^{n_1} \partial a_2^{n_2} \dots \partial a_M^{n_M}} \quad (2.19)$$

## 2.2.2 Réseaux RBF généralisés

La méthode précédente présente encore un inconvénient majeure : pour peu que le nombre de données d'apprentissage soit important, le réseau aura une taille prohibitive pour une utilisation ultérieure. De plus, la probabilité d'avoir un mauvais conditionnement est plus grande lorsque la taille des matrices augmente. Nous proposons ici de trouver une solution sous-optimale  $\widehat{f}^*$  sous la forme d'un développement impliquant un nombre  $R < K$  plus réduit de termes [Poggio et Girosi, 1989] :

$$\widehat{f}^*(\mathbf{a}, \boldsymbol{\chi}) = \sum_{r=1}^R \chi_r \varphi_r(\mathbf{a}) \quad (2.20)$$

avec

$$\varphi_r(\mathbf{a}) = g(\|\mathbf{a} - \mathbf{c}_r\|), \quad (2.21)$$

où les  $\mathbf{c}_r$  sont les centres des fonctions de Green, que nous verrons comment déterminer par la suite. Notons que cette solution sous-optimale nous permet de retrouver la solution précédente dans le cas où  $R = K$  et  $\mathbf{c}_k = \mathbf{a}_k$ .

Nous pouvons écrire à présent le critère régularisé que doit minimiser notre solution à l'aide des notations précédentes :

$$\mathcal{J}(\hat{f}^*) = \sum_{k=1}^K \left( \theta_k - \sum_{r=1}^R \chi_r g(\|\mathbf{a} - \mathbf{c}_r\|) \right)^2 + \lambda \|\mathbb{D}\hat{f}^*\|^2 \quad (2.22)$$

Le premier terme peut se réécrire sous forme de la norme de la différence des vecteurs  $\|\boldsymbol{\theta} - G\boldsymbol{\chi}\|^2$  avec les notations suivantes :

$$\boldsymbol{\theta} = (\theta_1 \quad \cdots \quad \theta_K)^\top \quad (2.23)$$

$$\boldsymbol{\chi} = (\chi_1 \quad \cdots \quad \chi_R)^\top \quad (2.24)$$

$$G = \begin{pmatrix} g(\|\mathbf{a}_1 - \mathbf{c}_1\|) & g(\|\mathbf{a}_1 - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{a}_1 - \mathbf{c}_R\|) \\ g(\|\mathbf{a}_2 - \mathbf{c}_1\|) & g(\|\mathbf{a}_2 - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{a}_2 - \mathbf{c}_R\|) \\ \vdots & \vdots & \ddots & \vdots \\ g(\|\mathbf{a}_K - \mathbf{c}_1\|) & g(\|\mathbf{a}_K - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{a}_K - \mathbf{c}_R\|) \end{pmatrix} \quad (2.25)$$

Nous obtenons une forme similaire à celle trouvée précédemment à la différence notable que la matrice  $G$  n'est plus ni carrée, ni symétrique.

Le terme de régularisation s'écrit quant à lui :

$$\begin{aligned} \|\mathbb{D}\hat{f}^*\|^2 &= \int_{\mathcal{A}} \mathbb{D}\hat{f}^*(\mathbf{a})\mathbb{D}\hat{f}^*(\mathbf{a})d\mathbf{a} \\ &= \int_{\mathcal{A}} \hat{f}^*(\mathbf{a})\mathbb{D}^\# \mathbb{D}\hat{f}^*(\mathbf{a})d\mathbf{a} \\ &= \int_{\mathcal{A}} \sum_{r=1}^R \chi_r g(\|\mathbf{a} - \mathbf{c}_r\|) \sum_{r=1}^R \chi_r \mathbb{D}^\# \mathbb{D}g(\|\mathbf{a} - \mathbf{c}_r\|)d\mathbf{a} \\ &= \int_{\mathcal{A}} \sum_{r=1}^R \chi_r g(\|\mathbf{a} - \mathbf{c}_r\|) \sum_{r=1}^R \chi_r \delta(\mathbf{a} - \mathbf{c}_r)d\mathbf{a} \\ &= \sum_{r=1}^R \sum_{j=1}^R \chi_r \chi_j g(\|\mathbf{c}_r - \mathbf{c}_j\|) \\ &= \boldsymbol{\chi}^\top G_0 \boldsymbol{\chi} \end{aligned} \quad (2.26)$$

où  $G_0$  est la matrice symétrique suivante :

$$G_0 = \begin{pmatrix} g(\|\mathbf{c}_1 - \mathbf{c}_1\|) & g(\|\mathbf{c}_1 - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{c}_1 - \mathbf{c}_R\|) \\ g(\|\mathbf{c}_2 - \mathbf{c}_1\|) & g(\|\mathbf{c}_2 - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{c}_2 - \mathbf{c}_R\|) \\ \vdots & \vdots & \ddots & \vdots \\ g(\|\mathbf{c}_R - \mathbf{c}_1\|) & g(\|\mathbf{c}_R - \mathbf{c}_2\|) & \cdots & g(\|\mathbf{c}_R - \mathbf{c}_R\|) \end{pmatrix}. \quad (2.27)$$

La minimisation de (2.22) par rapport au vecteur de paramètres  $\boldsymbol{\chi}$  donne le résultat suivant :

$$(G^\top G + \lambda G_0) \boldsymbol{\chi} = G^\top \boldsymbol{\theta}. \quad (2.28)$$

### 2.2.3 Propriétés d'approximation

A l'instar des perceptrons multi-couches [Dreyfus *et al.*, 2004], [Haykin, 1999], les réseaux RBF présentent de bonnes performances d'approximation. Cette famille de réseaux est suffisamment étendue pour approcher uniformément n'importe quelle fonction continue sur un ensemble compact. Nous citerons ici le théorème d'approximation universel, appliqué à cette famille [Park et Sandberg, 1991].

THÉORÈME 2.2.1 : Soit  $g : \mathcal{A} \rightarrow \mathbb{R}$  une fonction intégrable bornée continue et telle que

$$\int_{\mathcal{A}} g(\mathbf{a}) d\mathbf{a} \neq 0$$

Soit  $\mathcal{S}_G$  la famille des réseaux RBF, constituée des fonctions  $\hat{f} : \mathcal{A} \rightarrow \mathbb{R}$  telles que

$$\hat{f}(\mathbf{a}) = \sum_{r=1}^R \chi_r g\left(\frac{\mathbf{a} - \mathbf{c}_r}{\sigma}\right)$$

avec  $\sigma > 0$ ,  $\chi_r \in \mathbb{R}$  et  $\mathbf{c}_r \in \mathcal{A}$ ,  $\forall r = 1, \dots, R$ .

Alors, pour toute fonction continue  $f : \mathcal{A} \rightarrow \mathbb{R}$ , il existe un réseau RBF rattaché à un ensemble de centres  $\{\mathbf{c}_r\}_{r=1, \dots, R}$  et une taille  $\sigma > 0$  commune, tel que la fonction  $\hat{f}(\mathbf{x})$  réalisée par le réseau RBF soit suffisamment proche de  $f(\mathbf{a})$  au sens de la norme  $\mathcal{L}^p$ ,  $p \in [1, \infty[$ .

La définition de la norme  $\mathcal{L}^p$  est donnée en annexe A. Notons que dans le théorème tel qu'il vient d'être énoncé, le noyau  $g$  ne doit pas nécessairement satisfaire la propriété de symétrie radiale. Ce théorème ne se limite donc pas aux seuls réseaux RBF.

### 2.2.4 Conclusion

Avec les RBF généralisés, nous commençons à discerner la notion de modèle local. En effet, si l'on reprend la définition du réseau, donnée en (2.20)

$$\hat{f}^*(\mathbf{a}, \boldsymbol{\chi}) = \sum_{r=1}^R \chi_r \varphi_r(\mathbf{a})$$

avec

$$\varphi_r(\mathbf{a}) = g(\|\mathbf{a} - \mathbf{c}_r\|),$$

nous pouvons considérer que les  $\chi_r$  sont des approximations locales de  $f$  à l'ordre 0, au voisinage de  $\mathbf{c}_r$ . En d'autres termes, il s'agit de modèles locaux constants. Il peut donc s'avérer intéressant d'augmenter la complexité de ces modèles locaux afin d'améliorer davantage la qualité de l'estimation. Bien que l'on perde au passage la notion de réseau de neurones, au sens du mimétisme au neurones biologiques, on y gagne en lisibilité et en compréhension des divers éléments constituant le réseau<sup>1</sup>.

<sup>1</sup>Un des reproches qui est souvent fait aux réseaux de neurones est son côté boîte noire, à savoir qu'il manque une signification « physique » à ses paramètres

## 2.3 Mélange d'experts

Cette notion a été introduite au début des années 90 dans [Jacobs *et al.*, 1991], [Jacobs et Jordan, 1990] et [Jacobs *et al.*, 1990]. Elle met en œuvre un apprentissage coopératif permettant de traiter des problèmes complexes pour lesquels on ne dispose d'aucune connaissance a priori.

Les mélanges d'experts (MEX) sont une application du principe « diviser pour mieux régner ». Un problème donné, notamment complexe, est divisé en un certain nombre de sous-problèmes plus simples, affectés à un « expert ». Chaque expert est mis en compétition pour apprendre des exemples provenant de différentes régions de l'espace d'entrée, ces régions pouvant se chevaucher (nous parlerons par la suite de *recouvrement*). Il en résulte que chaque expert apprend une fonction différente, ce qui permet au MEX de décomposer une fonction et d'assigner à chaque fonction locale l'expert le mieux adapté. Notons que cette notion a été étendue à celle de mélange hiérarchique d'experts [Jordan et Jacobs, 1994].

### 2.3.1 Description

L'architecture d'un MEX est divisée en deux parties : d'un côté un ensemble d'experts et de l'autre un réseau dit *superviseur* (ou encore sélecteur ou « gating network » en anglais), comme le montre la figure 2.1(a). Les experts apprennent sur le même jeu de données. La modification de leurs paramètres sera plus ou moins importante selon le degré d'appartenance du point courant d'apprentissage aux régions d'expertise. Ce degré d'appartenance intervient ici sous forme de pondération donnée par le réseau superviseur.

Si tous les experts reçoivent les mêmes entrées, le réseau superviseur peut quant à lui en recevoir d'autres. Ce dernier possède également autant de sorties qu'il y a d'experts dans le mélange. La structure de chaque expert est en général la même pour chacun et celle-ci peut varier d'une application à une autre : fonction constante, linéaire ou encore polynomiale, perceptrons multi-couches, etc. Ce choix dépend de la complexité que l'on peut affecter à la solution proposée. Nous avons vu à la section précédente, que la notion de réseaux RBF généralisés peut s'interpréter comme un mélange d'experts dont la structure est une fonction constante des entrées du réseaux et dont le réseau superviseur est un réseau mono-couche à fonction d'activation gaussienne, comme le montre la figure 2.1(b).

### 2.3.2 L'apprentissage

Nous allons voir à présent comment est réalisé l'apprentissage d'un MEX dont les  $R$  experts, ou modèle locaux, sont des fonctions affines des entrées :

$$\hat{f}_r(\mathbf{a}) = \tilde{\mathbf{a}}^T \boldsymbol{\chi}_r, \quad \forall r = 1, \dots, R \quad (2.29)$$

où  $\boldsymbol{\chi}_r$  est le vecteur de paramètres du  $r^{\text{ème}}$  modèle local et où  $\tilde{\mathbf{a}}^T = (\mathbf{a}^T \ 1)$ . Nous venons concaténer 1 au vecteur  $\mathbf{a}$  afin de réécrire le modèle affine sous forme d'un modèle linéaire.

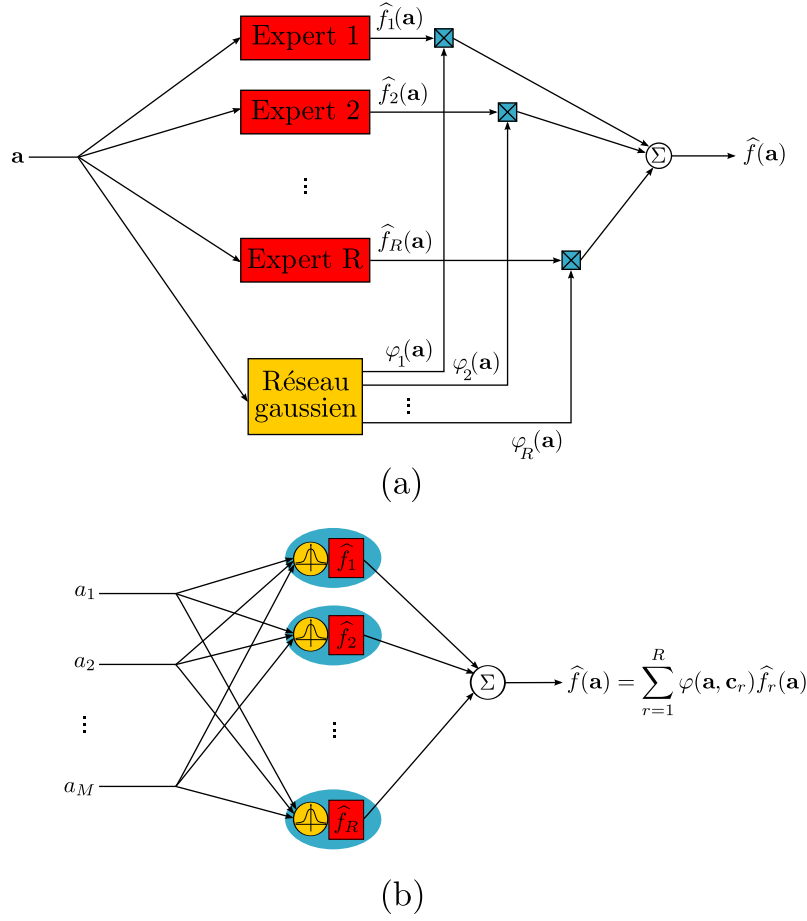


FIG. 2.1: Analogie entre un mélange d'experts (a) et un réseau RBF (b)

La sortie du MEX s'écrit donc :

$$\hat{f}(\mathbf{a}) = \sum_{r=1}^R \hat{f}_r(\mathbf{a}) \varphi_r(\mathbf{a}) \quad (2.30)$$

$$= \sum_{r=1}^R \varphi_r(\mathbf{a}) \tilde{\mathbf{a}}^T \boldsymbol{\chi}_r \quad (2.31)$$

$$= \sum_{r=1}^R \boldsymbol{\psi}_r(\mathbf{a})^T \boldsymbol{\chi}_r \quad (2.32)$$

$$\hat{f}(\mathbf{a}) = \boldsymbol{\psi}(\mathbf{a})^T \boldsymbol{\chi} \quad (2.33)$$

avec

$$\boldsymbol{\chi} = (\boldsymbol{\chi}_1 \quad \cdots \quad \boldsymbol{\chi}_R)^T \quad (2.34)$$

$$\boldsymbol{\psi}_r(\mathbf{a}) = \varphi_r(\mathbf{a}) \tilde{\mathbf{a}}, \quad \forall r = 1, \dots, R \quad (2.35)$$

$$\boldsymbol{\psi}(\mathbf{a})^T = (\boldsymbol{\psi}_1(\mathbf{a})^T \quad \cdots \quad \boldsymbol{\psi}_R(\mathbf{a})^T) \quad (2.36)$$

Pour la suite du calcul, nous avons conservé la structure de réseau RBF vue précédemment mais ce n'est pas un choix unique. Par exemple, l'algorithme original utilise un

réseau superviseur dont la fonction d'activation est la fonction « softmax » [Wira, 2002] :

$$\varphi_r(\mathbf{a}) = \frac{\exp\left(\frac{\mathbf{a}^\top \mathbf{c}_r}{\sigma}\right)}{\sum_{j=1}^R \exp\left(\frac{\mathbf{a}^\top \mathbf{c}_j}{\sigma}\right)}, \quad \forall r = 1, \dots, R \quad (2.37)$$

Pour plus de détails sur cette solution, le lecteur pourra se référer à [Haykin, 1999].

Posons à présent le problème d'interpolation approchée à notre MEX, comme nous l'avons au fait préalable au paragraphe §2.2.2. Le critère (non régularisé) à minimiser est donc le suivant :

$$\mathcal{J}(\hat{f}) = \frac{1}{2} \sum_{k=1}^K \left( \theta_k - \sum_{r=1}^R \psi_r(\mathbf{a}_k)^\top \boldsymbol{\chi}_r \right)^2 \quad (2.38)$$

$$= \frac{1}{2} \|\boldsymbol{\theta} - \Psi^\top X\|^2 \quad (2.39)$$

avec

$$\Psi^\top = \begin{pmatrix} \Psi_{11}^\top & \Psi_{12}^\top & \dots & \Psi_{1R}^\top \\ \Psi_{21}^\top & \Psi_{22}^\top & \dots & \Psi_{2R}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{K1}^\top & \Psi_{K2}^\top & \dots & \Psi_{KR}^\top \end{pmatrix} \quad (2.40)$$

$$\Psi_{kr} = \psi_r(\mathbf{a}_k) \quad (2.41)$$

$$X = (\boldsymbol{\chi}_1 \quad \boldsymbol{\chi}_2 \quad \dots \quad \boldsymbol{\chi}_R)^\top \quad (2.42)$$

$$\boldsymbol{\theta} = (\theta_1 \quad \theta_2 \quad \dots \quad \theta_R)^\top \quad (2.43)$$

La condition d'optimalité au premier ordre par rapport à  $X$  s'écrit donc :

$$\Psi \left( \Psi^\top X - \boldsymbol{\theta} \right) = 0 \quad (2.44)$$

$$\iff X^* = \left( \Psi \Psi^\top \right)^{-1} \Psi^\top \boldsymbol{\theta} \quad (2.45)$$

L'utilisation d'une fonction  $\varphi$  satisfaisant le théorème de Michelli et un ensemble de vecteurs distincts  $\mathbf{a}_k$  semble suffisant pour assurer l'inversibilité de  $\Psi \Psi^\top$ .

**Illustration sur un exemple simple** Afin de montrer l'intérêt d'une augmentation de complexité des modèles locaux, nous proposons d'appliquer sur un exemple simple, l'approche à modèles locaux constants (i.e. les réseaux RBF généralisés) d'un côté et l'approche à modèles locaux linéaires (i.e. un mélange d'experts).

Nous donnons donc une fonction  $f$  à identifier sur l'intervalle  $[0, 1]$ .

$$f(a) = \exp(-a) + 0,05 \cdot \sin(6\pi a)$$

Nous tirons  $K = 500$  points de cet intervalle à l'aide d'une loi uniforme et calculons l'image correspondante par  $f$ . Nous ajoutons également un bruit blanc gaussien centré de variance  $\sigma_b = 0,01$  sur les « mesures ».

Nous choisissons de placer 4 centres équi-répartis dans l'intervalle considéré. Le résultat pour le MEX à modèles constants est donné à la figure 2.2 et celui pour le MEX à modèle

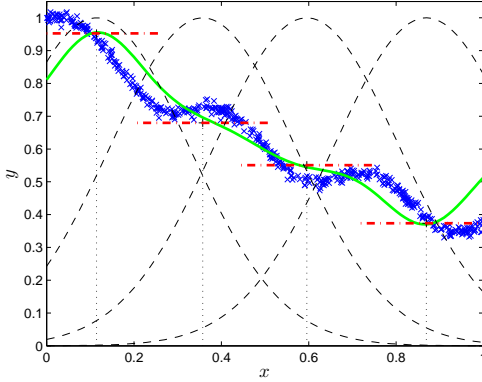


FIG. 2.2: Exemple de MEX à modèles locaux constants

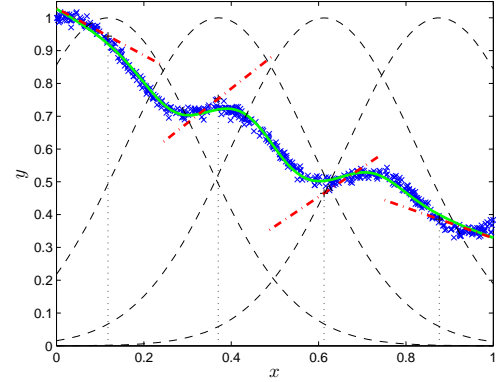


FIG. 2.3: Exemple de MEX à modèles locaux linéaires

linéaire à la figure 2.3. Sur ces figures, les points bleus représentent les données d'apprentissage, la courbe épaisse en trait plein vert représente la sortie du MEX sur l'intervalle, les droites en tirets-pointillés rouges représentent les modèles locaux et les courbes noires en pointillés montrent les RBF.

Nous voyons assez nettement que la performance est bien meilleure pour le deuxième MEX : l'erreur relative d'apprentissage pour le premier MEX est d'environ 10% alors que celle du second est environ de 2%. Toutefois, Nous pouvons voir que sur les bords de domaine, les MEX ne sont pas satisfaisants. Nous verrons par la suite comment remédier à ce problème soit en positionnant certains centres sur les frontières de l'espace d'entrée soit en normalisant les pondérations données par les RBF (cf. section §4.1).

### 2.3.3 Généralisation au cas matriciel

Avant de continuer, nous allons préciser davantage les espaces d'entrées et de sortie, afin de clarifier les notations utilisées dans la suite. Nous allons donc considérer que l'espace d'entrée  $\mathcal{A}$  est  $\mathbb{R}^M$  et l'espace de sortie  $\mathcal{O}$  est  $\mathbb{R}^{P \times Q}$ , où  $M$ ,  $P$  et  $Q$  sont des entiers naturels non nuls. Ce faisant, nous sortons du cas purement scalaire pour pouvoir traiter celui plus général des matrices réelles, qui nous sera utile par la suite lorsque nous aurons à interpoler les matrices d'un système à paramètres variants mis sous forme d'état.

A l'aide de manipulations quelque peu fastidieuses, nous allons nous ramener à une écriture de la sortie globale du MEX, linéaire par rapport aux paramètres à calculer. Tout d'abord, introduisons la forme du modèle local :

$$\hat{f}_r(\mathbf{a}) = \sum_{m=1}^{M+1} \tilde{a}_m \chi_{m,r}, \quad \chi_{m,r} \in \mathbb{R}^{P \times Q}, \quad \forall m, r \quad (2.46)$$

où  $\tilde{\mathbf{a}}^\top = (\mathbf{a}^\top \ 1)$ . La sortie global du MEX s'écrit alors :

$$\hat{f}(\mathbf{a}) = \sum_{r=1}^R \varphi_r(\mathbf{a}) \sum_{m=1}^{M+1} \tilde{a}_m \chi_{m,r} \quad (2.47)$$

$$= \sum_{r=1}^R \varphi_r(\mathbf{a}) (\tilde{\mathbf{a}}^\top \otimes I_P) \chi_r \quad (2.48)$$

En notant  $\tilde{A} = (\tilde{\mathbf{a}}^\top \otimes I_P)$ , on a

$$\hat{f}(\mathbf{a}) = \sum_{r=1}^R \varphi_r(\mathbf{a}) \tilde{A} \chi_r \quad (2.49)$$

$$= (\varphi_1(\mathbf{a}) \tilde{A} \cdots \varphi_R(\mathbf{a}) \tilde{A}) \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_R \end{pmatrix} \quad (2.50)$$

$$= (\boldsymbol{\varphi}(\mathbf{a})^\top \otimes \tilde{A}) \mathbf{X} \quad (2.51)$$

Nous pouvons donc en déduire une expression de la sortie du MEX :

$$\hat{f}(\mathbf{a}) = \boldsymbol{\Psi} \mathbf{X} \quad (2.52)$$

Ainsi, nous faisons à nouveau l'analogie avec les parties précédentes au niveau de l'inversibilité de la matrice  $\boldsymbol{\Psi}$ . Donc en notant  $\boldsymbol{\Theta}$  la concaténation des  $\Theta_k$ , le MEX doit minimiser le critère (non régularisé) suivant :

$$\mathcal{J}(\mathbf{X}) = \frac{1}{2} \|\boldsymbol{\Psi} \mathbf{X} - \boldsymbol{\Theta}\|^2 \quad (2.53)$$

## 2.4 Conclusion

Afin de répondre au problème d'apprentissage local, nous avons d'abord considéré les réseaux RBF sous la forme d'un ensemble de modèles locaux pondérés par des RBF. Ce faisant, nous l'avons présenté comme un cas particulier de mélange d'experts. Si dans le cas général, les modèles locaux peuvent avoir une structure quelconque, nous n'avons étudié que le cas de fonctions affines, afin de limiter la complexité paramétrique tout en améliorant la capacité de modélisation. Nous aboutissons donc à un objet linéaire en les paramètres à modifier, ce qui va nous faciliter l'implémentation des différentes méthodes que nous présentons dans la partie suivante.





Deuxième partie

Méthodologies et outils pour la  
retouche



## Présentation des méthodologies développées

CETTE partie constitue l'apport essentiel de la thèse à savoir une synthèse des diverses méthodes permettant de répondre au problème posé. De façon générale, les méthodologies comportent un aspect *hors ligne*, c'est-à-dire réalisable en bureau d'études sans contrainte temporelle, et un aspect *en ligne*, pouvant être appliqué en temps réel pendant un essai en vol. Nous allons ici aborder les différentes étapes de ces méthodes en fonction du besoin exprimé et préciser de manière plus spécifique le rôle joué par les MEX dans ce type de problème.

### De l'apprentissage à la retouche locale

Il s'agit avant toute chose d'identifier clairement l'usage auquel le (ou les) MEX va (vont) être destiné(s) et de faire l'inventaire des ressources à disposition afin de générer des données d'apprentissage et de validation pour ce(s) dernier(s). Dans le cadre des commandes de vol, les besoins identifiés sont la modélisation de coefficients aérodynamiques (cf. §6.3, p. 101), l'estimation de variables de la mécanique du vol et l'interpolation des lois de pilotage. Dans le cadre du présent mémoire, nous n'avons traité que le cas de lois de compensation feed-forward car c'est essentiellement ce type de loi qui est concerné par le problème initialement posé, à savoir l'automatisation du réglage de lois fastidieuses à régler. Le cas des lois feed-back a fait l'objet d'une étude [Cumer et Mouyon, 2003] et fait partie des perspectives de travail.

Chacune de ces applications présente des spécificités d'ordre pratique à différents niveaux que nous allons détailler dans les paragraphes suivants mais toutes font appel à l'identification d'une fonction même si ce n'est pas directement le besoin exprimé. Si cet aspect d'identification de fonction est évident pour la modélisation de coefficients aérodynamiques ou l'estimation de grandeurs aérodynamiques, il est moins apparent pour l'interpolation des lois de pilotage. Au sein d'Airbus, il est d'usage de synthétiser les lois sur des modèles linéarisés et d'interpoler *a posteriori* leurs paramètres. Cette interpolation peut alors se résumer à une identification de fonction comme nous l'avons présenté à la partie précédente. L'utilisation des MEX est d'autant plus justifiée que nous cherchons à retoucher localement ces modèles ou ces lois en temps réel.

Nous rappelons ici le problème 1.2.1 appliqué à un MEX à modèles locaux linéaires et autour duquel nous allons articuler les différents chapitres de cette partie.

PROBLÈME : *Etant donné un ensemble de  $K$  points distincts  $\{\mathbf{a}_k \in \mathcal{A} \mid k = 1 \dots K\}$  et un ensemble correspondant de  $K$  valeurs de référence à estimer  $\{\theta_k \in \mathbb{R} \mid k = 1 \dots K\}$ , déterminer les paramètres du MEX  $\hat{f}(\mathbf{a}, \boldsymbol{\chi})$  tels que :*

$$\boldsymbol{\chi}^* = \arg \min_{\boldsymbol{\chi}} \frac{1}{2} \sum_{k=1}^K (\theta_k - \hat{f}(\mathbf{a}_k, \boldsymbol{\chi}))^2 \quad (\text{II.54})$$

Cette formulation du problème à résoudre induit un certain nombre de questions auxquelles nous allons répondre dans les chapitres suivants :

- Comment générer les données d'apprentissage  $(\mathbf{a}_k, \boldsymbol{\theta}_k)$  ?
- Quelle structure donner à notre MEX (i.e. le nombre et le placement de centres, les RBF utilisées, etc.) et comment utiliser les données générées pour l'apprentissage ?
- Comment réaliser l'apprentissage proprement dit (i.e. résoudre le problème (II.54)) ?

- Comment traiter les questions ci-dessus dans le cadre hors ligne ? Dans le cadre en ligne ?

### Trame générale des méthodes proposées

Une trame générale est proposée à la figure II.1.

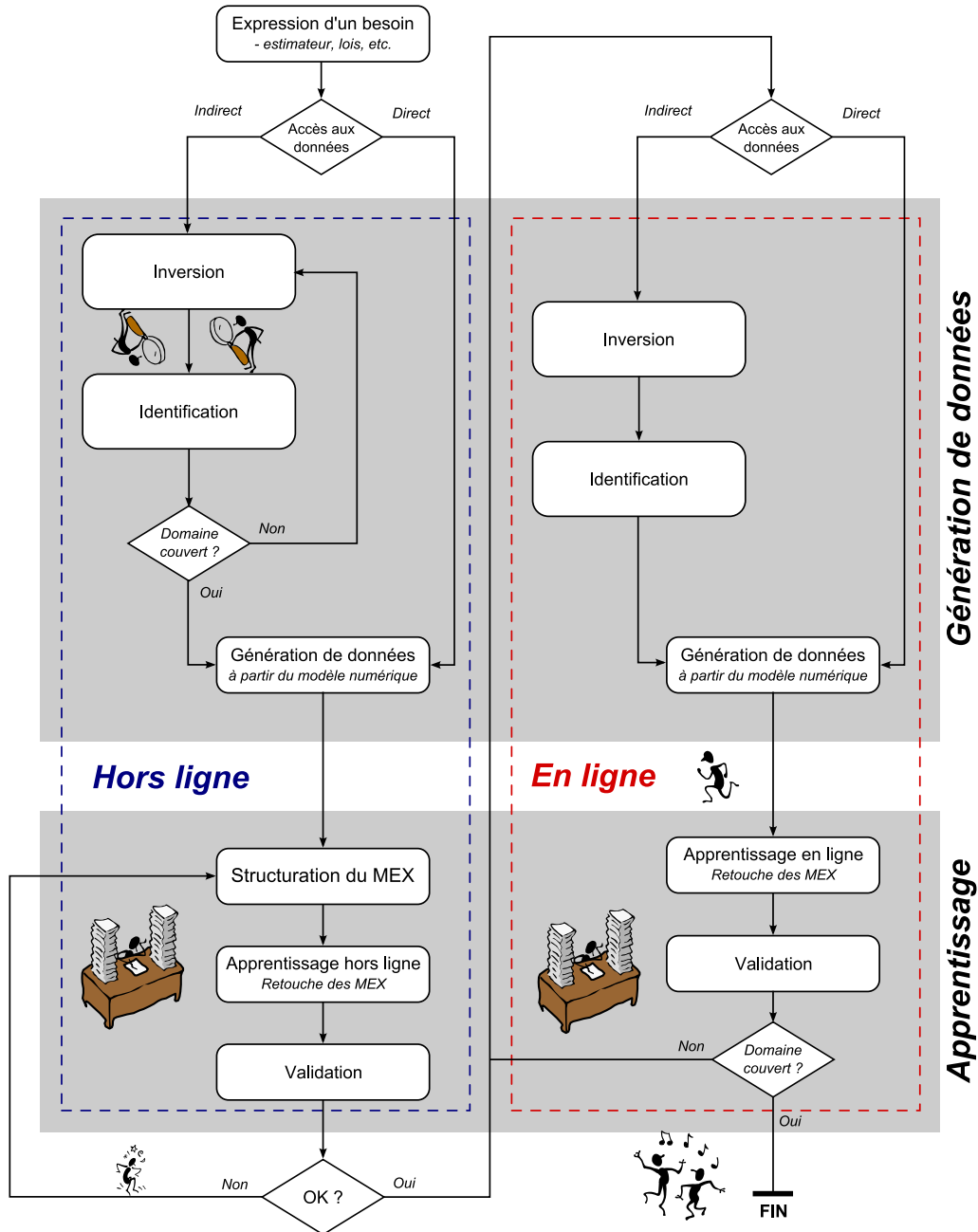


FIG. II.1: Méthodologie générale

Nous distinguons deux parties essentielles que sont la *génération de données* et l'*apprentissage*. Nous allons à présent aborder ces aspects d'un point de vue fonctionnel afin de donner au lecteur une vue d'ensemble suffisamment synthétique pour comprendre la

démarche. Chacune des fonctions sera ensuite traitée dans le détail dans les chapitres suivants.

## 1 Génération des données d'apprentissage (chapitre 3, p. 37)

Avant d'aborder la génération en soi, nous allons définir des termes génériques pour les ressources à disposition pour générer ces données. Nous avons tout d'abord un *modèle d'apprentissage* (ou *de synthèse*) et un *modèle de validation*. Le premier va nous permettre de générer les données dans le cadre hors ligne. Il s'agit par exemple des modèles prévisionnels issus des calculs de CFD mentionnés à l'introduction de ce mémoire. Ils sont généralement entâchés d'erreur par rapport au modèle de validation. Ce dernier sera par la suite considéré comme étant l'avion réel et pourra être utilisé à la fois dans le cadre hors ligne et le cadre en ligne.

Dans le cas de la modélisation de coefficients aérodynamiques ou de l'estimation de grandeur aérodynamique, il n'y a pas de problème particulier lié à la génération de données. En effet, ce sont des grandeurs calculées dans le modèle d'apprentissage.

Cependant, pour l'interpolation des lois de pilotage, il est nécessaire d'effectuer un travail de synthèse préalable à l'aide des modèles afin de calculer les paramètres des lois qui définissent alors nos données d'apprentissage. Nous proposons donc une méthode de synthèse et de retouche par inversion de données temporelles [Ronceray *et al.*, 2008]. La méthode se décompose en deux grandes étapes pour lesquelles nous donnons le fonctionnement dans les cadres hors ligne et en ligne.

Remarquons qu'étant donné que l'avion est un système à paramètres variants, les méthodes ci-dessous sont itérées jusqu'à ce que le correcteur soit défini sur l'ensemble du domaine de fonctionnement de l'avion, appelé domaine de vol, et que sa performance soit satisfaisante.

### 1.1 Inversion pour la synthèse d'une loi feedforward

Nous allons considérer le système présenté en figure II.2.

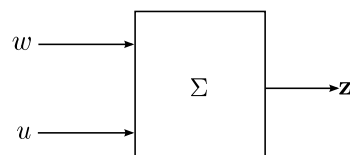


FIG. II.2: Système à inverser

$w$  est un signal que nous qualifierons de perturbation, non pas au sens stochastique du terme mais au sens où il s'agit d'une commande qui peut engendrer un (ou plusieurs) effet(s) secondaire(s) non désirable(s).  $z$  le signal des sorties sur lesquelles se manifeste cet effet et  $u$  est le signal de commande que nous souhaitons utiliser pour compenser cet effet indésirable de façon à obtenir un comportement temporel idéal  $z^*$ , en réponse à une perturbation type  $w_o$ .

**Hors ligne (§3.2.1, p. 39)** La méthode consiste ici à venir identifier le transfert de  $u$  vers  $z$  (sous forme de filtre à RIF) et de l'inverser par déconvolution afin de calculer la commande  $\Delta u^*$  permettant au système de se comporter idéalement (i.e.  $z = z^*$ ) lorsqu'il est soumis à  $w_o$ .

**En ligne (§3.2.2, p. 49)** La méthode consiste là encore à venir identifier le transfert  $\Sigma_2$  de  $u$  vers  $\mathbf{z}$  (sous forme d'état) et à l'asservir selon le schéma présenté en figure II.3.

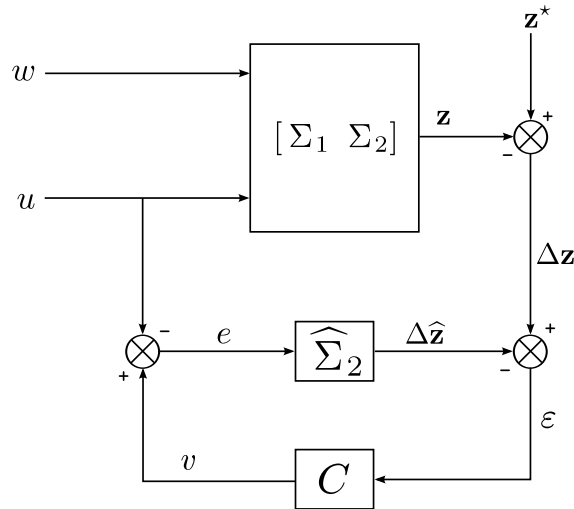


FIG. II.3: Inversion par asservissement

Nous montrerons que le signal  $v$  est une estimation de  $\Delta u^*$  lorsque  $\varepsilon \equiv 0$  et si  $\widehat{\Sigma}_2 \approx \Sigma_2$ . Le correcteur  $C$  pourra être calculé soit par synthèse modale, soit par synthèse LQG.

## 1.2 Identification de la loi

**Hors ligne (§3.3.1, p. 55)** Une fois  $\Delta u^*$  calculé, il faut identifier le correcteur  $K(s, \theta)$  reproduisant au mieux  $\Delta u^*$  avec  $w_o$  en entrée tel que le montre la figure II.4. Nous proposons

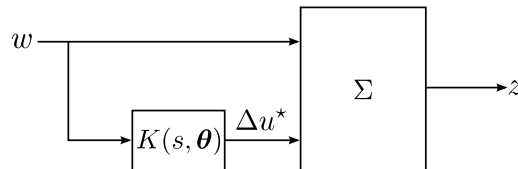


FIG. II.4: Schéma de synthèse

ici deux méthodes pour identifier le correcteur. La première est une approche empirique se basant sur l'observation des caractéristiques de  $\Delta u^*$  pour déterminer une structure de correcteur et sur la minimisation dans l'espace des paramètres  $\theta$  du critère suivant :

$$\mathcal{J}(\theta) = \|\Delta u^*(t) - K(s, \theta) * w_o(t)\|^2.$$

La seconde fait appel à l'utilisation de méthodes classiques d'identification dont nous donnerons quelques exemples.

**En ligne (§3.3.2, p. 57)** L'identification est ici réalisée par une technique de filtrage adaptatif utilisant le signal  $v$  comme référence. En partant du constat que la stabilité du correcteur  $K(s, \theta)$  devient difficile à gérer pour un ordre supérieur à 3, ce dernier est discretisé puis mis sous forme en treillis normalisé. Cette forme permet de définir une condition suffisante très simple de stabilité asymptotique exponentielle, quel que soit

l'ordre du correcteur. Nous donnons un algorithme de mise à jour des paramètres du filtre, appelé SPGL, ainsi qu'une variante permettant une implémentation dans un calculateur « basique ». Les résultats présentés dans ce paragraphe sont tirés de [Regalia, 1992] et [Regalia, 1994].

## 2 L'apprentissage (chapitre 4, p. 67)

### 2.1 Hors ligne

Une fois les données générées, il faut les séparer en un ensemble d'apprentissage et de validation afin de vérifier la capacité de généralisation du MEX. Un pré-traitement est effectué sur ces données afin de les normaliser (§4.1.1, p. 67).

Ensuite, nous donnons plusieurs stratégies de placement des centres des MEX en fonction de connaissances *a priori* et faisons l'inventaire des possibilités de réglage des divers paramètres : choix des fonctions à base radiale et leur normalisation éventuelle, choix de la norme utilisée, taille des fonctions à bases radiale, etc (§4.1.2 à §4.1.4, p. 68). Enfin, une fois la structure du MEX figée, nous calculons ses paramètres en utilisant une décomposition en valeurs singulières (§4.1.2.3, p. 71).

### 2.2 En ligne

Nous proposons deux méthodes pour venir effectuer l'apprentissage en ligne des MEX. La première est une application des moindres carrés récurrents à l'apprentissage d'un MEX (ou d'une combinaison linéaire de MEX) lorsque nous avons à disposition une référence mesurée (§4.2.1, p. 75). La seconde est une modification de l'algorithme SPGL où nous venons mettre à jour les MEX qui modélisent les coefficients du filtre en treillis à la place des coefficients selon la figure II.5.

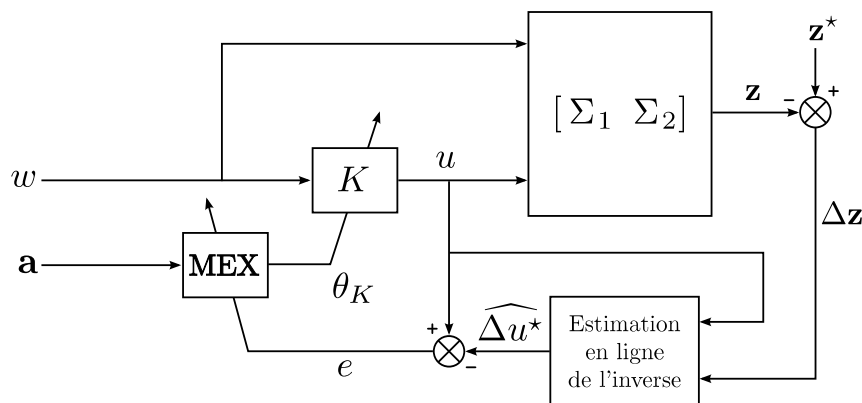


FIG. II.5: Schéma de retouche en ligne d'une loi feed-forward

Cette méthode est la concrétisation de l'idée qui avait été émise dans l'introduction, à savoir de venir modifier directement et en temps-réel la fonction d'interpolation des paramètres de la loi.

## 3 Régularisation (chapitre 5, p. 81)

Nous avons choisi de faire un chapitre consacré à la régularisation car elle intervient dans deux contextes différents : les mélanges d'experts (§2.2, p. 19) et l'inversion hors-



ligne (§3.2.1, p.39). Le fait de donner les détails relatifs à la théorie de la régularisation à ces moments aurait considérablement alourdi la présentation et bruité le propos. Nous donnons donc dans ce chapitre les éléments nécessaires à la compréhension de la notion de régularisation telle qu'elle est utilisée dans ce mémoire (§5.2 à §5.3, p. 82). Nous donnons ensuite deux méthodes pour le calcul du paramètre de régularisation : une heuristique permettant un réglage automatique (§5.4.2, p. 87) et un calcul plus théorique (§5.4.1, p. 85).

## Chapitre 3

# Outils pour la génération des données d'apprentissage

### Sommaire

---

<b>3.1</b>	<b>Formalisation du problème</b>	<b>37</b>
<b>3.2</b>	<b>Inversion</b>	<b>38</b>
3.2.1	Inversion hors ligne	39
3.2.2	Inversion en ligne	49
<b>3.3</b>	<b>Identification</b>	<b>55</b>
3.3.1	Identification hors ligne des paramètres de la loi	55
3.3.2	Identification en ligne des paramètres de la loi	57
<b>3.4</b>	<b>Conclusion</b>	<b>66</b>

---

L'OBTENTION de données d'apprentissage n'est pas systématiquement une formalité comme nous l'avons mentionné lors de la présentation de la partie II et dépend évidemment de l'usage qui va être fait du MEX au sein de l'application considérée. Dans le cas d'une modélisation de coefficient aérodynamique ou de l'estimation d'une grandeur aérodynamique, la génération de données est triviale et ne pose pas de problème particulier, étant donné que ces grandeurs sont directement accessibles depuis les modèles.

Dans un contexte commande de systèmes PV, nous avons vu la nécessité d'un travail préliminaire de synthèse de correcteurs que nous irons interpoler à l'aide de MEX. Ce travail nous permet de générer les paramètres des correcteurs sur un ensemble de points de l'espace paramétrique du système.

Nous rappelons dans ce chapitre la formalisation du problème posé par la génération de données pour interpoler une loi de compensation feed-forward et nous détaillons dans les sections suivantes les outils utilisés pour répondre à ce problème.

### 3.1 Formalisation du problème

Considérons un système  $\Sigma$ , illustré en figure 3.1(a) pour lequel nous adoptons les notations suivantes :  $w(t)$  est un signal d'entrée exogène,  $u(t)$  le signal de commande scalaire et  $z(t)$  le signal de sortie scalaire, afin d'alléger les notations. La généralisation au cas vectoriel sera considérée au paragraphe §3.2.1.3.

Dans [Sentoh et Bryson, 1992], les auteurs passent en revue différentes méthodes de contrôle inverse et proposent une formalisation du problème que nous allons utiliser et étendre par la suite.

PROBLÈME 3.1.1 : *Considérons les schémas suivants :*

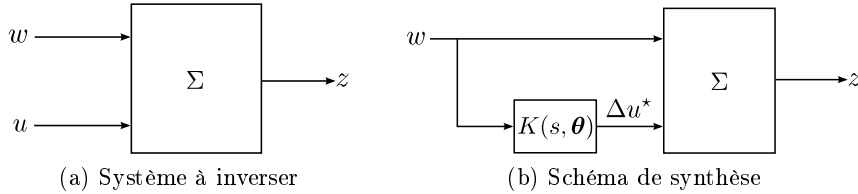


FIG. 3.1: Schémas mis en jeu pour l'inversion

1. Le premier objectif est de calculer la commande  $\Delta u^*(t)$  telle que le système  $\Sigma$ , que l'on donne sous la forme générale suivante,

$$\dot{\mathbf{x}} = f(\mathbf{x}, w, \Delta u^*), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.1)$$

$$z = h(\mathbf{x}, w, \Delta u^*) \quad (3.2)$$

ait sa sortie

$$z(t) = z^*(t) \quad (3.3)$$

où  $z^*(t)$  est connu a priori pour une perturbation donnée  $w_o(t)$ .

2. Le second objectif est de définir une loi de commande reproduisant de façon optimale  $\Delta u^*(t)$ , et utilisant les entrées. Il faut donc trouver un correcteur  $K(s, \theta)$  (cf. figure 3.1(b)) :

$$\dot{\mathbf{x}}_K = f_K(\mathbf{x}_K, w_o, \theta), \quad \mathbf{x}_K(0) = \mathbf{x}_{K0} \quad (3.4)$$

$$z_K = h_K(\mathbf{x}_K, w_o, \theta) \quad (3.5)$$

tel que :

$$z_K(t) = \Delta u^*(t) \quad (3.6)$$

Nous avons donc à résoudre un problème d'*inversion* suivi d'un problème d'*identification*. Nous allons aborder pour chacun d'eux les outils que nous avons mis en place dans le cadre hors ligne et dans le cadre en ligne.

## 3.2 Inversion

L'approche classique de la simulation inverse permet de calculer les signaux de commande pour reproduire un comportement désiré, issu de données réelles ou bien d'un cahier des charges. Une des applications de cette approche est l'évaluation de la faisabilité de manœuvres données, la manœuvrabilité maximale et des qualités de vol [Avanzini *et al.*, 1998]. Un autre domaine d'application intéressant est la conception de lois de pilotage [Boyle et

Chamitoff, 1999], [Sentoh et Bryson, 1992]. Cela a également été utilisé à des fins de validation de programmes de simulations [Grünhagen, 1993], de reconstruction de données non mesurables pour recouper des données d'essais en vol [Bucharles et Gimonet, 2000], [Jean, 2001], [Mouyon et Losser, 2002], [Mouyon et Vacher, 2001] et de retouche de correcteurs [Jean, 2001], [Mouyon *et al.*, 2002], [Mouyon et Losser, 2002].

Le paragraphe §3.2.1 décrit la méthode utilisée dans un contexte hors ligne. Il s'agit d'un algorithme itératif existant que nous avons accéléré en utilisant la particularité des données utilisées. La paragraphe §3.2.2 décrit la méthode utilisée pour estimer cet inverse en ligne. Nous avons utilisé l'asservissement d'un modèle du transfert à inverser sur l'erreur commise par le système.

### 3.2.1 Inversion hors ligne

Dans le problème posé au paragraphe §3.1, le système  $\Sigma$  doit reproduire, ou s'approcher d'une réponse idéale  $z^*$  lorsqu'il est soumis à une perturbation donnée  $w_o$ . Pour cela, la commande  $u$  est utilisée. Soit  $z_o$  la réponse du système  $\Sigma$  à la perturbation  $w_o$ . Formons alors :

$$\Delta z = z^* - z_o \quad (3.7)$$

Faisons à présent l'hypothèse que le transfert  $\Sigma_{u \rightarrow z}$  est un transfert à RIF  $h(t)$  et que la réponse  $z_o$  est linéaire en  $u$ , ce qui est d'autant plus légitime que nous considérons la variation de réponse induite par une variation de l'entrée. Formellement, l'objectif est de retrouver l'entrée  $\Delta u^*$  d'un filtre à RIF  $h$ , connaissant sa sortie  $\Delta z$ . Nous disposons d'un modèle, qui est la réponse  $z_\delta$  du filtre à une entrée impulsionnelle  $\delta$  éventuellement filtrée par un filtre à RIF connu  $f$ . L'utilisation de ce filtre se justifie par le fait que dans la pratique, l'ingénieur aéronautique utilise plutôt un créneau de longueur finie qu'un signal impulsionnel, à des fins d'identification. Si l'amplitude du créneau utilisé est raisonnable et si l'énergie transmise au système est contrôlée par la longueur du créneau, les résultats obtenus sont en général plus précis.

Le problème précédent se pose alors de la façon suivante :

PROBLÈME 3.2.1 : *Résoudre en  $\Delta u^*$  :*

$$\Delta z(t) = h * \Delta u^*(t) \quad (3.8)$$

*connaissant*

$$z_\delta(t) = h * f * \delta(t) \quad (3.9)$$

Effectuons le changement de variable suivant :

$$\Delta u^*(t) = f * e(t) \quad (3.10)$$

$$g(t) = h * f(t) \quad (3.11)$$

Nous obtenons donc le problème équivalent :

PROBLÈME 3.2.2 : *Résoudre en  $e(t)$  :*

$$\Delta z(t) = g * e(t) \quad (3.12)$$

connaissant

$$g(t) = h * f(t) \quad (3.13)$$

et calculer :

$$\Delta u^*(t) = f * e(t) \quad (3.14)$$

A présent, il faut tenir compte du fait que les données que nous allons traiter seront des données discrètes. En fonction de l'instant courant  $n\Delta t$ , le produit de convolution de  $e(t)$  par  $g(t)$  s'écrit :

$$\Delta z(n\Delta t) = \sum_{l=0}^L g(l\Delta t)e((n-l)\Delta t) \quad (3.15)$$

où  $L+1$  est la longueur de la RI  $g(t)$  discrétisée. Nous pouvons réécrire ce produit sous forme matricielle en considérant un horizon de longueur  $N_h$  et faisant l'hypothèse que les  $L$  commandes précédant l'instant initial sont nulles. Le problème se réécrit alors :

PROBLÈME 3.2.3 : Résoudre en  $\mathbf{e}$  :

$$\mathbf{z} = G\mathbf{e} \quad (3.16)$$

connaissant  $G$  et calculer :

$$\mathbf{u} = F\mathbf{e} \quad (3.17)$$

$$(3.18)$$

connaissant  $F$

Ce problème revient à résoudre au sens des moindres carrés :

$$\mathbf{e}^* = \arg \min_{\mathbf{e}} \|\mathbf{z} - G\mathbf{e}\|^2 \quad (3.19)$$

Remarquons que  $G$  est une matrice Toeplitz triangulaire inférieure traduisant un produit de convolution discret [Gray, 2006] :

$$G = \begin{pmatrix} g_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ g_1 & \ddots & \ddots & & & & \vdots \\ \vdots & & \ddots & \ddots & & & \vdots \\ g_L & & & \ddots & \ddots & & \vdots \\ 0 & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & 0 \\ 0 & \cdots & 0 & g_L & \cdots & g_1 & g_0 \end{pmatrix} \quad (3.20)$$

De nombreuses méthodes existent pour résoudre un tel problème. Par la suite, nous allons considérer deux de ces méthodes : l'inversion fréquentielle et l'algorithme de Landweber. Toutefois, le lecteur intéressé trouvera une revue plus complète des méthodes numériques pour résoudre des problèmes aux moindres carrés dans [Björck, 1996].

### 3.2.1.1 Inversion fréquentielle

Cette méthode, utilisée dans [Mouyon et Vacher, 2001], profite de la structure particulière de la matrice  $G$ . Elle permet en particulier de résoudre des problèmes de grande taille, comme nous allons le voir par la suite.

**Transformée de Fourier discrète** En remarquant que la matrice  $G$  est quasi-circulante, la difficulté du calcul de l'inverse peut être contournée par un passage dans la base de Fourier discrète, dans laquelle une matrice circulante est diagonalisable, ce qui permet de réduire la complexité numérique. Effectuons les modifications suivantes :

- $N_h$  doit être une puissance de 2 pour pouvoir utiliser la transformée de Fourier rapide (FFT). Si ce n'est pas le cas, il est possible de le compléter par « zero-padding ».
- $z$  et  $e$  doivent être ensuite  $N_h$ -périodisés
- la matrice  $G$  est augmentée de façon à la rendre circulante :

$$\tilde{G} = \begin{pmatrix} \tilde{G}_{11} & \tilde{G}_{12} \\ \tilde{G}_{21} & \tilde{G}_{22} \end{pmatrix} \quad (3.21)$$

avec :

$$\tilde{G}_{11} = G \quad (3.22)$$

$$\tilde{G}_{12} = \begin{pmatrix} 0 & g_L & \dots & g_1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & g_L \\ 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix} \text{ de taille } N_h \times (L+1) \quad (3.23)$$

$$\tilde{G}_{21} = \begin{pmatrix} 0 & \dots & 0 & g_L & \dots & g_1 \\ \vdots & & \vdots & \ddots & \ddots & \vdots \\ \vdots & & \vdots & & \ddots & g_L \\ 0 & \dots & 0 & \dots & \dots & 0 \end{pmatrix} \text{ de taille } (L+1) \times N_h \quad (3.24)$$

$$\tilde{G}_{22} = \begin{pmatrix} g_0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ g_L & \dots & \dots & g_0 \end{pmatrix} \text{ de taille } (L+1) \times (L+1) \quad (3.25)$$

Le produit de convolution circulaire suivant :

$$z(n\Delta t) = g \circledast e(n\Delta t) \quad (3.26)$$

$$= \sum_{m=0}^{N_h-1} g(m\Delta t) e((n-m)\Delta t) \quad (3.27)$$

$$= \sum_{m=0}^{N_h-1} g_m e_{n-m} \quad (3.28)$$

peut s'écrire après utilisation de la transformée de Fourier discrète :

$$\mathbf{z}_k = \mathbf{g}_k \cdot \mathbf{e}_k, \quad \forall k = 0, \dots, N_h - 1 \quad (3.29)$$

où :

$$\mathbf{x}_k = \sum_{n=0}^{N_h-1} x_n \exp\left(-\frac{2i\pi kn}{N_h}\right) \text{ avec } \mathbf{x} = \mathbf{z}, \mathbf{g} \text{ ou } \mathbf{e}, \quad \forall k = 0, \dots, N_h - 1 \quad (3.30)$$

avec  $i^2 = -1$ .

Ce problème de déconvolution circulaire peut ainsi être diagonalisé et facilement résolu. Nous obtenons alors :

$$\mathbf{e}_k^* = \frac{\mathbf{g}_k^* \mathbf{z}_k}{\mathbf{g}_k^* \mathbf{g}_k + \epsilon}, \quad \forall k = 0, \dots, N_h - 1 \quad (3.31)$$

où  $\epsilon$  un réel positif empêchant la division par zéro.

**Inversion fréquentielle régularisée** Un problème du type de celui posé en (3.19) étant généralement mal posé, nous régularisons le problème en introduisant un opérateur de régularisation afin de le transformer en un problème bien posé [Tikhonov et Arsenin, 1977] admettant une solution unique pour toutes les mesures possibles dans l'espace d'observation. Le critère ainsi minimisé devient (cf. §5.2, p. 82) :

$$\mathcal{J}_\lambda(\mathbf{e}) = \|\mathbf{z} - G\mathbf{e}\|^2 + \lambda\Omega(\mathbf{e}) \quad (3.32)$$

$$\text{avec } \Omega(\mathbf{e}) = \|D\mathbf{e}\|^2 \quad (3.33)$$

où  $D$  est un opérateur linéaire de régularisation correspondant à un filtrage linéaire<sup>1</sup> et  $\lambda$  le paramètre de régularisation à déterminer.

La solution régularisée de ce problème s'écrit alors :

$$\mathbf{e}_\lambda^* = \left(G^T G + \lambda D^T D\right)^{-1} G^T \mathbf{z} \quad (3.34)$$

Une fois les différents vecteurs et matrices traités pour être passés dans la base de Fourier discrète, nous avons les égalités suivantes (attention  $G \neq \mathbf{G}$ ) :

$$\tilde{G} = F \mathbf{G} F^{-1}, \quad \tilde{D} = F \mathbf{D} F^{-1} \quad (3.35)$$

où  $F$  désigne la matrice de passage dans la base de Fourier discrète et où  $\tilde{\mathbf{G}}$ , respectivement  $\tilde{\mathbf{D}}$ , est diagonale d'éléments  $\tilde{\mathbf{g}}_k$ , respectivement  $\tilde{\mathbf{d}}_k$ ,  $k = 0 \dots N_h - 1$ .

La transformée de Fourier de la solution régularisée du problème s'écrit :

$$\tilde{\mathbf{e}}_\lambda^* = \left(\tilde{\mathbf{G}}^* \tilde{\mathbf{G}} + \lambda \tilde{\mathbf{D}}^* \tilde{\mathbf{D}}\right)^{-1} \tilde{\mathbf{G}}^* \tilde{\mathbf{z}} \quad (3.36)$$

ce qui peut être réécrit en ne considérant que les composantes diagonales :

$$\tilde{e}_{\lambda,k}^* = \frac{\tilde{g}_k}{|\tilde{g}_k|^2 + \lambda |\tilde{d}_k|^2} \tilde{z}_k \quad (3.37)$$

Il ne reste dès lors qu'à appliquer la transformée de Fourier inverse à  $\tilde{\mathbf{e}}_\lambda^*$  et à le tronquer pour obtenir le signal souhaité.

<sup>1</sup> $D$  est donc une matrice de Toeplitz triangulaire inférieure dont la première colonne est la réponse impulsionnelle du filtre de régularisation

**Limites de l'inversion fréquentielle** L'inversion fréquentielle régularisée assure l'obtention rapide d'un résultat, non nécessairement pertinent. Nous nous proposons d'étudier la cause de ce problème. Considérons à nouveau la matrice  $\tilde{G}$ , telle que donnée aux équations (3.21) à (3.25).

Supposons que  $g_0$  et  $g_L$  soient non nuls. Les blocs triangulaires de  $\tilde{G}$  sont donc inversibles ce qui nous permet d'analyser la solution régularisée  $\tilde{\mathbf{e}}_\lambda^*$ .

A faible paramètre de régularisation,  $\tilde{\mathbf{e}}_\lambda^*$  est proche de la solution non régularisée  $\tilde{\mathbf{e}}^*$  de  $\tilde{\mathbf{z}} = \tilde{G}\tilde{\mathbf{e}}$ . Nous pouvons alors faire les remarques suivantes sur ses  $L$  dernières valeurs :

- Si ces  $L$  composantes sont non nulles, la solution  $\mathbf{e}^*$  est entachée d'erreur provenant du fait que l'erreur corrigée n'est pas  $\mathbf{z}$  mais

$$\mathbf{z} - G_{12} \begin{pmatrix} \tilde{e}_{N_h+1} \\ \vdots \\ \tilde{e}_{N_h+L+1} \end{pmatrix} \quad (3.38)$$

- Si les  $L$  dernières composantes de  $\tilde{\mathbf{e}}^*$  sont forcées à 0, par construction de la matrice  $\tilde{G}$  les  $L$  dernières composantes de  $\mathbf{e}^*$  sont également forcées à 0, ce qui affectera la qualité du résultat.

Nous pouvons donc voir que pour des faibles valeurs du paramètres de régularisation  $\lambda$ , la solution régularisée obtenue est entachée d'erreur sur  $L$  composantes. Ces effets de bord peuvent être tolérés si  $L$  est petit devant  $N_h$ , ce qui dans notre cas n'est pas nécessairement vrai.

### 3.2.1.2 Inversion itérative de Landweber

L'inversion fréquentielle décrite précédemment présente des faiblesses, dues aux diverses approximations réalisées, notamment l'hypothèse de convolution circulaire qui entraîne l'apparition d'effets de bords au début et à la fin de la solution temporelle calculée. Leur longueur est de l'ordre de celle de la RI du filtre  $h$ .

Nous proposons ici une méthode itérative à la résolution du problème inverse. Cette méthode repose sur le théorème suivant [Landweber, 1951] :

**THÉORÈME 3.2.1 :** *Soient  $A : \mathcal{A}_1 \rightarrow \mathcal{A}_2$  un opérateur borné et  $\mathbf{b} \in \text{Im } A \oplus (\text{Im } A)^\perp$ . La suite des itérations :*

$$\mathbf{x}(0) = 0 \quad (3.39)$$

$$\mathbf{x}(n+1) = \mathbf{x}(n) + \alpha A^\top (\mathbf{b} - A\mathbf{x}(n)), \text{ avec } 0 < \alpha < \frac{2}{\|A^\top A\|_2} \quad (3.40)$$

*converge vers la solution inverse généralisée de  $A\mathbf{x} = \mathbf{b}$  et où la norme 2 d'un opérateur linéaire  $A$  est donnée à l'annexe A, p. 157.*

Cet algorithme peut être considéré comme un algorithme du gradient pour la minimisation du critère des moindres carrés  $\mathcal{J}(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|^2$ , dont le gradient est  $\nabla \mathcal{J}(\mathbf{x}) = 2A^\top(A\mathbf{x} - \mathbf{b})$ .

*Démonstration.* Soit  $\mathbf{x}^*$  la solution de l'équation normale  $A^\top \mathbf{b} = A^\top A\mathbf{x}$  et  $\Delta \mathbf{x}(n) = \mathbf{x}(n) -$



$\mathbf{x}^*$ . Nous avons :

$$\Delta \mathbf{x}(n+1) = \Delta \mathbf{x}(n) + \alpha A^\top (\mathbf{b} - A\mathbf{x}(n)) \quad (3.41)$$

$$= \Delta \mathbf{x}(n) + \alpha \left( A^\top A \mathbf{x}^* - A^\top A \mathbf{x}(n) \right) \quad (3.42)$$

$$= \left( I - \alpha A^\top A \right) \Delta \mathbf{x}(n) \quad (3.43)$$

Pour assurer la convergence de  $\Delta \mathbf{x}(n)$  vers 0, il faut vérifier :

$$\left\| I - \alpha A^\top A \right\|_2 < 1 \quad (3.44)$$

Pour cela, nous allons considérer les valeurs singulières de  $A = U\Sigma V^*$  où  $U$  et  $V$  sont des matrices unitaires et où  $\Sigma$  est carrée de diagonale :

$$\begin{pmatrix} \sigma_1 & \dots & \sigma_l & 0 & \dots & 0 \end{pmatrix} \quad (3.45)$$

où les  $\sigma_i$  sont les valeurs singulières classées par ordre décroissant. Il vient donc :

$$\left\| I - \alpha A^\top A \right\|_2 \leq \max_j |1 - \alpha \sigma_j(A)^2| \quad (3.46)$$

$$\Leftrightarrow \begin{cases} 1 - \alpha \bar{\sigma}(A)^2 > -1 \\ 1 - \alpha \underline{\sigma}(A)^2 < 1 \end{cases} \quad (3.47)$$

$$\Leftrightarrow 0 < \alpha < \frac{2}{\bar{\sigma}(A)^2} = \frac{2}{\|A^\top A\|_2} \quad (3.48)$$

□

Notons que la convergence suivant les directions  $\mathbf{v}_i$  pour lesquelles  $\sigma_i$  est petit devant  $\bar{\sigma}$  est très lente.

L'inconvénient de cette méthode est que le nombre d'itérations nécessaires pour obtenir une approximation satisfaisante de la solution régularisée du problème, peut être parfois trop important pour définir un algorithme numériquement efficace. Des travaux sur l'accélération de cette méthode ont été réalisés [Egger, 2005], [Hanke, 1991] et rentrent dans le cadre plus général des méthodes semi-itératives (ou méthodes d'accélération polynomiale.) La méthode de Landweber est en fait un cas particulier de ce type de méthodes.

### 3.2.1.3 Application de l'algorithme au cas SIMO

Nous cherchons ici à calculer l'entrée  $E$  d'un système afin de reproduire au mieux les  $P$  sorties  $\mathbf{z}_p$  du système, vis-à-vis de réponses idéales  $\mathbf{z}_p^*$ . Reformulons le critère (3.32) :

$$\mathcal{J}_\lambda(\mathbf{e}) = \mathcal{J}_s(\mathbf{e}) + \lambda \mathcal{J}_r(\mathbf{e}) \quad (3.49)$$

avec

$$\mathcal{J}_s(\mathbf{e}) = \sum_{p=1}^P \omega_p^2 \|\mathbf{z}_p - G_p \mathbf{e}\|^2 \quad (3.50)$$

$$\mathcal{J}_r(\mathbf{e}) = \|D\mathbf{e}\|^2 \quad (3.51)$$

en notant  $\omega_p$  la pondération sur la  $p^{\text{ème}}$  sortie et  $G_p$  la réponse impulsionnelle du transfert de l'entrée vers cette même sortie. Nous pouvons réécrire le critère de la façon suivante :

$$\mathcal{J}_\lambda(\mathbf{e}) = \left\| \begin{pmatrix} \omega_1 \mathbf{z}_1 \\ \vdots \\ \omega_P \mathbf{z}_P \\ 0 \end{pmatrix} - \begin{pmatrix} \omega_1 G_1 \\ \vdots \\ \omega_P G_P \\ \sqrt{\lambda} D \end{pmatrix} \mathbf{e} \right\|^2 \quad (3.52)$$

$$= \|\mathbf{y} - G_\lambda \mathbf{e}\|^2 \quad (3.53)$$

L'algorithme de Landweber s'écrit alors :

$$\mathbf{e}(0) = 0 \quad (3.54)$$

$$\mathbf{e}(n+1) = \mathbf{e}(n) + \alpha G_\lambda^\top (\mathbf{y} - G_\lambda \mathbf{e}(n)) \quad (3.55)$$

avec

$$0 < \alpha < \frac{2}{\|G_\lambda^\top G_\lambda\|} \quad (3.56)$$

Nous formulons la proposition suivante, généralisant un résultat donné dans [Jean, 2001] à un opérateur de régularisation quelconque :

**PROPOSITION 3.2.1 :** *L'utilisation du critère régularisé dans l'algorithme de Landweber permet d'assurer une vitesse de convergence minimale fonction de  $\lambda$  et de l'opérateur de régularisation  $D$ . Cette vitesse minimale est donnée par :*

$$v_{min} = \lambda \underline{\sigma}(D)^2$$

où  $\underline{\sigma}^2(D)$  est la plus petite valeur singulière de  $D$ .

*Vérification.* Pour le montrer nous allons considérer l'expression de la matrice  $G_\lambda^\top G_\lambda$  :

$$G_\lambda^\top G_\lambda = \sum_{p=1}^P \omega_p^2 G_p^\top G_p + \lambda D^\top D \quad (3.57)$$

Or, étant donné que les matrices  $G_p^\top G_p$  pour tout  $p = 1, \dots, P$  et  $D^\top D$  sont symétriques, nous pouvons utiliser le théorème de Weyl cité ci-après [Franklin, 2000] :

**THÉORÈME DE WEYL :** *Soient  $A$  et  $B$  deux matrices hermitiennes complexes de dimension  $N$ . Soient  $\lambda_i(A)$ ,  $\lambda_j(B)$  et  $\lambda_m(A+B)$ , les valeurs propres respectives de  $A$ ,  $B$  et  $A+B$ , rangées par ordre croissant. Nous avons alors :*

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_N(B), \quad \forall k = 1, \dots, N \quad (3.58)$$

D'après ce théorème, nous pouvons déduire les inégalités suivantes sur les valeurs propres de  $G_\lambda^\top G_\lambda$  :

$$\lambda_j(G_\lambda^\top G_\lambda) = \sigma_j(G_\lambda)^2 \geq \sigma_j \left( \sum_{p=1}^P \omega_p^2 G_p^\top G_p \right)^2 + \lambda \underline{\sigma}(D)^2 \quad (3.59)$$

$$\geq \lambda \underline{\sigma}(D)^2, \quad \forall j = 1, \dots, N_h \quad (3.60)$$

Nous pouvons donc voir qu'une vitesse de convergence minimale sera garantie suivant toutes les directions si  $D^T D$  est définie strictement positive ou si l'opérateur de régularisation  $D$  n'a pas de valeur singulière nulle.  $\square$

Remarquons que les dernières conditions sont réalisés pour des opérateurs usuels tels que l'identité et la dérivation discrète.

**Détermination du pas  $\alpha$**  Le calcul exact de la norme de  $G_\lambda^T G_\lambda$  nécessaire à la détermination du pas d'adaptation  $\alpha$  demanderait un effort de calcul inconsidéré au regard de l'application. Le calcul d'un majorant de cette norme paraît plus judicieux afin de déterminer un minorant de  $\alpha_{max} = 2 / \|G_\lambda^T G_\lambda\|$ .

Nous pouvons aisément déterminer un premier majorant :

$$\|G_\lambda^T G_\lambda\|_2 \leq \text{trace}(G_\lambda^T G_\lambda) \leq N_h \left( \lambda + \sum_{p=1}^P \omega_p^2 \sum_{l=0}^L g_p(l)^2 \right) \quad (3.61)$$

Cependant, un majorant moins « grossier » peut être trouvé en remarquant que les  $G_p$  sont des matrices carrées Toeplitz triangulaires inférieures, traduisant un produit de convolution. Là encore, il s'agit d'un résultat tiré de [Jean, 2001], généralisé à un opérateur  $D$  quelconque.

PROPOSITION 3.2.2 : *Un meilleur majorant que celui proposé en (3.61), du paramètre  $\alpha$  de l'algorithme de Landweber utilisant un critère régularisé est donné par :*

$$\bar{\alpha} = \frac{2}{\lambda \bar{\sigma}(D)^2 + \sum_{p=1}^P \omega_p^2 \left( \sum_{k=0}^{N_h-1} |g_p(k)| \right)^2}$$

où  $\bar{\sigma}(D)$  est la plus grande valeur singulière de  $D$ .

*Vérification.* Nous avons :

$$G_p = \sum_{k=0}^{N_h-1} g_p(k) \mathcal{A}^k, \quad \forall p = 1, \dots, P \quad (3.62)$$

où  $\mathcal{A}$  est une matrice nilpotente inférieure de taille  $N_h$  telle que  $\mathcal{A}^{N_h} = 0$ .

Or  $\|\mathcal{A}\|_2 = 1$  donc nous pouvons écrire :

$$\|G_p\|_2 \leq \sum_{k=0}^{N_h-1} |g_p(k)| \|\mathcal{A}^k\|_2 \leq \sum_{k=0}^{N_h-1} |g_p(k)| \quad (3.63)$$

Nous en déduisons alors un majorant de la norme de  $G_\lambda^T G_\lambda$  :

$$\|G_\lambda^T G_\lambda\|_2 \leq \lambda \|D\|_2^2 + \sum_{p=1}^P \omega_p^2 \|G_p\|_2^2 \quad (3.64)$$

$$\leq \lambda \bar{\sigma}(D)^2 + \sum_{p=1}^P \omega_p^2 \left( \sum_{k=0}^{N_h-1} |g_p(k)| \right)^2 \quad (3.65)$$

Un majorant de  $\alpha$  aisément calculable est donc :

$$\bar{\alpha} = \frac{2}{\lambda\bar{\sigma}(D)^2 + \sum_{p=1}^P \omega_p^2 \left( \sum_{k=0}^{N_h-1} |g_p(k)| \right)^2} \quad (3.66)$$

□

Nous choisissons donc pour l'algorithme de Landweber  $0 < \alpha \leq \bar{\alpha}$ .

**Condition d'arrêt de l'algorithme** Nous pouvons déterminer a priori une condition d'arrêt de l'algorithme en considérant la suite (3.43) qui traduit l'évolution de l'erreur entre la solution régularisée des moindres carrés et la solution calculé à la  $n^{\text{ème}}$  itération. En appliquant cette équation au cas considéré ici avec  $\alpha = \bar{\alpha}/2$ , nous pouvons écrire :

$$\Delta \mathbf{e}(0) = -\mathbf{e}_\lambda^* \quad (3.67)$$

$$\Delta \mathbf{e}(n) = \left( I - \alpha G_\lambda^\top G_\lambda \right)^n \Delta \mathbf{e}(0) \quad (3.68)$$

A présent, nous cherchons à calculer le nombre d'itérations nécessaires  $n_{opt}$  pour que l'erreur commise dans toutes les directions de convergence, définies par les vecteurs singuliers à droite de  $G_\lambda$ , soit inférieure à un seuil  $\mu$  fois l'erreur initiale  $\Delta \mathbf{e}_0$ , ce qui donne la condition suivante :

$$\left( 1 - \alpha \sigma_i(G_\lambda)^2 \right)^{n_{opt}} \leq \mu, \quad \forall i = 1, \dots, N_h \quad (3.69)$$

Or d'après l'inégalité (3.60), nous avons :

$$\sigma_i(G_\lambda)^2 \geq \lambda \underline{\sigma}(D)^2, \quad \forall i = 1, \dots, N_h$$

Donc pour que (3.69) soit vérifiée, il suffit que :

$$\left( 1 - \alpha \sigma_i(G_\lambda)^2 \right)^{n_{opt}} \leq \left( 1 - \alpha \lambda \underline{\sigma}(D)^2 \right)^{n_{opt}} \leq \mu \quad (3.70)$$

$$n_{opt} \geq \frac{\ln(\mu)}{\ln(1 - \alpha \lambda \underline{\sigma}^2(D))} \quad (3.71)$$

Cette condition permet d'arrêter l'algorithme très efficacement aux forts paramètres de régularisation. Par contre la garantie de convergence à un seuil donné pour de faibles valeurs de  $\lambda$  imposerait un nombre d'itération bien trop important. Il est alors envisageable de se fixer un nombre d'itérations maximum  $n_{max}$  à comparer à  $n_{opt}$ . Il est également possible d'arrêter l'algorithme lorsque  $\|\Delta \mathbf{e}(n)\|$  est inférieur à un seuil  $\mu' \ll 1$ .

### 3.2.1.4 Accélération du calcul

L'inconvénient majeur de l'algorithme de Landweber est le temps de calcul lorsque  $N_h$  est grand. En effet, à chaque itération, nous devons calculer  $G_\lambda^\top G_\lambda \mathbf{e}(n)$  et  $G_\lambda^\top \mathbf{y}$ . Nous présentons dans la suite un moyen d'accélérer le calcul du produit en utilisant la forme particulière des matrices utilisées [Björck, 1996]. Nous allons donc considérer un produit matrice vecteur  $\mathbf{x} = A\mathbf{b}$ , où  $A$  est une matrice Toeplitz :

$$A = \begin{pmatrix} a_0 & a_1 & \dots & a_n \\ a_{-1} & a_0 & \dots & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{-m} & a_{-m+1} & \vdots & a_0 \end{pmatrix} \in \mathbb{R}^{(m+1) \times (n+1)} \quad (3.72)$$

Cette propriété de  $A$  permet de réduire le produit matrice/vecteur  $A\mathbf{x}$  à une convolution et de le calculer en utilisant la transformée de Fourier rapide (FFT) en  $O(n \ln(n))$  opérations. Ceci est également vrai lorsque  $A$  est une matrice Toeplitz rectangulaire [O'Leary et Simmons, 1981].

Soit  $\tilde{A}$  la matrice circulaire carrée obtenue à partir de  $A$  :

$$\tilde{A} = \left( \begin{array}{c|c} A & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right)$$

$$= \left( \begin{array}{cccc|cccc} a_0 & a_1 & \dots & a_n & a_{-m} & \dots & a_{-1} & \\ a_{-1} & a_0 & \dots & a_{n-1} & a_n & \dots & a_{-m} & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ a_{-m} & a_{-m+1} & \dots & a_0 & a_1 & \dots & a_n & \\ \hline a_n & a_{-m} & \dots & a_{-1} & a_0 & \dots & a_{n-1} & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ a_1 & a_2 & \dots & a_{-m} & a_{-m+1} & \dots & a_0 & \end{array} \right) \in \mathbb{R}^{p \times p}$$

où  $p = m + n - 1$ . Les vecteurs propres de  $\tilde{A}$  sont les vecteurs de Fourier, c'est-à-dire les colonnes de la matrice  $F = (f_{jk}) \in \mathbb{R}^{p \times p}$  :

$$f_{jk} = \frac{1}{\sqrt{p}} e^{2\pi i j k / p} \quad 0 \leq j, k \leq p \quad i^2 = -1$$

Les valeurs propres sont les éléments de la transformée de Fourier de la première colonne de  $\tilde{A}$  :

$$F (a_0 \ \dots \ a_{-m} \ a_n \ \dots \ a_1)^T = (\lambda_1 \ \dots \ \lambda_p)^T \quad (3.73)$$

La matrice  $\tilde{A}$  s'écrit alors :

$$\tilde{A} = F \Lambda F^* \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_p)$$

Pour calculer le produit  $\mathbf{x} = A\mathbf{b}$ , il suffit d'ajouter des zéros au vecteur  $\mathbf{b}$  et de calculer :

$$\mathbf{c} = \tilde{A} \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} = F \Lambda F^* \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} I_m & 0 \end{pmatrix} \mathbf{c}$$

Il suffit d'utiliser deux FFT et une multiplication avec une matrice diagonale ce qui ramène le nombre d'opérations de  $O(n^2)$  à  $O(n \ln(n))$ . Comme la matrice transposée de  $A$  est également Toeplitz, nous pouvons utiliser une méthode similaire pour calculer  $A^T \mathbf{b}$ . Enfin pour un calcul de type  $A^T A \mathbf{b}$ , il suffit de diviser l'opération en deux multiplications matrice/vecteur et de les traiter comme nous venons de le présenter.

### 3.2.1.5 Conclusion

Nous venons de présenter une façon de répondre à la première partie du problème 3.1.1, qui nous permet de déterminer un signal optimal de commande, par rapport à un comportement spécifié au préalable. Les travaux ont essentiellement consisté à améliorer la vitesse globale de l'algorithme en réduisant le nombre d'opération et en maximisant le pas d'adaptation. Nous allons à présent détailler la génération en ligne du signal  $\Delta u^*$ .

### 3.2.2 Inversion en ligne

Cette partie est très liée à la partie précédente. En effet, l'idée est de générer en temps-réel le signal inverse  $\Delta u^*$ . Nous avons considéré plusieurs approches au préalable et notamment la commande adaptative inverse [Hizal, 1999], [Widrow et Walach, 1995], [Widrow et Plett, 1996]. Bien que très attrayante par sa structure et sa solide base théorique, la complexité d'implémentation nous a incités à considérer d'autres solutions. En effet, la commande adaptative inverse utilise abondamment des filtres en cascade d'ordre élevé afin de générer la réponse impulsionnelle identifiée et les signaux inverses estimés. Dans notre contexte, il faudrait interpoler et adapter cette quantité imposante de paramètres en temps-réel ce qui ne respecte pas notre contrainte de complexité.

Nous sommes donc revenus à une utilisation « détournée » des techniques de commande dites classiques, qui consiste à asservir le modèle  $\widehat{\Sigma}_2$  du transfert de  $u$  à  $z$  sur une référence calculée à partir l'erreur la réponse  $z(t)$  du système et la réponse désirée  $z^*(t)$ .

Considérons le système initial, illustré figure 3.1(a), intégré au schéma présenté figure 3.2.

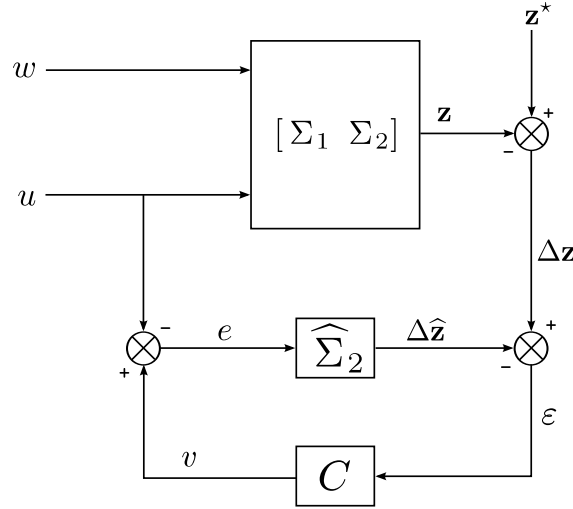


FIG. 3.2: Inversion par asservissement

PROPOSITION 3.2.3 : En reprenant les notations des paragraphes précédents, le schéma précédent permet d'estimer  $\Delta u^*$  en temps réel via la grandeur  $v$ , si l'asservissement  $C$  permet d'assurer  $\varepsilon \equiv 0$  et si  $\widehat{\Sigma}_2 \approx \Sigma_2$ .

Vérification. L'erreur de sortie est la suivante :

$$\varepsilon = (\mathbf{z}^* - \mathbf{z}) - \Delta \widehat{\mathbf{z}} \quad (3.74)$$

$$= \left( \mathbf{z}^* - (\Sigma_1 w + \Sigma_2 u) \right) - \widehat{\Sigma}_2 (v - u). \quad (3.75)$$

Si nous effectuons un asservissement afin d'annuler cette erreur, nous pouvons écrire indépendamment de la structure choisie pour le correcteur :

$$v = \widehat{\Sigma}_2^{-1} (\mathbf{z}^* - \Sigma_1 w + (\widehat{\Sigma}_2 - \Sigma_2) u) \quad (3.76)$$

et si  $\widehat{\Sigma}_2 \approx \Sigma_2$

$$v \approx \widehat{\Sigma}_2^{-1}(z^* - \Sigma_1 w) \quad (3.77)$$

$$\approx \Delta u^*. \quad (3.78)$$

Sous l'hypothèse que  $\widehat{\Sigma}_2$  est suffisamment représentatif de  $\Sigma_2$ , cette structure permet d'estimer en temps-réel le signal inverse pour  $\varepsilon = 0$ . La performance de l'estimation est donc limitée par la dynamique du correcteur. La notation  $\widehat{\Sigma}_2^{-1}$  n'est utilisée ici que pour donner un caractère didactique à la vérification. Nous verrons par la suite que nous n'avons pas besoin de calculer explicitement l'inverse de  $\widehat{\Sigma}_2$ .  $\square$

Cette hypothèse est compatible avec notre contexte mais n'est pas vraiment restrictive car il est raisonnable de considérer que le transfert de commande a fait l'objet d'une identification préalable rigoureuse. Afin d'assurer la validité de la proposition 3.2.3, l'asservissement doit être suffisamment rapide afin de réduire les phases transitoires où  $\varepsilon \neq 0$ . Un fort dépassement peut être envisagé s'il est bref, dans la mesure où l'adaptation pourra nous permettre de « filtrer » ces écarts brusques.

Etant donné que nous travaillons dans un contexte à paramètres variants ( $\Sigma_2(\mathbf{a})$ ) et que nous disposons uniquement d'un ensemble discret de données  $\{\widehat{\Sigma}_2(\mathbf{a}_k)\}_{k=1,\dots,K}$ , nous avons choisi de faire la synthèse des correcteurs sur chaque point puis d'interpoler les paramètres a posteriori, à l'aide de réseaux de neurones à base radiale. Une analyse de stabilité de la commande linéarisée des systèmes non-linéaires, séquencée par des RBF est donnée dans [Chai *et al.*, 1996], d'après [Rugh, 1991]. Nous proposons par la suite deux méthodes pour réaliser cet asservissement.

### 3.2.2.1 Réalisation par synthèse modale statique

Une première méthode consiste à synthétiser ce correcteur hors ligne par placement de structure propre par retour statique de sortie [Magni, 2002], en augmentant le système de façon à isoler les gains du correcteur (cf. figure 3.3). Pour cela, il nous faut des modèles  $\widehat{\Sigma}_2$  sous forme d'état, que nous estimons à partir des réponses impulsionnelles calculées précédemment à l'aide de la méthode des sous-espaces [De Cock et De Moor, 2003], [Overschee et Moor, 1996].

L'augmentation du système se fait en deux temps :

- Ajout des sorties intégrales et pseudo-dérivées (un dérivateur pur n'étant pas causal, il est remplacé par un filtre  $H_d(s) = s/(1 + \tau s)$ ) et modification des matrices en conséquence. Si le système s'écrit sous la forme suivante :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3.79)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (3.80)$$

le système augmenté s'écrit :

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\mathbf{u} \quad (3.81)$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{C}}\tilde{\mathbf{x}} + \tilde{\mathbf{D}}\mathbf{u} \quad (3.82)$$

avec :

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \mathbf{x}_i \\ \mathbf{x}_d \end{pmatrix}, \quad \tilde{\mathbf{y}} = \begin{pmatrix} \mathbf{y} \\ \int \mathbf{y} \\ \hat{\mathbf{y}} \end{pmatrix} \quad (3.83)$$

$$\tilde{\mathbf{A}} = \begin{pmatrix} A & 0 & 0 \\ C & 0 & 0 \\ C & 0 & \frac{1}{\tau}I \end{pmatrix}, \quad \tilde{\mathbf{B}} = \begin{pmatrix} B \\ D \\ D \end{pmatrix} \quad (3.84)$$

$$\tilde{\mathbf{C}} = \begin{pmatrix} C & 0 & 0 \\ 0 & I & 0 \\ \frac{1}{\tau}C & 0 & -\frac{1}{\tau^2}I \end{pmatrix}, \quad \tilde{\mathbf{D}} = \begin{pmatrix} D \\ 0 \\ \frac{1}{\tau}D \end{pmatrix} \quad (3.85)$$

- Calcul de la réalisation minimale du système : une telle construction va avoir pour effet de dupliquer les modes intégraux et de pseudo-dérivation autant de fois qu'il y a de sorties. Le fait de ramener le système à sa réalisation minimale va nous permettre de ne garder qu'un seul mode de chaque, le système étant SIMO. Pour notre exemple, cette réduction est essentielle car elle va nous permettre d'avoir davantage de sorties que d'états, ce qui permet de placer tous les états.

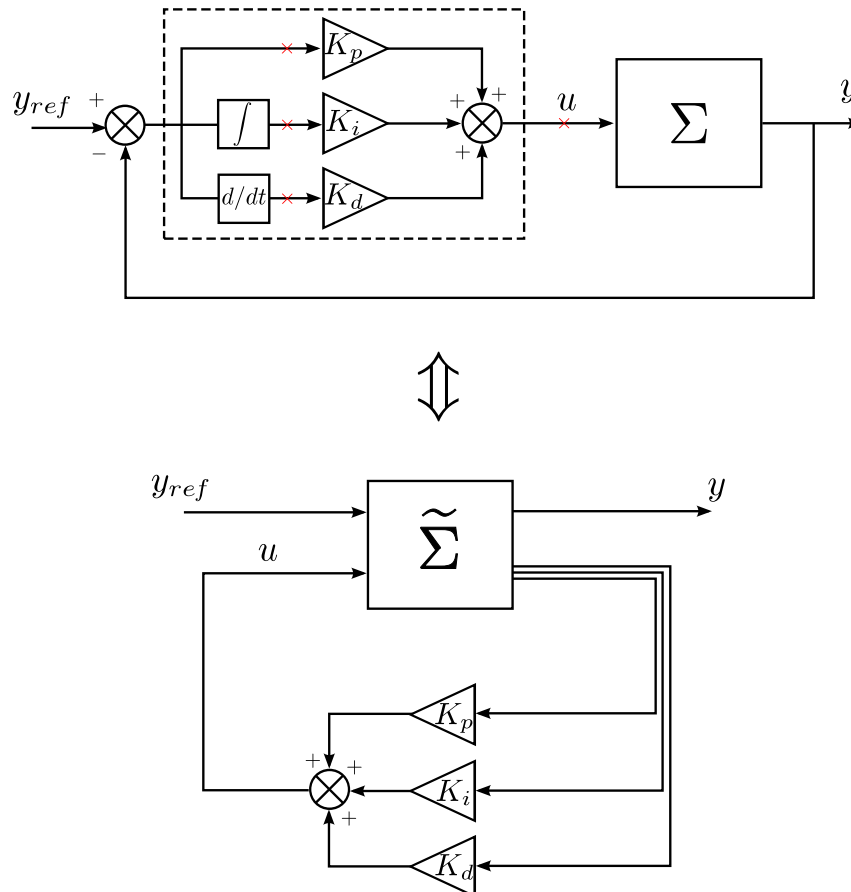


FIG. 3.3: Augmentation du système pour la synthèse modale statique



### 3.2.2.2 Réalisation par synthèse LQG

Dans cette partie, nous avons utilisé une méthode de synthèse de l'asservissement inverseur qui est une synthèse LQG avec modèle d'erreur avec contrôle intégral dont la structure finale est schématisée figure 3.4.

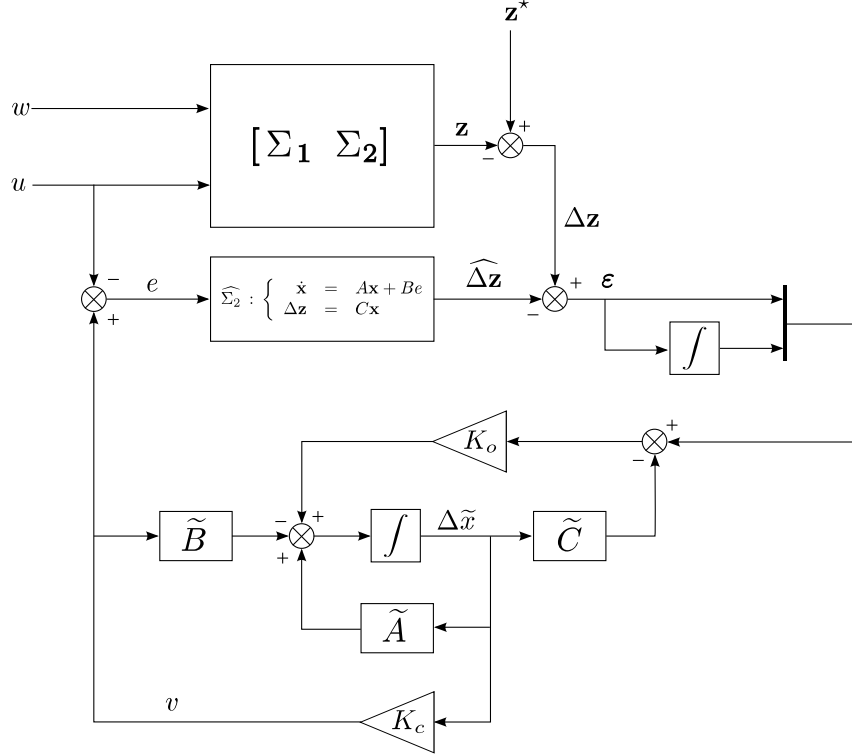


FIG. 3.4: Schéma de synthèse LQG avec modèle d'erreur et contrôle intégral

Tout d'abord, introduisons les notations suivantes :

$$\Sigma = (\Sigma_1 \quad \Sigma_2) : \begin{cases} \dot{\mathbf{x}}_n = A\mathbf{x}_n + Bu \\ \mathbf{z} = C\mathbf{x}_n + \nu \\ \Delta\mathbf{z} = \mathbf{z}^* - \mathbf{z} \\ \nu = \Sigma_1 w \end{cases} \quad (3.86)$$

$$\widehat{\Sigma}_2 : \begin{cases} \dot{\mathbf{x}} = A\mathbf{x} + Be \\ \widehat{\Delta\mathbf{z}} = C\mathbf{x} \end{cases} \quad (3.87)$$

Posons  $\mathbf{x}_\varepsilon = -\mathbf{x}_n$ . Nous pouvons donc construire un système  $\Sigma_\varepsilon$  tel que :

$$\Sigma_\varepsilon : \begin{cases} \dot{\mathbf{x}}_\varepsilon = A\mathbf{x}_\varepsilon - Bu \\ \Delta\mathbf{z} = C\mathbf{x}_\varepsilon + \nu' \\ \nu' = \mathbf{z}^* - \nu \end{cases} \quad (3.88)$$

Or, nous avons :

$$\varepsilon = \Delta\mathbf{z} - \widehat{\Delta\mathbf{z}}$$

Avec  $e = v - u$  et  $\Delta \mathbf{x} = \mathbf{x}_\varepsilon - \mathbf{x}$ , nous pouvons définir le modèle d'erreur suivant :

$$\Delta \dot{\mathbf{x}} = A\Delta \mathbf{x} - Bv \quad (3.89)$$

$$\varepsilon = C\Delta \mathbf{x} + \nu' \quad (3.90)$$

Nous pouvons également y inclure un contrôle intégral en ajoutant les états  $\Delta \dot{\xi} = C\Delta \mathbf{x}$ , ce qui donne le modèle augmenté suivant :

$$\Delta \dot{\tilde{\mathbf{x}}} = \tilde{A}\Delta \tilde{\mathbf{x}} - \tilde{B} \begin{pmatrix} v \\ 0 \end{pmatrix} \quad (3.91)$$

$$\tilde{\varepsilon} = \tilde{C}\Delta \tilde{\mathbf{x}} + \tilde{\nu} \quad (3.92)$$

$$(3.93)$$

avec

$$\tilde{\varepsilon} = \begin{pmatrix} \varepsilon \\ \int \varepsilon \end{pmatrix} \quad \Delta \tilde{\mathbf{x}} = \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \xi \end{pmatrix} \quad (3.94)$$

$$\tilde{A} = \begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix} \quad \tilde{B} = \begin{pmatrix} B \\ 0 \end{pmatrix} \quad (3.95)$$

$$\tilde{C} = \begin{pmatrix} C & 0 \\ 0 & I \end{pmatrix} \quad (3.96)$$

Cette représentation n'est pas minimale dans le cas SIMO, ce qui peut entraver la synthèse. Une fois que nous avons calculé une réalisation minimale, nous procédons à la mise sous forme compagne verticale afin de réduire le nombre de paramètre à interpoler par la suite. Rappelons que la forme compagne verticale d'un système donné est une réalisation dans laquelle la matrice d'état  $A$  apparaît sous la forme suivante :

$$A_c = TAT^{-1} = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & \cdots & 0 & -a_{n-1} \\ 0 & 1 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & -a_2 \\ 0 & \cdots & \cdots & 0 & 1 & -a_1 \end{pmatrix}$$

où les  $a_k$  sont les coefficients du polynôme caractéristique de  $A$  et  $T$  la matrice de changement de base. Pour des raisons de clarté, nous ne changerons pas les notations des matrices ainsi remaniées.

Nous allons à présent passer à la synthèse de l'observateur en proposant le filtre de Kalman suivant :

$$\Delta \dot{\tilde{\mathbf{x}}} = \tilde{A}\Delta \tilde{\mathbf{x}} - \tilde{B} \begin{pmatrix} v \\ 0 \end{pmatrix} + K_o(\tilde{\varepsilon} - \tilde{C}\Delta \tilde{\mathbf{x}}) \quad (3.97)$$

$$K_o = P_o\tilde{C}R_o^{-1} \quad (3.98)$$

$$0 = \tilde{A}P_o + P_o\tilde{A}^\top + Q_o - P_o\tilde{C}^\top R_o^{-1}\tilde{C}P_o \quad (3.99)$$

$$R_o = I \quad (3.100)$$

$$Q_o^{-1} = T_o\mathcal{G}_o(T_o) \quad (3.101)$$

$$\mathcal{G}_o(T_o) \triangleq \int_0^{T_o} e^{\tilde{A}^\top t} \tilde{C}^\top \tilde{C} e^{\tilde{A} t} dt \quad (3.102)$$

où  $\mathcal{G}_o(T_o)$  est le grammien d'observabilité partiel sur l'*horizon d'observation*  $T_o$ . Cette notion a été introduite dans [de Larminat, 1996], accompagnée de celle d'*horizon de commande*  $T_c$ , que nous verrons par la suite. Elle permet de simplifier la sélection des pondérations associées aux équations de Riccati utilisées en synthèse LQG.

Enfin, nous allons calculer le retour d'état qui minimise le critère entrée/sortie suivant :

$$\mathcal{J}_{LQ} = \int_0^{\infty} \left( \tilde{\varepsilon}(t)^\top S_c \tilde{\varepsilon}(t) + r_c e(t)^2 \right) dt \quad (3.103)$$

Nous obtenons :

$$v = K_c \Delta \tilde{\mathbf{x}} \quad (3.104)$$

$$K_c = \frac{1}{r_c} \tilde{B}^\top P_c \quad (3.105)$$

$$0 = \tilde{A}^\top P_c + P_c \tilde{A} + \tilde{C}^\top S_c \tilde{C} - \frac{1}{r_c} P_c \tilde{B} \tilde{B}^\top P_c \quad (3.106)$$

$$S_c = I \quad (3.107)$$

$$r_c = T_c \mathcal{H}_c(T_c) \quad (3.108)$$

$$\mathcal{H}_c(T_c) \triangleq \int_0^{T_c} \mathbf{h}(t)^\top \mathbf{h}(t) dt \quad (3.109)$$

où  $\mathbf{h}(t)$  est le vecteur des réponses impulsionnelles du système augmenté,  $\mathcal{H}_c(T_c)$  un coefficient de transfert d'énergie sur un *horizon de commande*  $T_c$ , mentionné précédemment.

### 3.2.2.3 Conclusion

Nous venons de voir une méthode qui va nous permettre d'inverser en temps réel le système à contrôler. De plus, le fait que nous utilisons pour identifier  $\widehat{\Sigma}_2$  les réponses impulsionnelles enregistrées lors de la génération hors ligne, nous assure que l'approximation  $\widehat{\Sigma}_2 \approx \Sigma_2$  est vérifiée. Le signal ainsi généré va nous servir de référence pour retoucher la loi feed-forward, comme il a été suggéré à la figure 3.5. Il reste à présent à déterminer en ligne les paramètres de la lois.

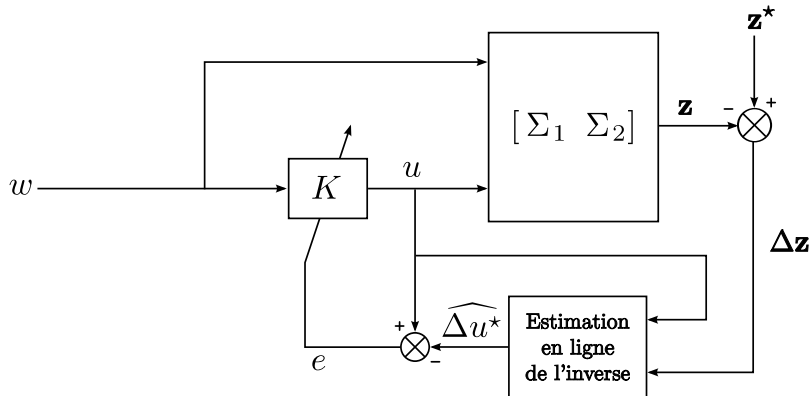


FIG. 3.5: Principe de la génération en ligne

### 3.3 Identification

A ce stade, nous devons répondre à la deuxième partie du problème 3.1.1 (p. 38), à savoir déterminer une loi capable de reproduire la commande  $\Delta u^*$  sur l'ensemble du domaine de fonctionnement du système à contrôler. Nous donnerons deux possibilités pour déterminer la structure de la loi de commande. Cette étape sera uniquement abordée dans la partie hors ligne dans la mesure où nous ne souhaitons pas modifier la structure de la loi dans un contexte temps-réel. Toutefois, une fois la structure déterminée, nous pourrions optimiser les paramètres à la fois dans un contexte hors ligne et en ligne.

#### 3.3.1 Identification hors ligne des paramètres de la loi

Pour cela, nous pouvons précéder de deux façons : l'une empirique basée sur un raisonnement qualitatif sur l'évolution de  $\Delta u^*$  sur l'ensemble du domaine de fonctionnement ; l'autre plus systématique faisant appel à des méthodes classiques.

##### 3.3.1.1 Approche empirique

La synthèse se fait ici par observations des réponses obtenues par inversion. A la condition que les évolutions paramétriques du système ne soient pas trop brusques, les signaux  $\Delta u^*$  ont de grandes chances de présenter des similitudes comportementales, se manifestant plus ou moins selon le point de fonctionnement.

A partir d'un raisonnement qualitatif, nous pouvons donc deviner un correcteur  $K(s, \theta)$  composé d'un ensemble de briques de bases simples (filtres de 1<sup>er</sup> et 2<sup>ème</sup> ordre par exemple) et résoudre le problème d'identification suivant pour chaque point de vol :

**PROBLÈME 3.3.1 :** *Soit  $K(s, \theta)$  un correcteur de structure donnée, paramétré par le vecteur  $\theta$ . Déterminer  $\theta^*$  tel que*

$$\theta^* = \arg \min_{\theta} \|K(s, \theta) * w(t) - \Delta u^*(t)\|^2 \quad (3.110)$$

L'avantage de cette méthode, hormis sa simplicité de mise en place, est qu'elle permet d'introduire une signification physique à la structure proposée. Du point de vue d'un ingénieur, il peut être préférable d'avoir une justification concrète d'une structure de loi.

Toutefois, cela reste une approche essais/erreurs, limitée à des structures de correction simples. De plus, étant donnée la façon dont le problème (3.110) est posé, l'interpolation *a posteriori* des paramètres peut parfois poser des difficultés étant donné qu'on n'impose pas de cohérence entre les différents points de vol. Nous pouvons alors envisager deux solutions.

Premièrement, nous pouvons calculer un correcteur médian  $K(s, \theta_o)$  et introduisons dans le critère d'identification, un terme de pénalité sur la distance entre le vecteur paramètre du point courant  $\theta_k$  et  $\theta_o$ . Le calcul est initialisé avec  $K(s, \theta_o)$ .

Deuxièmement, dans l'hypothèse où les  $\theta_k$  sont répartis régulièrement dans l'espace paramétrique, nous pouvons effectuer un calcul de « proche en proche ». Nous donnons un exemple en dimension 2 ( $\mathbf{a} = (a_1 \ a_2)^T$ ) à la figure 3.6, afin de clarifier le propos.

Soit  $\theta_{k,l}$  la valeur de  $\theta$  au point du maillage  $(a_{1,k}, a_{2,l})$ . Notons  $N_1$  (respectivement  $N_2$ ) le nombre de points de maillage selon la dimension  $a_1$  (respectivement  $a_2$ ). Nous donnons l'algorithme suivant :

- Pour  $l = 1$  à  $N_2$
- Pour  $k = 1$  à  $N_1$

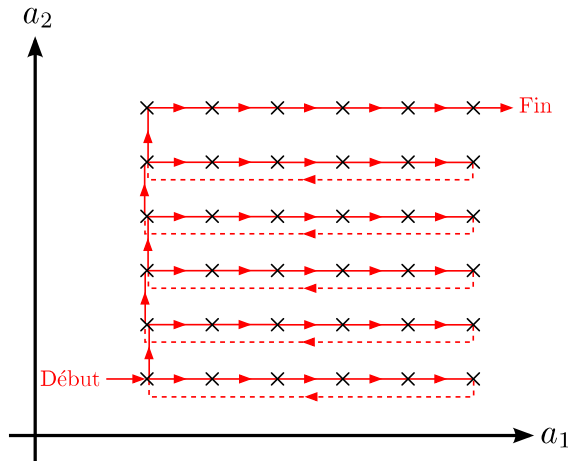


FIG. 3.6: Exemple de synthèse de proche en proche en dimension 2

- Si  $k > 1$ 
  - $\theta_{\text{init}} = \theta_{k-1,l}$
- Ou si  $l > 1$ 
  - $\theta_{\text{init}} = \theta_{k,l-1}$
- Sinon
  - $\theta_{\text{init}} = \theta_o$
- Fin
- Identification de  $\theta_{k,l}$  à partir de  $\theta_{\text{init}}$
- Fin
- Fin

### 3.3.1.2 Approche systématique

Il s'agit ici de mettre à profit les nombreuses méthodes d'identification existant dans la littérature. Les principaux avantages sont les garanties théoriques de ces méthodes ainsi que la possibilité d'élaborer des correcteurs d'ordre plus importants que précédemment. Cependant, il se peut que les conditions nécessaires au bon déroulement de ces méthodes ne soient pas réunies. Enfin, la perte du raisonnement qualitatif peut rebuter l'utilisateur qui ne souhaite pas se satisfaire d'une solution « toute faite ».

### 3.3.1.3 Conclusion

Le propos de cette partie est resté volontairement ouvert car le choix de la méthode dépend fortement de l'application traitée. En effet, selon les caractéristiques des signaux utilisés pour l'identification, une méthode sera plus adaptée qu'une autre. Dans un contexte hors ligne, l'approche empirique semble la plus à même de répondre rapidement au besoin. Nous verrons un exemple d'application au chapitre 8. Cependant, si les signaux utilisés sont trop complexes pour pouvoir élaborer une structure, l'approche systématique semble recommandée. Enfin, l'utilisation de bases de filtres particuliers pourrait améliorer la cohérence des paramètres par construction, c'est-à-dire l'expression du filtre  $K(s, \theta)$  dans ce type de base [Akçay et Ninness, 1998b], [Akçay et Ninness, 1998a].

### 3.3.2 Identification en ligne des paramètres de la loi

Au paragraphe §3.3.1, nous avons suggéré que l'approche pratique pour déterminer une structure était une approche empirique, basée sur un raisonnement qualitatif. Toutefois, cela peut paraître moins pratique dans un cadre d'adaptation en temps-réel d'une boucle ouverte. En effet, la stabilité devient complexe à intégrer à la mise à jour des coefficients lorsque l'ordre du correcteur est supérieur à 3. C'est là que les méthodes théoriques d'identification vont intervenir, plus particulièrement la méthode de Steiglitz-McBride appliquée à une structure de filtre particulière : le filtre en treillis normalisé.

#### 3.3.2.1 Description d'un filtre en treillis normalisé

Soit un filtre ARMA d'ordre  $N$  attaqué en entrée par un signal  $w(k)$  et générant la sortie  $\hat{u}(k)$ . La relation entrée/sortie s'écrit [Fleury, 2001] :

$$\sum_{m=0}^N a_m \hat{u}(k-m) = \sum_{p=1}^P b_p w(k-p), \quad \text{avec } a_0 = 1 \text{ et } N > P$$

La partie gauche de la relation est la partie AR et celle de droite la partie MA du filtre. Un tel filtre peut s'exprimer sous forme d'un filtre en treillis normalisé dont la structure est représentée figure 3.7.

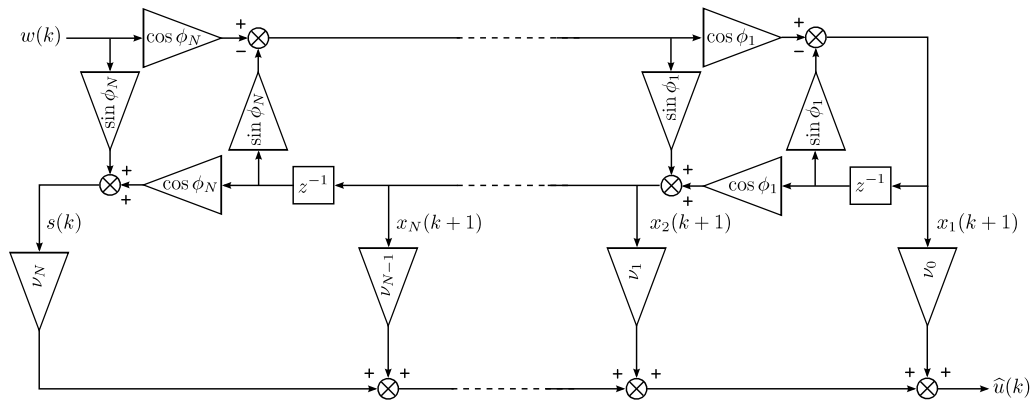


FIG. 3.7: Filtre en treillis normalisé

Soit la représentation d'état discrète d'un tel filtre d'ordre  $N$  :

$$\hat{u}(k) = \mathbf{h}(k)^\top \begin{pmatrix} \mathbf{x}(k+1) \\ s(k) \end{pmatrix} \quad (3.111)$$

$$\begin{pmatrix} \mathbf{x}(k+1) \\ s(k) \end{pmatrix} = Q(k) \begin{pmatrix} \mathbf{x}(k) \\ w(k) \end{pmatrix} \quad (3.112)$$

$$Q(k) = Q_1(k) \times \dots \times Q_N(k), \quad (3.113)$$

avec

$$Q_n(k) = \begin{pmatrix} I_{n-1} & & 0 & 0 \\ 0 & -\sin \phi_n(k) & \cos \phi_n(k) & 0 \\ 0 & \cos \phi_n(k) & \sin \phi_n(k) & 0 \\ 0 & & 0 & I_{N-n} \end{pmatrix}, \quad \forall n = 1, \dots, N$$

$$\mathbf{h}(k)^\top = (\nu_0(k) \quad \dots \quad \nu_N(k)).$$

$Q(k)$  étant le produit de  $N$  matrices de rotations de Givens, elle est orthogonale [Golub et Van Loan, 1996].

La relation entre les termes  $(a_n, b_l)$  et  $(\phi_n, \nu_n)$  n'est pas triviale mais nous pouvons les rapprocher en assimilant les  $\phi_n$  à la partie AR du filtre en treillis et les  $\nu_n$  à la partie MA.

Nous donnons par la suite les grandes lignes visant à montrer les conditions suffisantes de stabilité asymptotique exponentielle d'un tel filtre. Pour davantage de détails, le lecteur intéressé pourra se référer à [Regalia, 1992].

Nous commençons par étudier la stabilité du système interne et posons pour le moment  $w(k) \equiv 0$ , soit :

$$\begin{pmatrix} \mathbf{x}(k+1) \\ s(k) \end{pmatrix} = Q(k) \begin{pmatrix} \mathbf{x}(k) \\ 0 \end{pmatrix}.$$

La matrice  $Q(k)$  étant orthogonale, nous avons l'égalité suivante :

$$\|\mathbf{x}(k+1)\|^2 - \|\mathbf{x}(k)\|^2 = -s(k)^2, \quad \forall k \quad (3.114)$$

Nous pouvons voir que l'énergie du système interne décroît à chaque instant, selon l'énergie délivrée par  $s(k)$ . Nous allons à présent raisonner sur une fenêtre temporelle de longueur  $\tau$  et décomposer  $Q(k)$  sous la forme suivante :

$$Q(k) = \begin{pmatrix} Q_{11}(k) & \mathbf{q}_{12}(k) \\ \mathbf{q}_{21}(k)^\top & q_{22}(k) \end{pmatrix}$$

où  $Q_{11}(k)$  est de taille  $N \times N$ ,  $\mathbf{q}_{12}(k)$  et  $\mathbf{q}_{21}(k)$  sont de taille  $N \times 1$  et  $q_{22}(k)$  est un scalaire. Nous définissons les matrices suivantes :

$$Q_t(k+\tau, k) \triangleq Q_{11}(k+\tau-1) \times \dots \times Q_{11}(k) \quad (3.115)$$

$$Q_o(k+\tau, k) \triangleq \begin{pmatrix} \mathbf{q}_{21}(k)^\top \\ \mathbf{q}_{21}(k+1)^\top Q_{11}(k) \\ \vdots \\ \mathbf{q}_{21}(k+\tau-1)^\top Q_{11}(k+\tau-2) \times \dots \times Q_{11}(k) \end{pmatrix} \quad (3.116)$$

$Q_t(\cdot)$  et  $Q_o(\cdot)$  sont respectivement les matrices de transition d'état et d'observabilité. Nous pouvons déduire :

$$\mathbf{x}(k+\tau) = Q_t(k+\tau, k)\mathbf{x}(k) \quad (3.117)$$

$$\begin{pmatrix} s(k) \\ \vdots \\ s(k+\tau) \end{pmatrix} = Q_o(k+\tau, k)\mathbf{x}(k) \quad (3.118)$$

**DÉFINITION 3.3.1 :** *Un système est uniformément observable si et seulement si*

$$\exists \tau \in \mathbb{N}^* \text{ et } \exists \delta > 0 \mid Q_o(k+\tau, k)^\top Q_o(k+\tau, k) \geq \delta I, \quad \forall k \quad (3.119)$$

Nous pouvons déduire de (3.114), (3.117) et (3.118) :

$$\begin{aligned}\|\mathbf{x}(k + \tau)\|^2 &= \mathbf{x}(k)^\top \left( I - Q_o(k + \tau, k)^\top Q_o(k + \tau, k) \right) \mathbf{x}(k) \\ &= \mathbf{x}(k)^\top Q_t(k + \tau, k)^\top Q_t(k + \tau, k) \mathbf{x}(k), \quad \forall \mathbf{x}(k)\end{aligned}\quad (3.120)$$

$$\Rightarrow Q_t(k + \tau, k)^\top Q_t(k + \tau, k) = I - Q_o(k + \tau, k)^\top Q_o(k + \tau, k) \quad (3.121)$$

Il apparaît alors une formulation équivalente de la contrainte d'observabilité uniforme :

$$(3.119) \iff \|Q_t(k + \tau, k)\|^2 \leq 1 - \delta, \quad \forall k, \quad (3.122)$$

ce qui nous permet d'écrire la relation suivante :

$$\|\mathbf{x}(k + m\tau)\|^2 \leq (1 - \delta)^m \|\mathbf{x}(k)\|^2, \quad \forall m \quad (3.123)$$

Nous pouvons donc déduire la stabilité exponentielle asymptotique :

$$\lim_{k \rightarrow \infty} \exp(\eta\tau) \|\mathbf{x}(k)\|^2 = 0, \quad \text{avec } 0 \leq \eta < -\frac{\ln(1 - \delta)}{\tau}, \quad (3.124)$$

et ce, quel que soit l'état initial.

En résumé, l'orthogonalité de  $Q(k)$  nous permet de dire que la stabilité asymptotique exponentielle du système interne est assurée à condition que le vecteur d'état soit uniformément observable par  $s(k)$ . Pour la suite, nous énonçons la propriété suivante :

**PROPRIÉTÉ :** *Le calcul direct de  $Q_o(k + N, k)$  pour laquelle les lignes ont été permutées, met en évidence une matrice triangulaire dont les termes diagonaux sont :*

$$q_n^o = \prod_{i=1}^n \cos \phi_{N+1-i}(k + n - i), \quad \forall n = 1, \dots, N \quad (3.125)$$

La vérification de cette assertion est donnée en annexe D afin de ne pas encombrer davantage le propos qui est déjà suffisamment chargé.

D'après (3.119), il est alors suffisant que la plus petite valeur singulière de  $Q_o(n)$  soit minorée<sup>2</sup>. Or, cette plus petite valeur singulière est nulle si et seulement si une des valeurs propres  $q_n^o$  est nulle. Dès lors, si nous supposons l'existence d'un réel positif  $\epsilon$  tel que :

$$|\cos \phi_n(k)| \geq \epsilon, \quad \forall n, k \quad (3.126)$$

alors

$$|q_n^o| \geq \epsilon^N, \quad \forall n, k \quad (3.127)$$

Etant donné que les valeurs singulières et propres de  $Q_o(k + N, k)$  dont les lignes ont été permutées, sont des fonctions continues des  $\phi_n$ , nous pouvons déduire de l'inégalité précédente qu'il existe une constante positive  $\delta$  telle que :

$$\underline{\sigma}(Q_o(k + N, k)) \geq \sqrt{\delta}, \quad \forall n \quad (3.128)$$

La condition (3.126) assure donc l'observabilité uniforme du système interne, et par extension sa stabilité asymptotique exponentielle.

L'étude de la stabilité du système forcé utilise un résultat de la théorie des systèmes à temps variant [Silverman et Anderson, 1968] :

<sup>2</sup>La permutation des lignes n'affecte pas les valeurs singulières



PROPOSITION 3.3.1 : Soit un système à temps variant. Si son évolution au cours du temps est bornée et si sa partie interne est exponentiellement stable, alors le système global est EBSB (entrée bornée – sortie bornée) et EBEB (entrée bornée – états bornés). De plus, sa réponse impulsionnelle décroît exponentiellement.

Etant donné que  $Q(k)$  est orthogonale, l'évolution au cours du temps du filtre est bien bornée si le système est exponentiellement stable. Donc si les  $\phi_n(k)$  satisfont (3.126) et si les  $\nu_n(k)$  restent bornés, les conditions de stabilité asymptotique exponentielle et de bornitude sont vérifiées et le filtre en treillis est alors exponentiellement stable EBSB.

### 3.3.2.2 Méthode de Steiglitz-McBride généralisée

Le figure 3.8 représente une structure équivalente à celle de Steiglitz-McBride pour l'identification, en utilisant le modèle d'état donné par (3.111) et (3.112).

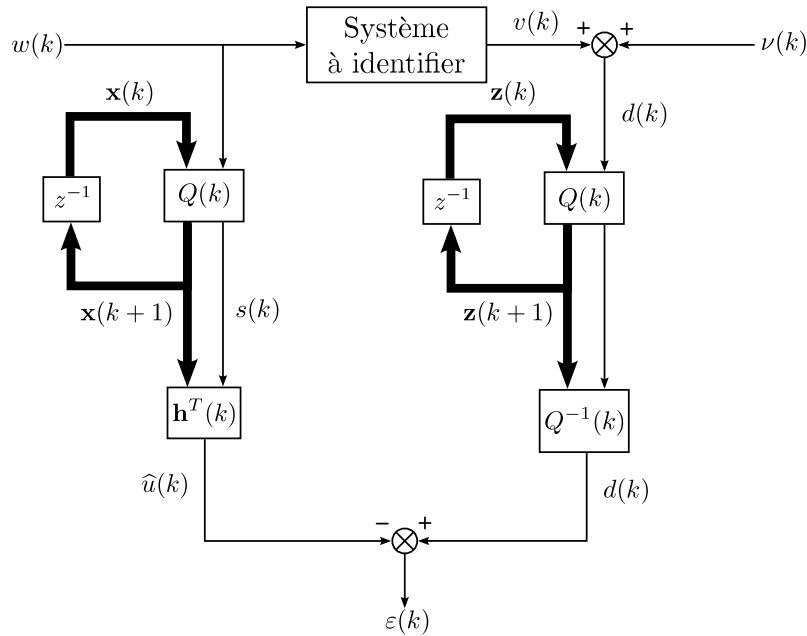


FIG. 3.8: Calcul de l'erreur a priori

La matrice  $Q^{-1}(k)$  n'est pas utilisée entièrement étant donné que l'on n'a besoin que d'une sortie de la matrice, en l'occurrence celle correspondant à la dernière ligne de  $Q^{-1}(k)$ . Notons également que pour un filtre en treillis,  $Q^{-1}(k) = Q(k)^T$  étant donné que  $Q(k)$  est orthogonale, ce qui simplifie encore davantage le calcul.

L'erreur *a priori* s'écrit :

$$\varepsilon(k) = \mathbf{q}_{N+1}(k)^T \begin{pmatrix} \mathbf{z}(k+1) \\ s(k) \end{pmatrix} - \mathbf{h}(k)^T \begin{pmatrix} \mathbf{x}(k+1) \\ w(k) \end{pmatrix} \quad (3.129)$$

La détermination des gradients de l'erreur par rapport aux  $\nu_n$ , notés  $\nabla_{\nu_n} \varepsilon(k)$ , est triviale et donne :

$$\nabla_{\nu_n} \varepsilon(k) = \begin{cases} x_{n+1}(k), & \text{si } n = 0, \dots, N-1 \\ w(k), & \text{si } n = N \end{cases} \quad (3.130)$$

Les gradients par rapport aux  $\phi_n$  sont plus difficiles à calculer. Nous déduisons de (3.129) une expression des  $\nabla_{\phi_n} \varepsilon(k)$  :

$$\nabla_{\phi_n} \varepsilon(k) = \nabla_{\phi_n} \mathbf{q}_{N+1}(k)^\top \begin{pmatrix} \mathbf{z}(k+1) \\ s(k) \end{pmatrix} \quad (3.131)$$

Les termes  $\nabla_{\phi_n} \mathbf{q}_{N+1}(k)^\top$  peuvent être calculés à l'aide du lemme suivant :

LEMME 3.3.1 : *Soit  $Q(k)$  une matrice orthogonale construite par  $N$  rotation de Givens, tel qu'en (3.113). Sa transposée peut alors être écrite à l'aide de vecteurs lignes sous la forme :*

$$Q(k)^\top = \begin{pmatrix} \nabla_{\phi_1} \mathbf{q}_{N+1}(k)^\top / \gamma_1(k) \\ \vdots \\ \nabla_{\phi_N} \mathbf{q}_{N+1}(k)^\top / \gamma_N(k) \\ \mathbf{q}_{N+1}(k)^\top \end{pmatrix} \quad (3.132)$$

avec

$$\gamma_n(k) = \begin{cases} 1, & \text{si } n = N \\ \gamma_{n+1}(k) \cos \phi_{n+1}(k), & \text{si } n = N-1, \dots, 1 \end{cases} \quad (3.133)$$

Nous déduisons de ce lemme et de l'équation (3.129) :

$$\begin{pmatrix} \nabla_{\phi_1} \varepsilon(k) / \gamma_1(k) \\ \vdots \\ \nabla_{\phi_N} \varepsilon(k) / \gamma_N(k) \\ d(k) \end{pmatrix} = Q(k)^\top \begin{pmatrix} \mathbf{z}(k+1) \\ s(k) \end{pmatrix} \quad (3.134)$$

Or, nous pouvons voir à la figure 3.8 que :

$$\begin{pmatrix} \mathbf{z}(k+1) \\ s(k) \end{pmatrix} = Q(k) \begin{pmatrix} \mathbf{z}(k) \\ d(k) \end{pmatrix} \quad (3.135)$$

d'où, en utilisant l'orthogonalité de  $Q(k)$

$$\begin{pmatrix} \mathbf{z}(k) \\ d(k) \end{pmatrix} = Q(k)^\top \begin{pmatrix} \mathbf{z}(k+1) \\ s(k) \end{pmatrix} \quad (3.136)$$

de telle façon que le vecteur d'état  $\mathbf{z}(k)$  donne, à un ensemble de coefficients près, les gradients souhaités :

$$\nabla_{\phi_n} \varepsilon(k) = \begin{cases} -z_n(k) \prod_{j=n+1}^N \cos \phi_j(k), & n = 1, \dots, N-1 \\ -z_N(k), & n = N \end{cases} \quad (3.137)$$

A la section §3.3.2, un schéma légèrement différent est utilisé. En effet, au lieu de préfiltrer le signal de référence bruité  $d(k)$  pour calculer les gradients, on utilise directement l'estimée  $\hat{y}(k)$ . L'avantage est de retirer le bruit de mesure du calcul des gradients. Cette méthode est une modification de la méthode erreur de sortie (MOEM) et il est montré dans [Regalia, 1992] que l'ensemble des points stationnaires utilisant ces gradients modifiés sont cohérents avec la méthode d'erreur de sortie « classique » (OEM).

### 3.3.2.3 Algorithme simplifié de gradient partiel

Un algorithme permettant l'adaptation des  $\{\nu_n(k)\}$  et des  $\{\phi_n(k)\}$  afin de minimiser l'erreur a priori entre la référence  $v(k)$  et  $\hat{u}(k)$  au sens des moindres carrés, est proposé dans [Nikolakopoulos *et al.*, 2007] à partir d'une étude théorique sur le généralisation de la technique d'identification de Steiglitz-McBride [Regalia, 1992]. L'algorithme est donné par la suite sous forme de pseudo-code et schématisé à la figure 3.9 :

#### Données à l'instant $k$

- Coefficients du filtre

$$\begin{aligned} \nu_n(k), \quad n = 0, \dots, N \\ \phi_n(k), \quad n = 1, \dots, N \end{aligned}$$

- Etats du filtre

$$\begin{aligned} x_n(k), \quad n = 1, \dots, N \\ x_{N+1}(k) = w(k) \end{aligned}$$

- Etats du post-filtre

$$\xi_n(k), \quad n = 1, \dots, N$$

- Nouvelles données

$$\begin{aligned} w(k), \quad \text{entrée} \\ v(k), \quad \text{référence} \end{aligned}$$

#### Filtrage

- soit  $g_{N+1}(k) = w(k)$ . Récursivement pour  $n = N + 1, N, \dots, 2$  :

$$\begin{pmatrix} g_{n-1}(k) \\ x_n(k) \end{pmatrix} = \begin{pmatrix} \cos \phi_n(k) & -\sin \phi_n(k) \\ \sin \phi_n(k) & \cos \phi_n(k) \end{pmatrix} \begin{pmatrix} g_n(k) \\ x_{n-1}(k-1) \end{pmatrix}$$

Pour  $n = 1$ ,  $x_1(k) = g_1(k)$ .

- Sortie du filtre

$$\hat{u}(k) = \sum_{n=1}^{N+1} \nu_{n-1}(k) x_n(k)$$

- Erreur *a priori*

$$\varepsilon(k) = v(k) - \hat{u}(k)$$

#### Mise à jour des coefficients

- Gradients par rapport à  $\{\nu_n(k)\}_{n=0, \dots, N}$

$$\nabla_{\nu_n} \varepsilon(k) = x_n(k)$$

- Gradients par rapport à  $\{\phi_n(k)\}_{n=1, \dots, N}$  obtenus par l'utilisation de régresseurs filtrés : soit  $\gamma_N = 1$ . Récursivement pour  $n = N, N - 1, \dots, 1$  :

$$\begin{aligned} \nabla_{\phi_n} \varepsilon(k) &= \gamma_n \xi_{n-1}(k) \\ \gamma_{n-1} &= \gamma_n \cos \phi_n(k) \end{aligned}$$

– Mise à jour par LMS<sup>3</sup>

$$\nu_n(k+1) = \nu_n(k) + \eta\varepsilon(k)\nabla_{\nu_n}\varepsilon(k), \quad n = 0, \dots, N \quad (3.138)$$

$$\phi_n(k+1) = \phi_n(k) + \eta\varepsilon(k)\nabla_{\phi_n}\varepsilon(k), \quad n = 1, \dots, N \quad (3.139)$$

#### Condition de stabilité

–  $\forall n = 1, \dots, N$

$$|\phi_n(k+1)| > \pi/2 \quad \Rightarrow \quad \phi_n(k+1) = \phi_n(k)$$

#### Post-filtrage

– Soit  $z_{N+1}(k) = -\hat{u}(k)$ . Récursivement for  $n = N+1, N, \dots, 2$  :

$$\begin{pmatrix} z_{m-1}(k) \\ \xi_n(k+1) \end{pmatrix} = \Phi_n(k) \begin{pmatrix} z_n(k) \\ \xi_n(k) \end{pmatrix}$$

Pour  $n = 1$ ,  $\xi_1(k+1) = z_0(k)$

#### 3.3.2.4 Variante de l'algorithme

Afin d'alléger la charge de calcul, l'algorithme précédent est modifié de façon à adapter directement  $\sin \phi_n(k)$  et  $\cos \phi_n(k)$ , ce qui permet d'éviter le calcul de fonctions trigonométriques [Regalia, 1994].

L'approche consiste ici à déduire comment  $Q(k)$  évolue en fonction des  $\phi_n(k)$ . Pour cela, nous donnons la propriété suivante :

**PROPOSITION 3.3.2 :** *Soit  $\mathbf{q}$  un vecteur unitaire quelconque. Il existe une matrice de Hessenberg  $Q$  dont la dernière colonne est  $\mathbf{q}$ . De plus, les angles de rotations  $\phi_n$  peuvent être choisis tels que*

$$|\phi_n| < \frac{\pi}{2}, \quad \forall n$$

si et seulement si *le premier élément de  $\mathbf{q}$  est positif.*

[Regalia, 1994] propose la vérification suivante.

*Vérification.* Si  $\mathbf{q}$  est la dernière colonne de  $Q$ , matrice orthogonale, nous pouvons écrire :

$$\begin{aligned} Q^\top Q = I &\Rightarrow Q^\top \mathbf{q} = (0 \quad \dots \quad 0 \quad 1)^\top \\ &\Rightarrow Q_N^\top \times \dots \times Q_1^\top \mathbf{q} = (0 \quad \dots \quad 0 \quad 1)^\top \end{aligned}$$

Notons  $\mathbf{q} = (q_0 \quad \dots \quad q_N)^\top$ . Nous allons à présent déterminer les coefficients  $\phi_n$  d'une matrice de Hessenberg satisfaisant la propriété. Posons  $\phi_1$  tel que :

$$\cos \phi_1 = \frac{q_0}{\sqrt{q_0^2 + q_1^2}} \quad \text{et} \quad \sin \phi_1 = \frac{q_1}{\sqrt{q_0^2 + q_1^2}}$$

et appliquons la matrice  $Q_1^\top$  :

$$Q_1^\top \mathbf{q} = \begin{pmatrix} -\sin \phi_1 & \cos \phi_1 & & \\ \cos \phi_1 & \sin \phi_1 & & \\ & & & \\ & & & I_{N-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \mathbf{q}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{q_0^2 + q_1^2} \\ \mathbf{q}_2 \end{pmatrix}$$

<sup>3</sup>Nous pouvons également utiliser la version normalisée des LMS.

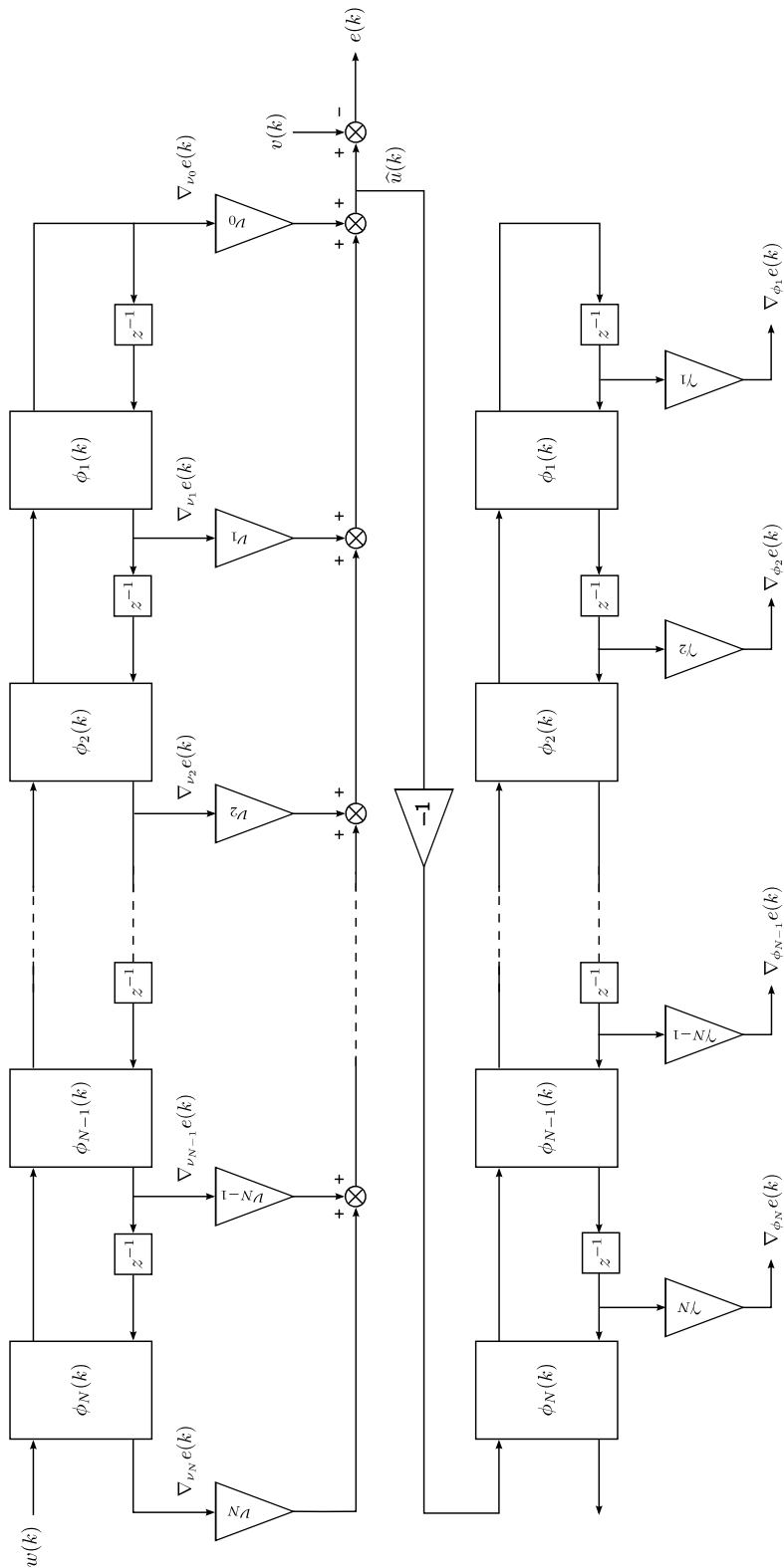


FIG. 3.9: Schéma de l'algorithme SPGL

A présent, posons  $\phi_2$  tel que :

$$\cos \phi_2 = \frac{\sqrt{q_0^2 + q_1^2}}{\sqrt{q_0^2 + q_1^2 + q_2^2}} \text{ et } \sin \phi_2 = \frac{q_2}{\sqrt{q_0^2 + q_1^2 + q_2^2}}$$

et appliquons  $Q_2^\top$  à  $Q_1^\top \mathbf{q}$  :

$$Q_2^\top Q_1^\top \mathbf{q} = \begin{pmatrix} 1 & & & \\ & -\sin \phi_2 & \cos \phi_2 & \\ & \cos \phi_2 & \sin \phi_2 & \\ & & & I_{N-2} \end{pmatrix} \begin{pmatrix} 0 \\ \sqrt{q_0^2 + q_1^2} \\ q_2 \\ \mathbf{q}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sqrt{q_0^2 + q_1^2 + q_2^2} \\ \mathbf{q}_3 \end{pmatrix}$$

Nous pouvons donc conclure que par récurrence, les  $\phi_n$  peuvent être choisis de façon à supprimer les autres entrées non nulles et arriver à  $Q^\top \mathbf{q} = (0 \ \dots \ 0 \ 1)^\top$ .  $\square$

De plus, si nous choisissons  $q_0 > 0$ , nous pouvons vérifier par récurrence que les  $\phi_n$  sont uniques, ainsi que l'assertion suivante [Regalia, 1994] :

$$\phi_n < \frac{\pi}{2}, \quad \forall n = 1, \dots, N$$

A présent, rappelons l'équation de mise à jour des  $\phi_n(k)$  donnée en (3.139) :

$$\phi_n(k+1) = \phi_n(k) + \eta \varepsilon(k) \nabla_{\phi_n} \varepsilon(k), \quad \forall n = 1, \dots, N$$

Posons  $\boldsymbol{\phi}(k) = (\phi_1(k) \ \dots \ \phi_N(k))^\top$  et  $\mathbf{q}(\boldsymbol{\phi}(k))$  la dernière colonne de  $Q(k)$ . Si  $\mu$  est suffisamment petit, la variation

$$\Delta \boldsymbol{\phi}(k) \triangleq \boldsymbol{\phi}(k+1) - \boldsymbol{\phi}(k)$$

sera suffisamment faible pour permettre le développement limité au premier ordre suivant :

$$\mathbf{q}(\boldsymbol{\phi}(k+1)) = \left( \mathbf{q}(\boldsymbol{\phi}(k)) + \sum_{n=1}^N \nabla_{\phi_n} \varepsilon(k) \Delta \phi_n(k) \right) / \zeta \quad (3.140)$$

$$\zeta = \left\| \mathbf{q}(\boldsymbol{\phi}(k)) + \sum_{n=1}^N \nabla_{\phi_n} \varepsilon(k) \Delta \phi_n(k) \right\| \quad (3.141)$$

où  $\zeta$  nous permet d'assurer la normalisation du vecteur à chaque instant. Le lemme 3.3.1 (p. 61) nous permet d'écrire :

$$\begin{aligned} & \mathbf{q}(\boldsymbol{\phi}(k)) + \sum_{n=1}^N \nabla_{\phi_n} \varepsilon(k) \Delta \phi_n(k) \\ &= (\nabla_{\phi_1} \varepsilon(k) \ \dots \ \nabla_{\phi_N} \varepsilon(k) \ \mathbf{q}(\boldsymbol{\phi}(k))) \begin{pmatrix} \Delta \phi_1(k) \gamma_1(k) \\ \vdots \\ \Delta \phi_N(k) \gamma_N(k) \\ 1 \end{pmatrix} \end{aligned} \quad (3.142)$$

$$= Q(k) \begin{pmatrix} \Delta \phi_1(k) \gamma_1(k) \\ \vdots \\ \Delta \phi_N(k) \gamma_N(k) \\ 1 \end{pmatrix} \quad (3.143)$$

Or, d'après (3.139), nous avons :

$$\Delta\phi_n(k) = \eta\varepsilon(k)\nabla_{\phi_n(k)}\varepsilon(k) \quad (3.144)$$

d'où

$$\mathbf{q}(\phi(k+1)) \approx \frac{1}{\zeta}Q(k) \begin{pmatrix} \eta\varepsilon(k)\nabla_{\phi_1}\varepsilon(k)\gamma_1(k) \\ \vdots \\ \eta\varepsilon(k)\nabla_{\phi_N}\varepsilon(k)\gamma_N(k) \\ 1 \end{pmatrix} \quad (3.145)$$

Nous venons de déterminer une approximation du dernier vecteur colonne de la matrice  $Q(k+1)$ . Nous allons donc utiliser la proposition 3.3.2 afin de calculer les cosinus et sinus des angles de rotations  $\phi_n(k+1)$ . En posant  $\mathbf{q}(\phi(k+1)) = (q_0 \dots q_N)^\top$ , la contrainte

$$Q(k+1)^\top \mathbf{q}(\phi(k+1)) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

peut se résumer sous la forme des suppressions récurrentes suivantes :

$$\begin{pmatrix} \zeta_n \\ 0 \end{pmatrix} = \begin{pmatrix} \cos \phi_n(k+1) & \sin \phi_n(k+1) \\ -\sin \phi_n(k+1) & \cos \phi_n(k+1) \end{pmatrix} \begin{pmatrix} \zeta_{n-1} \\ q_n \end{pmatrix}, \quad \forall n = 1, \dots, N \quad (3.146)$$

avec  $\zeta_0 = q_0$ . Ce calcul récursif nous permet donc de déterminer directement les cosinus et sinus des angles de rotations.

### 3.3.2.5 Conclusion

La génération en ligne des paramètres de la loi feed-forward requiert une structure particulière de correcteur, permettant de vérifier aisément la stabilité de ce dernier. Nous avons donné deux algorithmes d'identification basés sur les moindres carrés normalisés (NLMS). Il faudra donc se rappeler que le vecteur  $\boldsymbol{\theta}(k)$  ne constituera une référence qu'une fois que l'algorithme NLMS aura convergé.

## 3.4 Conclusion

Au cours de ce chapitre, nous avons mis en évidence le caractère non-trivial de la génération de données d'apprentissage dans un contexte indirect. En effet, nous avons dû faire appel à des outils variés afin d'atteindre notre objectif. Même si le travail ne présente pas une généralité à toute épreuve, il propose toutefois une démarche de réflexion vis-à-vis du problème posé et des ressources à disposition pour y répondre.

# Chapitre 4

## Outils pour l'apprentissage

### Sommaire

---

<b>4.1</b>	<b>Stratégies d'apprentissage hors ligne</b>	<b>67</b>
4.1.1	Pré-traitement des données	67
4.1.2	Placement des centres fixé <i>a priori</i>	68
4.1.3	Placement par algorithme des <i>k</i> -moyennes	72
4.1.4	Placement supervisé des centres	73
4.1.5	Conclusion	74
<b>4.2</b>	<b>Stratégies d'apprentissage en ligne</b>	<b>74</b>
4.2.1	Moindres carrés récursifs	75
4.2.2	Extension de l'algorithme SPGL	77
<b>4.3</b>	<b>Conclusion</b>	<b>79</b>

---

C E CHAPITRE aborde les aspects pratiques liés au paramétrage et à l'optimisation des réseaux RBF et mélanges gaussiens d'experts dans un contexte hors ligne et en ligne. Si l'utilisation de l'aspect en ligne peut ne pas être systématique, il peut être intéressant de considérer le réglage hors ligne comme une initialisation avant adaptation.

### 4.1 Stratégies d'apprentissage hors ligne

Nous abordons dans ce paragraphe le travail préliminaire à tout apprentissage à savoir la définition de la structure du mélange d'experts. Au cours du chapitre 2, nous avons donné un caractère générique aux divers éléments structuraux du MEX, comme le placement des centres des RBF ou le calcul de la taille de RBF. Nous proposons ici de rentrer dans les détails pratiques et de faire un inventaire des possibilités offertes à l'utilisateur en fonction de son besoin. La partie sera articulée autour des différentes stratégies de placement des centres.

#### 4.1.1 Pré-traitement des données

La première chose à faire une fois en possession des données d'apprentissage est de les séparer en deux ensembles d'apprentissage et de validation. Bien que les méthodes d'apprentissage présentées dans la suite du chapitre soient suffisamment rapides au niveau du temps de calcul, il peut être intéressant de déterminer la taille minimale de l'ensemble



d'apprentissage pour laquelle la performance est satisfaisante. Nous donnerons un exemple au chapitre 7.

La deuxième chose à faire est de les normaliser afin de gommer les éventuels problèmes de conditionnement numérique liés à des différences d'unités des données et de faciliter le calcul des différents paramètres du MEX. Nous abordons ici les différentes manières que nous avons utilisées pour normaliser les données.

Deux cas se sont présentés à nous : un premier où nous pouvons connaître a priori les bornes supérieure et inférieure d'évolution des données et un second où cette information est donnée par la base de données générée. Si le calcul effectué pour la normalisation des données est identique pour chaque cas, la différence réside dans le fait qu'il est probable dans le second cas, qu'un point de validation se situe en dehors des bornes calculées sur les données d'apprentissage. Il faut alors intégrer ce nouveau dans le calcul de normalisation.

$$\begin{aligned}\underline{\mathbf{a}} &= \begin{pmatrix} \inf_k a_{1,k} \\ \vdots \\ \inf_k a_{M,k} \end{pmatrix} \\ \bar{\mathbf{a}} &= \begin{pmatrix} \sup_k a_{1,k} \\ \vdots \\ \sup_k a_{M,k} \end{pmatrix} \\ \boldsymbol{\rho}_{\mathbf{a}} &= \frac{1}{2} (\bar{\mathbf{a}} - \underline{\mathbf{a}}) \\ \boldsymbol{\mu}_{\mathbf{a}} &= \frac{1}{2} (\bar{\mathbf{a}} + \underline{\mathbf{a}}) \\ \mathbf{a}_n &= \text{diag}(\boldsymbol{\rho}_{\mathbf{a}})^{-1} (\mathbf{a} - \boldsymbol{\mu}_{\mathbf{a}})\end{aligned}$$

#### 4.1.2 Placement des centres fixé *a priori*

La solution la plus simple est de fixer au préalable la structure du MEX. Les centres peuvent être choisis aléatoirement dans les données d'apprentissage à la condition que ces données soit distribuées dans l'espace d'entrée de manière suffisamment représentative pour le problème considéré. Dans [Poggio et Girosi, 1989], les auteurs donnent un corollaire du théorème de Michelli (theorem 2.1, p. 18) élargissant ce dernier au cas des réseaux RBF généralisés (i.e avec moins de centres que d'exemples d'apprentissage) :

**COROLLAIRE 4.1.1 :** *Soient  $g$  une fonction satisfaisant le théorème 2.1 et un ensemble de  $K$  vecteurs distincts  $\{\mathbf{a}_k \in \mathcal{A} \mid k = 1, \dots, K\}$ . Si  $G$  est la matrice de taille  $K \times (K - l)$  obtenue à partir de la matrice de taille  $K \times K$ , dont le  $ij^{\text{ème}}$  élément est  $g(\|\mathbf{a}_i - \mathbf{a}_j\|^2)$  et dont on a supprimé  $l$  colonnes au hasard, alors la matrice de taille  $(K - l) \times (K - l)$   $G^T G$  est inversible.*

Ce corollaire est toutefois conservatif dans la mesure où le choix des centres doit être fait dans l'ensemble des points d'apprentissage. En pratique, d'autres choix sont possibles comme par exemple répartir les centres uniformément dans l'espace d'entrée pourvu que ce dernier soit un fermé et préalablement normalisé. Nous perdons les garanties du corollaire 4.1.1 mais c'est un choix qui fonctionne dans la pratique [Broomhead et Lowe, 1988]. Le problème est que le nombre de neurones nécessaires croît exponentiellement avec la

dimension de l'espace d'entrée. Toutefois des considérations physiques sur la fonction à estimer, peuvent aider à réduire le nombre de neurones.

#### 4.1.2.1 Influence de la norme utilisée

Le choix de la norme se pose dès lors que la dimension de l'espace d'entrée est supérieure ou égale à 2. En effet, celle-ci va influencer sur l'interaction entre chacune des fonctions de pondération et ses voisines. Nous qualifierons par la suite cette interaction de *recouvrement* [Ronceray *et al.*, 2007a].

Nous avons considéré la norme euclidienne et la norme- $\infty$

Les figures 4.1 et 4.2 illustrent les recouvrements respectifs des RBF dans le cas d'utilisation d'une norme  $\|\cdot\|_2$  et d'une norme  $\|\cdot\|_\infty$ , dont les définitions sont données en annexe A, p. 157.

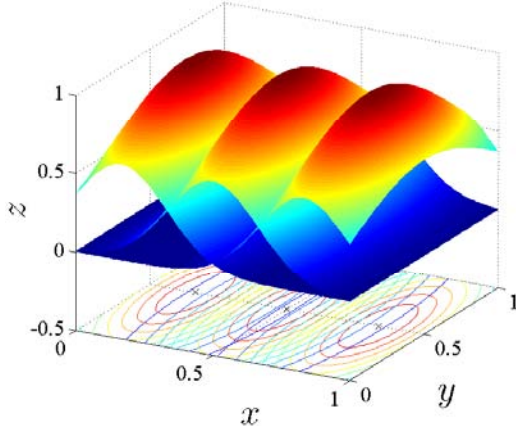


FIG. 4.1: Norme  $\|\cdot\|_2$

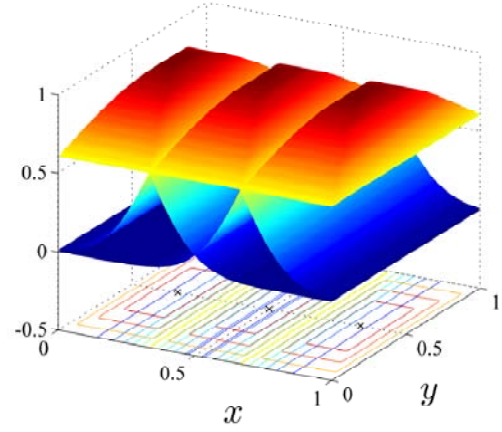


FIG. 4.2: Norme  $\|\cdot\|_\infty$

Nous pouvons voir sur ces figures que l'utilisation de la norme- $\infty$  permet une meilleure maîtrise du recouvrement des pondérations locales (intersection d'hyper-rectangles au lieu d'hyper-ellipses.)

Un autre aspect intéressant du choix de la norme est la gestion de la taille des RBF via la pondération de la norme par une matrice  $S$ . Nous donnons les notations suivantes :

$$\|\mathbf{a}\|_{2,S}^2 = \|\mathbf{S}\mathbf{a}\|_2^2 = \mathbf{a}^\top \mathbf{S}^\top \mathbf{S} \mathbf{a} \quad (4.1)$$

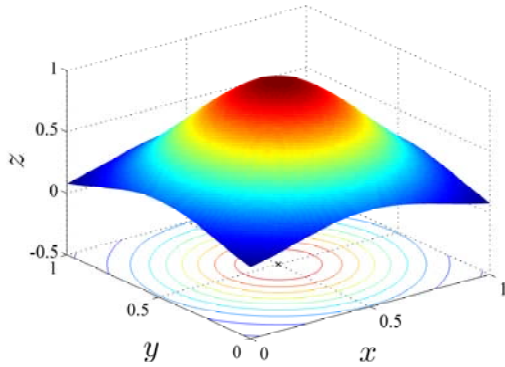
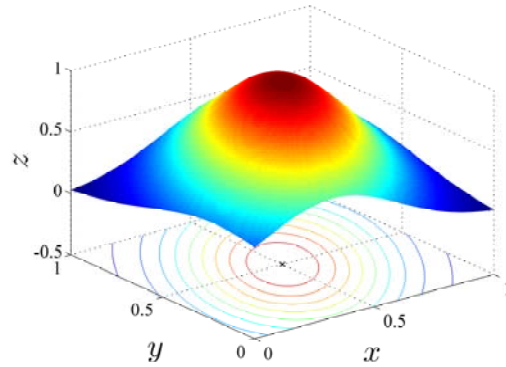
$$\|\mathbf{a}\|_{\infty,S}^2 = \|\mathbf{S}\mathbf{a}\|_\infty^2 = \max_m \left| \sum_{m'=1}^M s_{m,m'} a_{m'} \right| \quad (4.2)$$

Afin d'établir l'analogie avec les fonctions gaussiennes multivariées, nous posons :

$$\varphi_r(\mathbf{a}) = \exp\left(-\frac{1}{2}(\mathbf{a} - \mathbf{c}_r)^\top \Sigma_r^{-1}(\mathbf{a} - \mathbf{c}_r)\right) \quad (4.3)$$

$$= \exp\left(-(\mathbf{a} - \mathbf{c}_r)^\top \mathbf{S}^\top \mathbf{S}(\mathbf{a} - \mathbf{c}_r)\right) \quad (4.4)$$

$$\frac{1}{2} \Sigma_r^{-1} = \mathbf{S}^\top \mathbf{S} \quad (4.5)$$

FIG. 4.3:  $\Sigma_r = I$ FIG. 4.4:  $\Sigma_r$  : matrice de dilatation et rotation

Le choix de  $\Sigma_r^{-1}$  permet de privilégier une direction  $\mathbf{a}_o$  dans l'espace d'entrée à partir du centre  $\mathbf{c}_r$ , en effectuant une rotation et/ou une dilatation dans cette direction.

Dans le cas du maillage régulier, ces matrices vont nous permettre de mieux gérer le recouvrement des diverses pondérations, dans le cas où les dimensions de l'espace d'entrée n'ont pas la même finesse de maillage.

Nous proposons deux valeurs possibles pour  $\Sigma_r$  :

$$\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_M) \quad (4.6)$$

$$\text{Maillage quelconque : } \sigma_m = \frac{d_{max}}{\sqrt{2}R} \quad \forall m = 1, \dots, M \quad (4.7)$$

$$\text{Maillage régulier : } \sigma_m = \frac{1}{\Delta_m \ln 2} \quad \forall m = 1, \dots, M \quad (4.8)$$

où  $d_{max}$  est la distance maximum entre les centres choisis et  $\Delta_m$  est le pas de maillage sur la dimension  $m$ . Ces choix font en sorte que les RBF ne soient pas trop « pointues » ou trop « écrasées ». Il est toujours possible d'affiner ces valeurs en effectuant une série d'essais jusqu'à atteindre un résultat satisfaisant. Il faudra cependant veiller au recouvrement entre les modèles locaux voisins. En effet s'ils interagissent trop entre eux, la structure du MEX risque de perdre son caractère local.

#### 4.1.2.2 Choix des fonctions de pondération

Nous avons déjà cité au §2.1 (p. 17) quelques unes des fonctions à bases radiales utilisables. Nous en proposons deux autres à la figure 4.5, sur laquelle nous pouvons distinguer une fonction gaussienne, une multiquadrique inverse, un noyau quadratique et une fonction trapézoïdale [Atkeson *et al.*, 1997].

**Effet de la normalisation** Il est également possible (et recommandé) de normaliser les  $\varphi_r$  de façon à améliorer le comportement aux limites de l'espace d'entrée normalisé comme le montre la figure 4.6. La normalisation se fait de manière simple en divisant chaque  $\varphi_r$  par  $\sum_j \varphi_j$  :

$$\tilde{\varphi}_r(\mathbf{a}) = \frac{\varphi(\|\mathbf{a} - \mathbf{c}_r\|_{\Sigma_r})}{\sum_{j=1}^R \varphi(\|\mathbf{a} - \mathbf{c}_j\|_{\Sigma_j})}, \quad \text{avec } \sum_{r=1}^R \tilde{\varphi}_r(\mathbf{a}) = 1$$

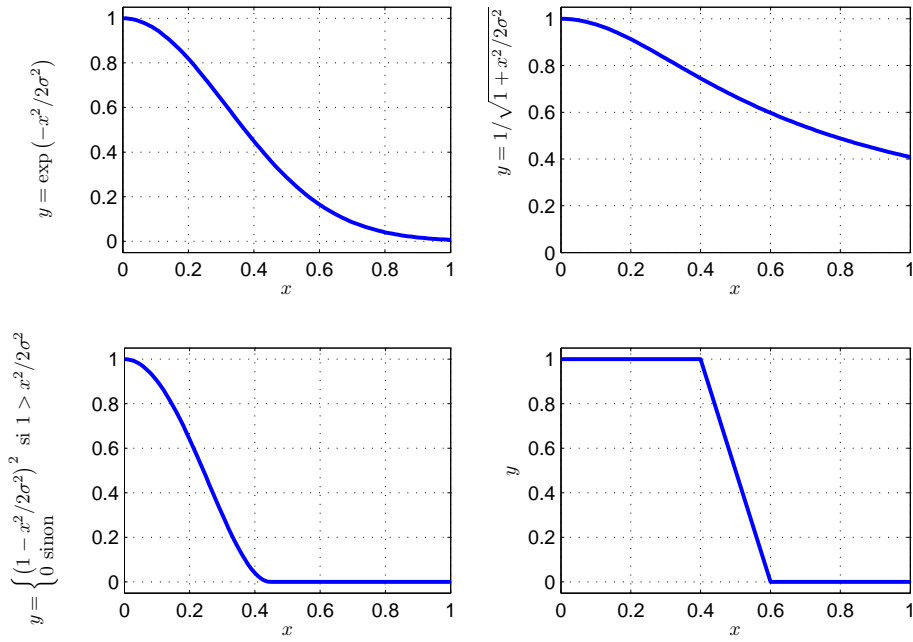


FIG. 4.5: Fonctions à base radiale

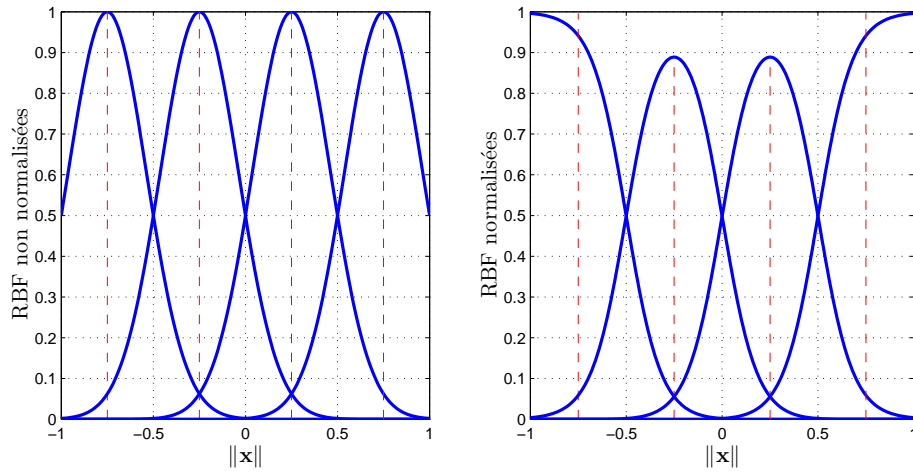


FIG. 4.6: Normalisation des pondérations

Nous pouvons observer deux phénomènes sur la figure 4.6 : une meilleure gestion des bords de domaine et une diminution du poids maximum accordé aux modèles locaux définis « au milieu » du domaine.

#### 4.1.2.3 Calculs des poids

Dans cette approche, les seuls paramètres à calculer sont les paramètres  $\chi_r$  des modèles locaux du MEX. Nous rappelons le critère (2.53) qu'ils doivent vérifier :

$$\mathcal{J}(\mathbf{X}) = \frac{1}{2} \|\Psi\mathbf{X} - \Theta\|^2 \quad (4.9)$$

D'où en utilisant la méthode pseudo-inverse

$$\mathbf{X} = \mathbf{\Psi}^\dagger \mathbf{\Theta} \quad (4.10)$$

Pour calculer la pseudo-inverse, nous allons utiliser la décomposition en valeurs singulières [Golub et Van Loan, 1996] :

THÉORÈME 4.1.1 : Si  $\mathbf{\Psi}$  est une matrice réelle de taille  $N \times M$ , alors il existe deux matrices orthogonales

$$U = (\mathbf{u}_1 \quad \dots \quad \mathbf{u}_N)$$

et

$$V = (\mathbf{v}_1 \quad \dots \quad \mathbf{v}_M)$$

telles que

$$U^\top \mathbf{\Psi} V = \text{diag}(\sigma_1, \dots, \sigma_P), \quad P = \min(M, N) \quad (4.11)$$

où

$$\sigma_1 \geq \dots \geq \sigma_P \geq 0$$

Les  $\mathbf{u}_n$  sont appelés les vecteurs singuliers à gauche de  $\mathbf{\Psi}$  et les  $\mathbf{v}_m$  les vecteurs singuliers à droite de  $\mathbf{\Psi}$ . Les  $\sigma_p$  sont appelées les valeurs singulières de  $\mathbf{\Psi}$ . D'après ce théorème de décomposition, la matrice pseudo-inverse de  $\mathbf{\Psi}$ , de taille  $M \times N$  est définie par

$$\mathbf{\Psi}^\dagger = V \Sigma^+ U^\top \quad (4.12)$$

avec

$$\Sigma^+ = \text{diag} \left( \frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_P}, 0, \dots, 0 \right) \quad (4.13)$$

L'expérience montre que cette méthode est relativement insensible à la régularisation.

### 4.1.3 Placement par algorithme des $k$ -moyennes

Une autre approche pour le placement des centres consiste à utiliser les données d'apprentissage disponibles. Pour cela, nous proposons l'utilisation de l'algorithme des  $k$ -moyennes. Cet algorithme partitionne l'ensemble des données en sous-groupes, aussi homogènes que possibles. Il place ensuite les centres des fonctions dans les régions où sont présentes des données significatives. Soit  $R$  le nombre de centres à trouver. La détermination du nombre optimal  $R$  peut demander quelques essais. Soient  $\{\mathbf{c}_r(n)\}_{r=1, \dots, R}$  les centres des RBF à l'itération  $n$ . L'algorithme effectue les opérations suivantes :

1. **Initialisation** : choisir des valeurs aléatoires pour les positions initiales des centres  $\mathbf{c}_r(0)$ , la seule restriction étant que ces valeurs soient distinctes. Il serait judicieux de choisir des vecteurs de norme faible.
2. **Echantillonnage** : tirer un vecteur  $\mathbf{a}$  de l'espace d'entrée avec une probabilité donnée.  $\mathbf{a}$  est le vecteur d'entrée à l'itération  $n$ .
3. **Test de similarité** : soit  $r(\mathbf{a})$  l'indice du centre le plus proche de  $\mathbf{a}$ .

$$r(\mathbf{a}) = \arg \min_r \|\mathbf{a}(n) - \mathbf{c}_r(n)\|, \quad r = 1, \dots, R \quad (4.14)$$

4. **Mise à jour** : ajuster les centres des RBF en utilisant la règle suivante

$$\mathbf{c}_r(n) = \begin{cases} \mathbf{c}_r(n) + \eta (\mathbf{a}(n) - \mathbf{c}_r(n)) & \text{si } r = r(\mathbf{a}), \\ \mathbf{c}_r(n) & \text{sinon,} \end{cases} \quad (4.15)$$

où  $\eta$  le *taux d'apprentissage* soumis à  $0 < \eta < 1$ .

5. **Itération** : incrémenter  $n$  par 1, et revenir à l'étape 2 et continuer la procédure jusqu'à ce qu'aucune modification notable n'apparaisse dans la position des centres  $\mathbf{c}_r$ .

L'algorithme décrit ci-dessus est un cas particulier d'un procédé d'apprentissage compétitif intitulé *carte auto-organisatrice*.

Les figures 4.7 et 4.8 montrent deux exemples d'utilisations de cet algorithme.

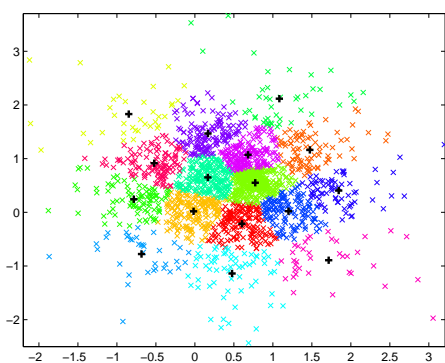


FIG. 4.7: Exemple d'utilisation sur un ensemble de distributions normales

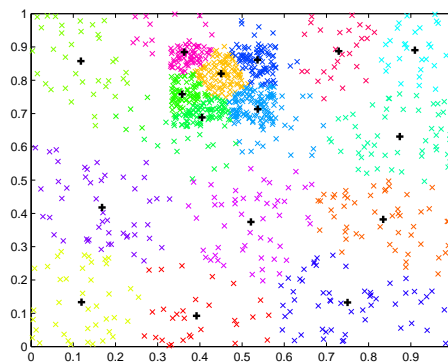


FIG. 4.8: Exemple d'utilisation sur un ensemble de distributions uniformes

Nous pouvons observer que l'algorithme vient augmenter la concentration de centres dans les zones à forte densité de données. Nous pouvons y voir un moyen d'augmenter la « résolution » de notre MEX si nous observons des zones de l'espace d'entrée où il n'est pas suffisamment représentatif de la fonction à estimer. Il faudra donc générer davantage de données dans ces zones. De plus, cet algorithme permet de s'affranchir du « fléau de la dimension » imposé par le maillage régulier, en laissant l'utilisateur choisir le nombre de centres.

Cet algorithme est cependant limité par la représentativité des données utilisées et par son caractère local. En effet, le résultat trouvé dépend fortement des conditions initiales et il est possible de se retrouver « bloqué » dans une région de l'espace avec peu de points. Il existe toutefois des améliorations permettant de retrouver un caractère plus global à la solution.

#### 4.1.4 Placement supervisé des centres

Dans cette approche, tous les paramètres du MEX sont optimisés par apprentissage supervisé de type essais/erreurs. En effet, il est aisé à implémenter en utilisant une descente de gradient qui généralise l'algorithme des moindres carrés.

Tout d'abord, il s'agit de définir la valeur instantanée de la fonction coût :

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k=1}^K \|\varepsilon_k\|^2 \quad (4.16)$$

où  $K$  est le nombre de données d'apprentissage et  $\varepsilon_k$  le signal d'erreur défini par

$$\varepsilon_k = \boldsymbol{\theta}_k - \hat{f}(\mathbf{a}_k) \quad (4.17)$$

$$= \boldsymbol{\theta}_k - \sum_{r=1}^R \varphi(\|\mathbf{a}_k - \mathbf{c}_r\|_{S_r}) \sum_{m=1}^{M+1} \tilde{a}_m \boldsymbol{\chi}_{m,r} \quad (4.18)$$

$$= \boldsymbol{\theta}_k - \sum_{r=1}^R \varphi(\|\mathbf{a}_k - \mathbf{c}_r\|_{S_r}) \tilde{A}_k \boldsymbol{\chi}_r \quad (4.19)$$

L'objectif est ici de déterminer les paramètres  $\boldsymbol{\chi}_{m,r}$ ,  $\mathbf{c}_r$  et  $\Sigma_r^{-1} = S_r^T S_r$  de façon à minimiser  $\mathcal{E}$ . Pour cela, nous allons calculer les gradients du critère en fonction de chacun de ces paramètres.

### Paramètres des modèles locaux

$$\nabla_{\boldsymbol{\chi}_r} \mathcal{E}(n) = - \sum_{k=1}^K \tilde{A}_k^T \mathbf{e}_k(n) \varphi(\|\mathbf{a}_k - \mathbf{c}_r(n)\|_{S_r}) \quad (4.20)$$

$$\boldsymbol{\chi}_r(n+1) = \boldsymbol{\chi}_r(n) - \nu_1 \nabla_{\boldsymbol{\chi}_r} \mathcal{E}(n), \quad r = 1, \dots, R \quad (4.21)$$

### Positions des centres

$$\nabla_{\mathbf{c}_r} \mathcal{E}(n) = -2 \sum_{k=1}^K \mathbf{e}_k^T(n) \tilde{A}_k \boldsymbol{\chi}_r(n) \varphi'(\|\mathbf{a}_k - \mathbf{c}_r(n)\|_{S_r}) \Sigma_r^{-1}(n) (\mathbf{a}_k - \mathbf{c}_r(n)) \quad (4.22)$$

$$\mathbf{c}_r(n+1) = \mathbf{c}_r(n) - \nu_2 \nabla_{\mathbf{c}_r} \mathcal{E}(n), \quad r = 1, \dots, R \quad (4.23)$$

### Taille de RBF

$$\nabla_{\Sigma_r^{-1}} \mathcal{E}(n) = - \sum_{k=1}^K \mathbf{e}_k^T(n) \tilde{A}_k \boldsymbol{\chi}_r(n) B_{k,r} \varphi'(\|\mathbf{a}_k - \mathbf{c}_r(n)\|_{S_r}) \quad (4.24)$$

$$B_{k,r} = (\mathbf{a}_k - \mathbf{c}_r(n)) (\mathbf{a}_k - \mathbf{c}_r(n))^T \quad (4.25)$$

$$\Sigma_r^{-1}(n+1) = \Sigma_r^{-1}(n) - \nu_3 \nabla_{\Sigma_r^{-1}} \mathcal{E}(n), \quad r = 1, \dots, R \quad (4.26)$$

Encore une fois, l'algorithme est limité par la représentativité des données utilisées.

#### 4.1.5 Conclusion

Cette partie nous a permis de mettre à disposition un ensemble de moyens afin de définir la structure des MEX.

## 4.2 Stratégies d'apprentissage en ligne

Nous passons à présent aux méthodes d'apprentissage utilisables dans un contexte temps réel. La référence et les entrées du MEX sont alors des signaux à temps discret que nous notons  $\boldsymbol{\theta}(k)$  et  $\mathbf{a}_k$ . Nous allons au préalable rappeler l'expression de la sortie vectorielle d'un MEX à modèles locaux linéaires, conformément au chapitre 2.

$$\hat{f}(\mathbf{a}(\mathbf{k}), \boldsymbol{\chi}(k)) = \boldsymbol{\psi}(\mathbf{a}(k))^T \boldsymbol{\chi}(k)$$

Afin d'alléger les notations, nous noterons  $\boldsymbol{\psi}(\mathbf{a}(k)) = \boldsymbol{\psi}(k)$ .

### 4.2.1 Moindres carrés récurrents

Etant donné que nous avons résolu des problèmes aux moindres carrés dans la partie précédente, il nous a semblé naturel d'utiliser les variantes récursives, dont celle des moindres carrés récurrents que nous présentons ci-après avec un facteur d'oubli  $\lambda$ .

$$\widehat{\chi}(k) = \widehat{\chi}(k-1) + \Lambda(k) \left( \theta(k) - \boldsymbol{\psi}(k)^\top \widehat{\chi}(k-1) \right) \quad (4.27)$$

$$\Lambda(k) = \frac{\Pi(k-1)\boldsymbol{\psi}(k)}{\lambda + \boldsymbol{\psi}(k)^\top \Pi(k-1)\boldsymbol{\psi}(k)} \quad (4.28)$$

$$\Pi(k) = \frac{1}{\lambda} \left( \Pi(k-1) - \frac{\Pi(k-1)\boldsymbol{\psi}(k) \cdot \boldsymbol{\psi}(k)^\top \Pi(k-1)}{\lambda + \boldsymbol{\psi}(k)^\top \Pi(k-1)\boldsymbol{\psi}(k)} \right) \quad (4.29)$$

$$\Pi^{-1}(k) = \lambda \Pi^{-1}(k-1) + \boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.30)$$

Le critère minimisé à chaque itération par cette procédure est le suivant :

$$\mathcal{J}(\widehat{\chi}(k)) = \sum_{i=1}^k \lambda^{k-i} (\theta(i) - \boldsymbol{\psi}(i)^\top \widehat{\chi}(i))^2 + \lambda^k \left( \widehat{\chi}(k) - \widehat{\chi}(0) \right)^\top \Pi_0^{-1} \left( \widehat{\chi}(k) - \widehat{\chi}(0) \right) \quad (4.31)$$

Ce critère nous permet de mieux cerner le rôle du facteur d'oubli qui consiste à privilégier les mesures proches de l'instant courant par rapport aux conditions initiales.

#### 4.2.1.1 Matrice de covariance et facteur d'oubli

La matrice de covariance est un indicateur de l'erreur paramétrique, c'est-à-dire de la précision de l'estimateur. Il existe quelques suggestions quant au choix de  $\Pi(0)$ , en fonction de la connaissance initiale sur les paramètres [Landau, 2002]. Nous l'initialiserons sous la forme  $\Pi(0) = \pi_0 \cdot I$  où  $\pi_0$  est un scalaire positif et  $I$  la matrice identité.

- S'il n'y a pas d'information a priori sur les paramètres :  $\pi_0 = 1000$  et  $\widehat{\chi}(0) = 0$
- Sinon :  $\pi_0 \ll 1$  et  $\widehat{\chi}(0) = \widehat{\chi}_0$

Remarquons que la trace de la matrice de covariance  $\Pi(k)$  est une mesure de la valeur des gains d'adaptation. Si  $\text{trace}(\Pi(k))$  ne décroît pas de façon significative, l'estimation est alors mauvaise, certainement à cause d'un signal d'excitation non adapté.

Le facteur d'oubli  $\lambda$  est introduit afin de pondérer l'importance accordée à la mesure courante par rapport aux données initiales. En effet, si nous choisissons  $\lambda = 1$ , nous pouvons voir que  $\text{trace}(\Pi(k))$  décroît lorsque  $k$  augmente. De fait, l'erreur d'estimation à un instant donné a de moins en moins d'importance dans le critère relativement à celle des instants précédents. Cette solution est donc à conseiller dans la cas où le système fournissant la référence est stationnaire.

Pour les systèmes instationnaires, il est préférable d'utiliser un facteur d'oubli fixe  $0 < \lambda < 1$ . Ceci permet d'oublier les données initiales, en leur accordant de moins en moins de poids dans le critère à mesure que  $k$  augmente. Typiquement,  $\lambda$  est choisi dans des valeurs comprises entre 0,95 et 0,99.

#### 4.2.1.2 Oubli directionnel par décomposition de la matrice d'information

L'oubli exponentiel est une technique qui a largement fait ses preuves dans le domaine de l'estimation paramétrique. Cependant, lorsque le signal d'entrée n'excite pas suffisamment l'estimateur, on voit apparaître un phénomène de « wind-up » de l'estimateur [Aström



et Wittenmark, 1995]. Des stratégies pour pallier ce problème ont été mises en place et parmi elles, l'oubli directionnel [Cao et Schwartz, 1999], [Cao et Schwartz, 2000], [Kulhavy et Kárny, 1984]. L'idée de base est que seule l'information qui est dans la direction de la mesure doit être oubliée.

Il est possible d'implémenter cette idée sous différentes formes parmi lesquelles nous choisirons la décomposition de la matrice d'information pour la nature élégante de son raisonnement et sa simplicité de mise en œuvre.

Tout d'abord, reprenons (4.30) de mise à jour de la matrice d'information en écrivant  $T_k = \Pi(k)^{-1}$  :

$$T(k) = \lambda T(k-1) + \boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.32)$$

Décomposons  $T$  sous la forme :

$$T(k-1) = T^+(k-1) + T^-(k-1) \quad (4.33)$$

avec

$$T^-(k-1)\boldsymbol{\psi}(k) = 0, \quad \forall \boldsymbol{\psi}(k) \neq 0 \quad (4.34)$$

soit  $\boldsymbol{\psi}(k) \in \text{Ker } T^-(k-1)$ . Un lemme démontré dans [Cao et Schwartz, 2000] nous donne alors :

$$\text{Im } T^-(k-1) \subseteq Z(k)^\perp \quad (4.35)$$

où  $Z(k)^\perp$  est le complémentaire orthogonal de  $\boldsymbol{\psi}(k)$  :

$$Z(k)^\perp = \{\mathbf{x} \mid \langle \mathbf{x}, \boldsymbol{\psi}(k) \rangle = 0\} \quad (4.36)$$

Nous avons alors :

$$T(k-1)\boldsymbol{\psi}(k) = T^+(k-1)\boldsymbol{\psi}(k) \quad (4.37)$$

Dans ces conditions,  $T^-(k-1)$  et  $T^+(k-1)$  ne sont pas uniques et il faut imposer des conditions sur le rang de ces matrices afin de garantir l'unicité de la solution. Etant donné que l'information est mise à jour par  $\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top$  qui est une matrice de rang 1, il est raisonnable d'imposer  $\text{rang}(T^+(k-1)) = 1$  et  $\text{rang}(T^-(k-1)) = \text{rang}(T(k-1)) - 1$  [Cao et Schwartz, 2000].

Nous trouvons :

$$T^+(k-1) = \alpha(k)T(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top T(k-1) \quad (4.38)$$

$$\alpha(k) = \frac{1}{\boldsymbol{\psi}(k)^\top T(k-1)\boldsymbol{\psi}(k)} \quad (4.39)$$

$$T^-(k-1) = T(k-1) - \alpha(k)T(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top T(k-1) \quad (4.40)$$

Il faut s'assurer que  $\alpha(k)$  est non nul. Définissons alors une zone morte pour  $\boldsymbol{\psi}(k)$  :

$$\alpha(k) = 0, \quad \text{si } |\boldsymbol{\psi}(k)| < \epsilon \quad (4.41)$$

Pour revenir à l'idée initiale, nous appliquons l'oubli exponentiel à la seule matrice  $T^+(k-1)$  :

$$T(k-1) = T^-(k-1) + \lambda T^+(k-1) + \boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.42)$$

$$= \bar{T}(k-1) + \boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.43)$$

Nous obtenons :

$$\bar{T}(k-1) = (I - M(k))T(k-1) \quad (4.44)$$

$$M(k) = (1 - \lambda)\alpha(k)T(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.45)$$

où  $N(k) = I - M(k)$  est la matrice d'oubli et remplit une fonction identique à celle du facteur d'oubli  $\lambda$  dans l'oubli exponentiel. Elle est inversible et possède  $n-1$  valeurs propres égales à 1 et une seule égale à  $\lambda$  [Cao et Schwartz, 2000].

Il faut à présent définir l'équation de mise à jour de la matrice de covariance à l'aide du lemme d'inversion (...):

$$\Pi(k) = \bar{\Pi}(k-1) - \frac{\bar{\Pi}(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top\bar{\Pi}(k-1)}{1 + \boldsymbol{\psi}(k)^\top\bar{\Pi}(k-1)\boldsymbol{\psi}(k)} \quad (4.46)$$

où  $\bar{\Pi}(k-1)$  est la matrice de covariance modifiée et définie comme suit :

$$\bar{\Pi}(k-1) = \begin{cases} \Pi(k-1) + \frac{1-\lambda}{\lambda} \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top}{\boldsymbol{\psi}(k)^\top T(k-1)\boldsymbol{\psi}(k)}, & |\boldsymbol{\psi}(k)| > \epsilon \\ \Pi(k-1), & |\boldsymbol{\psi}(k)| \leq \epsilon \end{cases} \quad (4.47)$$

L'algorithme complet s'écrit donc :

$$\hat{\boldsymbol{\chi}}(k) = \hat{\boldsymbol{\chi}}(k-1) + \Lambda(k) \left( \theta(k) - \boldsymbol{\psi}(k)^\top \hat{\boldsymbol{\chi}}(k-1) \right) \quad (4.48)$$

$$\Lambda(k) = \frac{\bar{\Pi}(k-1)\boldsymbol{\psi}(k)}{1 + \boldsymbol{\psi}(k)^\top\bar{\Pi}(k-1)\boldsymbol{\psi}(k)} \quad (4.49)$$

$$\bar{\Pi}(k-1) = \begin{cases} \Pi(k-1) + \frac{1-\lambda}{\lambda} \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top}{\boldsymbol{\psi}(k)^\top T(k-1)\boldsymbol{\psi}(k)}, & |\boldsymbol{\psi}(k)| > \epsilon \\ \Pi(k-1), & |\boldsymbol{\psi}(k)| \leq \epsilon \end{cases} \quad (4.50)$$

$$\Pi(k) = \bar{\Pi}(k-1) - \frac{\bar{\Pi}(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top\bar{\Pi}(k-1)}{1 + \boldsymbol{\psi}(k)^\top\bar{\Pi}(k-1)\boldsymbol{\psi}(k)} \quad (4.51)$$

$$M(k) = \begin{cases} (1 - \lambda) \frac{T(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top}{\boldsymbol{\psi}(k)^\top T(k-1)\boldsymbol{\psi}(k)}, & |\boldsymbol{\psi}(k)| > \epsilon \\ 0, & |\boldsymbol{\psi}(k)| \leq \epsilon \end{cases} \quad (4.52)$$

$$T(k) = (I - M(k))T(k-1) + \boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^\top \quad (4.53)$$

Cette méthode va nous permettre une garantie supplémentaire que l'adaptation du MEX est bien locale de par la sélection de la direction de modification.

### 4.2.2 Extension de l'algorithme SPGL

Revenons à présent au cas de la retouche de loi feed-forward. Au chapitre précédent, nous avons présenté une méthode pour adapter les paramètres de la loi, mise sous forme de filtre en treillis normalisé. Cependant, nous rappelons que le but n'est pas de développer une commande adaptative mais une méthode de retouche locale de paramètres de lois interpolées. Il faut donc « mémoriser » les valeurs optimales. Pour cela, nous allons introduire dans l'algorithme SPGL (cf. paragraphe §3.3.2.3, p. 62) le fait que les coefficients du filtre  $\phi_n$  et  $\nu_n$  sont générés par des MEX, dont les entrées sont les paramètres faisant varier le système global, comme le suggère la figure 4.9.

L'algorithme va donc venir modifier les paramètres des MEX. La modification en elle-même est simple dans la mesure où l'on utilise des MEX à modèles locaux linéaires. Nous

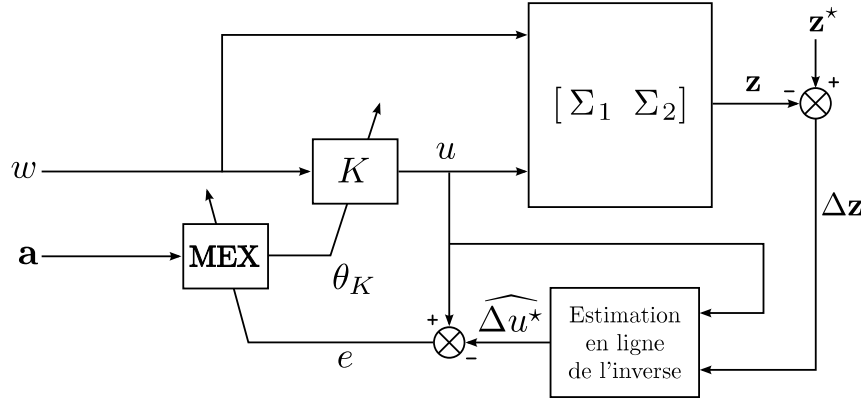


FIG. 4.9: Schéma de retouche en ligne d'une loi feed-forward

avons vu à l'équation 2.52 que l'on peut exprimer la sortie d'un tel MEX linéairement en fonction de ses paramètres. En gardant les mêmes notations, nous pouvons écrire :

$$\nu_n(k) = \psi_{\nu_n}(k)^\top \chi_{\nu_n}(k) \quad (4.54)$$

$$\phi_n(k) = \psi_{\phi_n}(k)^\top \chi_{\phi_n}(k) \quad (4.55)$$

Nous proposons donc une extension de l'algorithme pour le cas où les paramètres du filtre en treillis sont interpolés par un MEX à modèles locaux linéaires :

PROPOSITION 4.2.1 : Les équations de mise à jour des paramètres du réseau se déduisent de (3.138) et (3.139) :

$$\chi_{\nu_n}(k+1) = \chi_{\nu_n}(k) + \eta' \varepsilon(k) \psi_{\nu_n}(k) \nabla_{\chi_{\nu_n}} \varepsilon(k), \quad \forall n = 0, \dots, N \quad (4.56)$$

$$\chi_{\phi_n}(k+1) = \chi_{\phi_n}(k) + \eta' \varepsilon(k) \psi_{\phi_n}(k) \nabla_{\chi_{\phi_n}} \varepsilon(k), \quad \forall n = 1, \dots, N \quad (4.57)$$

*Vérification.* La vérification est très simple. Il suffit d'utiliser le théorème de composition des dérivées. En effet, nous pouvons écrire les équations de mise à jour des paramètres du réseau par LMS :

$$\chi_{\nu_n}(k+1) = \chi_{\nu_n}(k) + \eta' \varepsilon(k) \nabla_{\chi_{\nu_n}} \varepsilon(k), \quad \forall n = 0, \dots, N \quad (4.58)$$

$$\chi_{\phi_n}(k+1) = \chi_{\phi_n}(k) + \eta' \varepsilon(k) \nabla_{\chi_{\phi_n}} \varepsilon(k), \quad \forall n = 1, \dots, N \quad (4.59)$$

Or, la règle de dérivation des fonctions composées nous donne :

$$\begin{aligned} \nabla_{\chi_{\nu_n}} \varepsilon(k) &= \nabla_{\chi_{\nu_n}} \nu_n(k) \times \nabla_{\nu_n} \varepsilon(k) \\ &= \psi_{\nu_n}(k) \nabla_{\nu_n} \varepsilon(k) \end{aligned} \quad (4.60)$$

Nous avons de même :

$$\nabla_{\chi_{\phi_n}} \varepsilon(k) = \psi_{\phi_n}(k) \nabla_{\phi_n} \varepsilon(k) \quad (4.61)$$

Donc en substituant respectivement dans (4.58) et (4.59), les équations (4.60) et (4.61), nous retrouvons les équations de mises à jour.  $\square$

### 4.3 Conclusion

A l'instar du paragraphe §3.3 sur l'identification de loi, nous pouvons voir que la partie hors ligne sert essentiellement à déterminer une structure optimale pour le MEX, l'apprentissage en lui-même étant assez simple. Nous donnerons un exemple plus concret de l'influence des paramètres décrits au paragraphe §4.1 au chapitre 7. Du fait de la structure linéaire des MEX par rapport au vecteur paramètre  $\chi$ , la mise en place des algorithmes d'apprentissage en ligne reste assez simple.



# Chapitre 5

## Régularisation

### Sommaire

---

<b>5.1</b>	<b>Notion de problème mal-posé . . . . .</b>	<b>81</b>
<b>5.2</b>	<b>Cas d'une solution discrète . . . . .</b>	<b>82</b>
<b>5.3</b>	<b>Cas d'une solution continue . . . . .</b>	<b>82</b>
<b>5.4</b>	<b>Détermination du paramètre de régularisation . . . . .</b>	<b>85</b>
5.4.1	Détermination théorique . . . . .	85
5.4.2	Détermination automatique par heuristique . . . . .	87
<b>5.5</b>	<b>Conclusion . . . . .</b>	<b>89</b>

---

Nous avons choisi de faire un chapitre à part sur la régularisation dans la mesure où elle intervient à la fois dans les problématiques de déconvolution et d'apprentissage. Dans le premier cas, nous cherchons à résoudre un problème inverse discret et dans l'autre nous cherchons une solution continue à partir de données discrètes. Dans le cas d'applications réelles, les problèmes peuvent s'avérer *mal-posés*, par opposition à la notion de problème *bien-posé* au sens d'Hadamard que nous définissons dans le paragraphe suivant. La solution directe au problème *mal-posé* n'est pas utilisable en général. Nous allons donc étudier une méthode pour rendre un problème *mal-posé* en un problème *bien-posé*.

### 5.1 Notion de problème mal-posé

DÉFINITION 5.1.1 : *Considérons un espace d'entrée  $\mathcal{X}$  et un espace de sortie  $\mathcal{Y}$ , reliés par une application inconnue  $f$ . Le problème de reconstruction de cette application est dit bien-posé si les trois conditions suivantes sont vérifiées [Tikhonov et Arsenin, 1977] :*

1. Existence

$$\forall \mathbf{x} \in \mathcal{X}, \quad \exists \mathbf{y} \in \mathcal{Y} \mid \mathbf{y} = f(\mathbf{x})$$

2. Unicité

$$\forall (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}^2, \quad f(\mathbf{x}_1) = f(\mathbf{x}_2) \iff \mathbf{x}_1 = \mathbf{x}_2$$

3. Continuité

$$\forall \epsilon > 0, \quad \exists \mu = \mu(\epsilon) \mid d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) < \epsilon \implies d_{\mathcal{Y}}(f(\mathbf{x}_1), f(\mathbf{x}_2)) < \mu(\epsilon)$$

*Si une de ces trois conditions n'est pas vérifiée, le problème est dit mal-posé.*

Dans notre contexte, nous utilisons des données issues de phénomènes physiques et il se peut qu'au sein de ces données, il n'y ait pas suffisamment d'informations pour reconstruire la fonction de façon unique et la présence de bruit peut faire que le critère de continuité n'est pas vérifié. Il est possible de transformer ce problème mal-posé en un problème bien posé, en utilisant la régularisation.

## 5.2 Cas d'une solution discrète

Un cas d'utilisation typique de la régularisation est celui de la résolution d'un problème aux moindres carrés linéaires du type :

$$\text{Calculer } \mathbf{x}^* = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \quad (5.1)$$

La solution au problème directe se calcule en utilisant la condition d'optimalité au premier ordre :

$$\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b} \quad (5.2)$$

La qualité de cette solution est directement liée au conditionnement numérique de la matrice  $\mathbf{A}$ . La régularisation va nous permettre ici d'améliorer le conditionnement de la solution en la « stabilisant ». La méthode proposée par Tikhonov [Tikhonov et Arsenin, 1977] est de résoudre le problème suivant :

$$\text{Calculer } \mathbf{x}_\lambda^* = \arg \min_{\mathbf{x}} \frac{1}{2} \left( \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda \|\mathbf{D}\mathbf{x}\|^2 \right) \quad (5.3)$$

où  $\mathbf{D}$  est un opérateur de régularisation et  $\lambda$  est appelé le paramètre de régularisation. La solution s'écrit alors :

$$\mathbf{x}_\lambda^* = \left( \mathbf{A}^\top \mathbf{A} + \lambda \mathbf{D}^\top \mathbf{D} \right)^{-1} \mathbf{A}^\top \mathbf{b} \quad (5.4)$$

Nous pouvons voir qu'il est possible d'agir sur le conditionnement de la matrice à inverser à l'aide des paramètres  $\lambda$  et  $\mathbf{D}$ . Nous devinons qu'il faudra donc faire un compromis entre la qualité de la solution régularisée vis-à-vis du problème direct initial et sa « stabilité » au sens numérique du terme. C'est l'objet du paragraphe §5.4. Usuellement, l'opérateur  $\mathbf{D}$  peut être choisi égal à la matrice identité où encore à la matrice de dérivation discrète.

## 5.3 Cas d'une solution continue

Dans le cas des réseaux RBF et des MEX, nous cherchons une solution continue à un problème d'interpolation de données discrètes. Nous allons donc généraliser le raisonnement du paragraphe précédent à la recherche de solution continue à un problème aux moindres carrés non-linéaires. Le principe sera à nouveau de stabiliser la solution au problème, initialement mal-posé, en utilisant une fonction auxiliaire positive ou nulle.

Considérons le problème d'interpolation approchée suivant :

$$\text{Calculer } \hat{f}^* : \mathcal{X} \rightarrow \mathbb{R} \mid \hat{f}^* = \arg \min_{\hat{f}} \sum_{k=1}^K \left\| y_k - \hat{f}(\mathbf{x}_k) \right\|^2 \quad (5.5)$$

où les  $(\mathbf{x}_k, y_k)$  sont les couples entrées/sorties à interpoler.

Dans la théorie de la régularisation, la définition du critère à minimiser pour résoudre le problème contient deux termes :

$$\mathcal{J}(\hat{f}) = \mathcal{J}_s(\hat{f}) + \lambda \mathcal{J}_r(\hat{f}) \quad (5.6)$$

$$\mathcal{J}_s(\hat{f}) = \frac{1}{2} \sum_{k=1}^K (y_k - \hat{f}(\mathbf{x}_k))^2 \quad (5.7)$$

$$\mathcal{J}_r(\hat{f}) = \frac{1}{2} \|\mathbb{D}\hat{f}\|^2 \quad (5.8)$$

où  $\mathcal{J}_s(\hat{f})$  est le terme d'erreur standard et  $\mathcal{J}_r(\hat{f})$  est le terme de régularisation.  $\mathbb{D}$  est un opérateur différentiel linéaire et permet d'assurer que la solution au problème est  $\mathcal{C}^\infty$  et donc que la propriété de continuité est vérifiée. Cet opérateur, également appelé stabilisateur, permet d'intégrer au critère une connaissance a priori sur la forme de la solution, ce qui rend son choix dépendant du problème.  $\lambda > 0$  est le paramètre de régularisation et permet d'ajuster le poids relatif des deux termes. En d'autres termes,  $\mathcal{J}_r(\hat{f})$  représente une pénalité sur la complexité de la solution, dont l'influence est modérée par  $\lambda$ .

Notons que  $\|\cdot\|$  dans l'équation (5.8) représente une norme sur l'espace de fonction auquel  $\mathbb{D}\hat{f}$  appartient. En général, il s'agit de l'espace  $\mathcal{L}_2$  (espace des fonctions réelles de carré intégrable).

La minimisation de  $\mathcal{J}(\hat{f})$  passe par l'utilisation de la différentielle de Fréchet de  $\mathcal{J}(\hat{f})$ . Le détail des calculs étant en dehors du périmètre de ce mémoire, nous nous contenterons de présenter le résultat tout en le commentant afin qu'il reste compréhensible. Le lecteur intéressé pourra trouver le détail dans [Haykin, 1999].

Notons  $\hat{f}_\lambda$  la fonction qui minimise (5.6). La condition d'optimalité au premier ordre s'écrit alors :

$$\forall \mathbf{x} \in \mathcal{X}, \quad \mathbb{D}^\# \mathbb{D} \hat{f}_\lambda(\mathbf{x}) = \frac{1}{\lambda} \sum_{k=1}^K (y_k - \hat{f}_\lambda(\mathbf{x}_k)) \delta(\mathbf{x} - \mathbf{x}_k) \quad (5.9)$$

où  $\delta$  est la fonction de Dirac,  $\mathbb{D}^\#$  est l'unique opérateur adjoint de  $\mathbb{D}$  tel que pour toutes fonctions  $u$  et  $v$  suffisamment différentiables, nous avons :

$$\int_{\mathcal{X}} u(\mathbf{x}) \mathbb{D}v(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{X}} v(\mathbf{x}) \mathbb{D}^\#u(\mathbf{x}) d\mathbf{x} \quad (5.10)$$

Nous pouvons considérer que cet opérateur adjoint  $\mathbb{D}^\#$  joue un rôle équivalent vis-à-vis de  $\mathbb{D}$ , à celui de la transposée d'une matrice donnée vis-à-vis de cette matrice.

Le calcul de la solution  $\hat{f}_\lambda$  passe donc par la résolution d'une équation différentielle de la forme

$$\mathbb{L}\hat{f}_\lambda(\mathbf{x}) = h(\mathbf{x}) \quad (5.11)$$

avec

$$\mathbb{L} = \mathbb{D}^\# \mathbb{D} \quad (5.12)$$

et

$$h(\mathbf{x}) = \frac{1}{\lambda} \sum_{k=1}^K (y_k - \hat{f}_\lambda(\mathbf{x}_k)) \delta(\mathbf{x} - \mathbf{x}_k). \quad (5.13)$$



La solution d'une telle équation se calcule en utilisant la fonction de Green  $g(\mathbf{x}, \boldsymbol{\xi})$  de l'opérateur  $L$  (cf. annexe A.2, p.160) :

$$\begin{aligned}\widehat{f}_\lambda(\mathbf{x}) &= \int_{\mathcal{X}} g(\mathbf{x}, \boldsymbol{\xi}) \left( \frac{1}{\lambda} \sum_{k=1}^K (y_k - \widehat{f}_\lambda(\mathbf{x}_k)) \delta(\boldsymbol{\xi} - \mathbf{x}_k) \right) d\boldsymbol{\xi} \\ &= \frac{1}{\lambda} \sum_{k=1}^K (y_k - \widehat{f}_\lambda(\mathbf{x}_k)) \int_{\mathcal{X}} g(\mathbf{x}, \boldsymbol{\xi}) \delta(\boldsymbol{\xi} - \mathbf{x}_k) d\boldsymbol{\xi}\end{aligned}$$

D'où

$$\widehat{f}_\lambda(\mathbf{x}) = \frac{1}{\lambda} \sum_{k=1}^K (y_k - \widehat{f}_\lambda(\mathbf{x}_k)) g(\mathbf{x}, \mathbf{x}_k) \quad (5.14)$$

La solution optimale  $\widehat{f}_\lambda(\mathbf{x})$  au problème de la régularisation est un développement linéaire de  $K$  fonctions de Green, les  $\mathbf{x}_k$  sont les centres de développement et les coefficients  $(y_k - \widehat{f}_\lambda(\mathbf{x}_k))/\lambda$  sont les coefficients du développement. En d'autres termes, la solution appartient à un sous-espace de dimension  $K$  de l'espace des fonctions  $\mathcal{C}^\infty$ , dont une base est constituée par l'ensemble des fonctions de Green centrées en  $\mathbf{x}_k$ .

Les coefficients du développement sont alors donnés par

$$\chi_k = \frac{1}{\lambda} (y_k - \widehat{f}_\lambda(\mathbf{x}_k)), \quad k = 1, \dots, K, \quad (5.15)$$

à l'aide desquels nous réécrivons l'équation (5.14) :

$$\widehat{f}_\lambda(\mathbf{x}) = \sum_{k=1}^K \chi_k g(\mathbf{x}, \mathbf{x}_k) \quad (5.16)$$

En évaluant (5.16) pour chaque  $\mathbf{x}_k$ , et en adoptant les notations suivantes :

$$\begin{aligned}\widehat{\mathbf{f}}_\lambda &= \left( \widehat{f}_\lambda(\mathbf{x}_1) \quad \cdots \quad \widehat{f}_\lambda(\mathbf{x}_K) \right)^\top \\ \mathbf{y} &= (y_1 \quad \cdots \quad y_K)^\top \\ \boldsymbol{\chi} &= (\chi_1 \quad \cdots \quad \chi_K)^\top \\ G &= \begin{pmatrix} g(\mathbf{x}_1, \mathbf{x}_1) & g(\mathbf{x}_1, \mathbf{x}_2) & \cdots & g(\mathbf{x}_1, \mathbf{x}_K) \\ g(\mathbf{x}_2, \mathbf{x}_1) & g(\mathbf{x}_2, \mathbf{x}_2) & \cdots & g(\mathbf{x}_2, \mathbf{x}_K) \\ \vdots & \vdots & \ddots & \vdots \\ g(\mathbf{x}_K, \mathbf{x}_1) & g(\mathbf{x}_K, \mathbf{x}_2) & \cdots & g(\mathbf{x}_K, \mathbf{x}_K) \end{pmatrix}\end{aligned}$$

nous obtenons

$$\boldsymbol{\chi} = \frac{1}{\lambda} (\mathbf{y} - \widehat{\mathbf{f}}_\lambda) \quad (5.17)$$

$$\widehat{\mathbf{f}}_\lambda = G\boldsymbol{\chi} \quad (5.18)$$

d'où par substitution de  $\widehat{\mathbf{f}}_\lambda$

$$(G + \lambda I)\boldsymbol{\chi} = \mathbf{y} \quad (5.19)$$

où  $I$  est la matrice identité de taille  $K$  et  $G$  la matrice de Green. Par ailleurs, l'opérateur  $L$  associé à la fonction de Green  $g$ , est auto-adjoint et nous en déduisons que la matrice de Green  $G$  est symétrique. Nous pouvons donc faire intervenir le théorème de Michelli pour déterminer l'inversibilité de  $G$ , si tous les  $\mathbf{x}_k$  sont distincts. En pratique, il est possible de choisir  $\lambda$  suffisamment grand afin que  $G + \lambda I$  soit inversible. Nous avons donc une solution unique au système (5.19) :

$$\boldsymbol{\chi}^* = (G + \lambda I)^{-1} \mathbf{y} \quad (5.20)$$

Ajoutons pour conclure que les fonctions de Green  $g(\mathbf{x}, \mathbf{x}_k)$  dépendent de la forme de l'opérateur  $D$ , et donc de l'hypothèse faite sur la forme de la solution.

## 5.4 Détermination du paramètre de régularisation

Nous avons pu constater le rôle central du paramètre  $\lambda$  dans la théorie de la régularisation des réseaux RBF. Nous proposons dans ce paragraphe une approche théorique de l'estimation de  $\lambda$ . Une autre méthode plus empirique sera présentée dans un autre contexte dans la partie 3.2.

### 5.4.1 Détermination théorique

Afin de faciliter les notations, nous allons considérer un problème de régression non-linéaire, décrit par un modèle dont les sorties observées  $y_k$  à l'instant  $k$ , suite à une excitation  $\mathbf{x}_k$  sont définies par :

$$y_k = f(\mathbf{x}_k) + \epsilon_k, \quad \forall k = 1, \dots, K \quad (5.21)$$

où  $f$  est  $\mathcal{C}^\infty$  et  $\epsilon_k$  est un bruit blanc centré et de variance  $\sigma^2$ , c'est-à-dire :

$$E\{\epsilon_k\} = 0, \quad \forall k \quad (5.22)$$

$$E\{\epsilon_k \epsilon_m\} = \sigma^2 \cdot \delta_{km} \quad (5.23)$$

avec

$$\delta_{km} = \begin{cases} 1 & \text{si } k = m \\ 0 & \text{sinon} \end{cases} \quad (5.24)$$

où  $E\{\cdot\}$  est l'espérance mathématique. Réécrivons (5.21) sous forme vectorielle :

$$\begin{aligned} \mathbf{y} &= \mathbf{f} + \boldsymbol{\varepsilon} \\ \mathbf{y} &= (y_1 \quad \dots \quad y_K) \\ \mathbf{f} &= (f(\mathbf{x}_1) \quad \dots \quad f(\mathbf{x}_K)) \\ \boldsymbol{\varepsilon} &= (\varepsilon_1 \quad \dots \quad \varepsilon_K) \end{aligned}$$

Le but est de reconstruire la fonction  $f$  à l'aide de l'ensemble  $\mathcal{D}$  des données d'apprentissage. Notons  $\hat{f}_\lambda$  l'estimée régularisée de  $f$  pour une valeur donnée de  $\lambda$ . Elle minimise donc la fonctionnelle de Tikhonov (5.6), que nous adaptons au contexte :

$$\mathcal{J}(\hat{f}) = \frac{1}{2} \sum_{k=1}^K (y_k - \hat{f}(\mathbf{x}_k))^2 + \frac{\lambda}{2} \|\mathcal{D}\hat{f}\|^2 \quad (5.25)$$

Il n'est pas évident de déterminer une valeur de  $\lambda$  permettant d'atteindre à la fois une performance d'apprentissage satisfaisante,  $\sum_{k=1}^K (y_k - \hat{f}(\mathbf{x}_k))^2$  et une solution suffisamment stable,  $\|\mathcal{D}\hat{f}\|^2$ , afin d'avoir une bonne généralisation.

Dans un premier temps, nous allons chercher la valeur de  $\lambda$  qui minimise l'espérance de l'erreur quadratique moyenne  $e_{QM}(\lambda)$  sur l'ensemble des données disponibles :

$$e_{QM}(\lambda) = \frac{1}{K} \sum_{k=1}^K \left( f(\mathbf{x}_k) - \hat{f}_\lambda(\mathbf{x}_k) \right)^2 \quad (5.26)$$

Exprimons  $\hat{f}_\lambda(\mathbf{x}_i)$  sous forme d'une combinaison linéaire des  $y_k$ , tel que :

$$\hat{f}_\lambda(\mathbf{x}_i) = \sum_{k=1}^K \pi_{ik}(\lambda) y_k \quad (5.27)$$

ou encore sous forme matricielle

$$\hat{\mathbf{f}}_\lambda = \Pi(\lambda) \mathbf{y} \quad (5.28)$$

avec

$$\hat{\mathbf{f}}_\lambda = \left( \hat{f}_\lambda(\mathbf{x}_1) \quad \cdots \quad \hat{f}_\lambda(\mathbf{x}_K) \right)^\top \quad (5.29)$$

$$\mathbf{y} = (y_1 \quad \cdots \quad y_K)^\top \quad (5.30)$$

$$\Pi(\lambda) = \begin{pmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1K} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{K1} & \pi_{K2} & \cdots & \pi_{KK} \end{pmatrix} \quad (5.31)$$

La matrice  $\Pi(\lambda)$  de taille  $K \times K$  est appelée la *matrice d'influence*.

L'espérance de l'erreur quadratique moyenne est alors donnée par [Haykin, 1999] :

$$E \{ e_{QM}(\lambda) \} = \frac{1}{K} \left\| (I - \Pi(\lambda)) \mathbf{f} \right\|^2 + \frac{\sigma^2}{K} \text{trace} (\Pi^2(\lambda)). \quad (5.32)$$

Toutefois, cette expression n'est pas très pratique à manipuler étant donné qu'elle nécessite de connaître la fonction  $f$  à approcher. Nous pouvons alors utiliser un estimateur non-biaisé :

$$\hat{e}_{QM}(\lambda) = \frac{1}{K} \left\| (I - \Pi(\lambda)) \mathbf{y} \right\|^2 + \frac{\sigma^2}{K} \text{trace} (\Pi^2(\lambda)) - \frac{\sigma^2}{K} \text{trace} \left( (I - \Pi(\lambda))^2 \right) \quad (5.33)$$

Un choix judicieux pour  $\lambda$  serait alors la valeur qui minimise  $\hat{e}_{QM}(\lambda)$ .

Cependant, pour utiliser cet estimateur, il faut connaître la variance  $\sigma^2$  du bruit, ce qui n'est pas systématiquement le cas. Nous pouvons alors utiliser la *validation croisée* [Craven et Wahba, 1979], qui utilise la technique du « leave-one-out ». Il s'agit de déterminer pour une valeur donnée de  $\lambda$ , la fonction  $\hat{f}_\lambda^{[j]}(\mathbf{x})$  qui minimise le critère suivant :

$$\mathcal{J}(\hat{f}) = \frac{1}{2} \sum_{\substack{k=1 \\ k \neq j}}^K (y_k - \hat{f}(\mathbf{x}_k))^2 + \frac{\lambda}{2} \|\mathcal{D}\hat{f}\|^2. \quad (5.34)$$

En retirant le  $j^{\text{ème}}$  terme, nous pouvons prendre comme indicateur de pertinence de  $\lambda$  la capacité de l'estimateur à prédire ce point manquant, ce qui équivaut à tester la généralisation de l'estimateur sur ce point. Nous introduisons donc la mesure de justesse de  $\lambda$  :

$$V_o(\lambda) = \frac{1}{K} \sum_{j=1}^K \left( f(\mathbf{x}_j) - \hat{f}_\lambda^{[j]}(\mathbf{x}_j) \right)^2, \quad (5.35)$$

qui dépend uniquement des données. L'estimée de validation croisée ordinaire de  $\lambda$  est définie comme étant la valeur qui minimise  $V_o(\lambda)$ . De plus, cette mesure de justesse peut se réécrire de la façon suivante [Haykin, 1999] :

$$V_o(\lambda) = \frac{1}{K} \sum_{j=1}^K \left( \frac{y_j - \hat{f}_\lambda(\mathbf{x}_j)}{1 - \pi_{jj}(\lambda)} \right)^2. \quad (5.36)$$

Cette expression nous permet de mettre en évidence que tous les points ne sont pas traités de la même manière, étant donné que les  $\pi_{jj}(\lambda)$  sont différents. Ce problème peut être contourné en utilisant la *validation croisée généralisée*, où la fonction  $V_o(\lambda)$  est modifiée de la façon suivante :

$$V(\lambda) = \frac{1}{K} \sum_{j=1}^K \omega_j \left( \frac{y_j - \hat{f}_\lambda(\mathbf{x}_j)}{1 - \pi_{jj}(\lambda)} \right)^2 \quad (5.37)$$

avec

$$\omega_j = \left( \frac{1 - \pi_{jj}(\lambda)}{\frac{1}{K} \text{trace}(I - \Pi(\lambda))} \right)^2 \quad (5.38)$$

En utilisant (5.28), la fonction de validation croisée généralisée devient :

$$V(\lambda) = \frac{\frac{1}{K} \|(I - \Pi(\lambda))\mathbf{y}\|^2}{\left( \frac{1}{K} \text{trace}(I - \Pi(\lambda)) \right)^2} \quad (5.39)$$

On peut alors montrer que pour  $K$  suffisamment grand, l'erreur quadratique moyenne  $e_{QM}(\lambda^*)$ , avec

$$\lambda^* = \arg \min_{\lambda} V(\lambda)$$

est proche de son minimum.

### 5.4.2 Détermination automatique par heuristique

Le réglage de  $\lambda$  n'est guidé par aucune connaissance *a priori* bien que nous sachions que l'ensemble des solutions obtenues quand  $\lambda$  parcourt l'intervalle  $[0, +\infty[$ , présente certaines caractéristiques qui peuvent être mises à profit afin d'estimer au moins le bon ordre de grandeur pour  $\lambda$ . Nous rappelons le critère utilisé dans le cas d'une solution discrète :

$$\mathcal{J}\lambda = \mathcal{J}_s(\mathbf{x}) + \lambda \mathcal{J}_r(\mathbf{x}) \quad (5.40)$$

$$\mathcal{J}_s(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \quad (5.41)$$

$$\mathcal{J}_r(\mathbf{x}) = \|D\mathbf{x}\|^2 \quad (5.42)$$

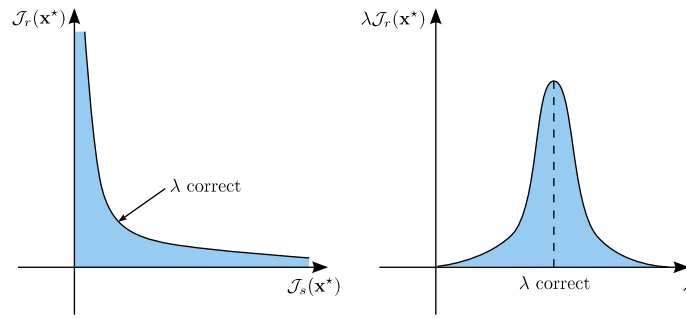


FIG. 5.1: Caractérisation de la solution en fonction de  $\lambda$

La figure 5.1 indique l'allure classique des évolutions des normes de la solution et de l'erreur d'équation. Une bonne valeur de  $\lambda$  est une valeur assez petite pour que  $\mathcal{J}_s(\mathbf{x}^*)$  soit faible, mais pas trop pour que  $\mathcal{J}_r(\mathbf{x}^*)$  soit suffisamment petite.

Quelques heuristiques ont été testées pour localiser la zone des « bonnes » valeurs grâce à quelques considérations théoriques. Dans le cas où  $D = I$ , c'est-à-dire une pondération en amplitude, une analyse par décomposition en valeurs singulières de la matrice  $A$ , permet de vérifier aisément qu'il existe toujours une valeur de  $\lambda$  qui maximise  $\lambda \mathcal{J}_r(\mathbf{x}^*)$  et qu'elle est toujours comprise entre les valeurs singulières extrémales de  $A$ . Ce maximum se situe donc toujours sur la plage « intéressante », dans la dynamique de  $A$ . Nous avons retenu comme borne supérieure à cet intervalle :

$$\lambda^+ = \arg \max_{\lambda} \|\lambda \mathcal{J}_r(\mathbf{x}^*)\| \quad (5.43)$$

Une borne inférieure de cette zone est plus difficile à définir. A la figure 5.2, la courbe bleue, qui représente  $\lambda \mathcal{J}_r(\mathbf{x}^*) / \max_{\lambda} \lambda \mathcal{J}_r(\mathbf{x}^*)$  en fonction de  $\ln \lambda$ , présente généralement un plateau dans une zone intermédiaire où le conditionnement du problème d'inversion est rendu correct grâce à la régularisation, sans que celle-ci soit trop régularisée comme le montre la figure 5.2. Cette zone est cependant difficile à localiser de manière automatique et peut être relativement étroite dans certains cas.

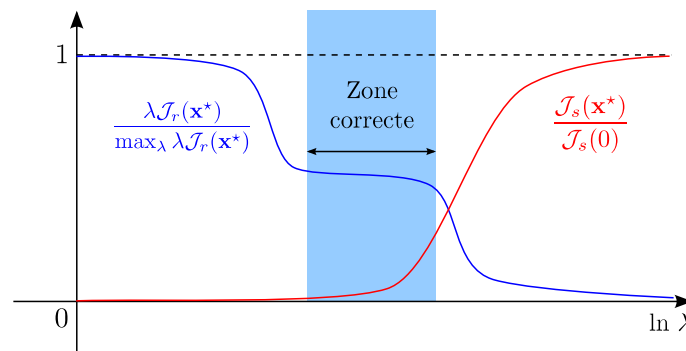


FIG. 5.2: Caractérisation de la solution en fonction de  $\lambda$

Nous retenons finalement comme borne inférieure :

$$\lambda^- = \min_{\lambda} \left\{ \lambda \mid \frac{\lambda \mathcal{J}_r(\mathbf{x}^*)}{\max_{\lambda} \lambda \mathcal{J}_r(\mathbf{x}^*)} = \frac{\mathcal{J}_s(\mathbf{x}^*)}{\mathcal{J}_s(0)} \right\} \quad (5.44)$$

---

Le réglage de  $\lambda$  nécessite donc le calcul des diverses solutions pour évaluer les bornes pour évaluer  $\lambda^-$  et  $\lambda^+$ . Nous choisirons alors la valeur médiane (sur une échelle logarithmique) comme paramètre de régularisation.

## 5.5 Conclusion

Nous avons donc vu que la régularisation permet de stabiliser la solution d'un problème mal posé en venant pénaliser la norme de celle-ci et de ses dérivées. Dans le contexte de l'apprentissage, nous pouvons raisonnablement penser que la régularisation va aider à atteindre une bonne performance de généralisation du MEX. En effet, en se rappelant la figure 1.4 (p. 12), la solution ayant une bonne performance de généralisation est moins « agitée » que celle généralisant mal. Du point de vue de l'inversion, la régularisation va permettre de modérer l'amplitude du signal inverse afin de calculer une solution réalisable physiquement.



Troisième partie

Applications





# Chapitre 6

## Présentation de l'avion

### Sommaire

---

<b>6.1</b>	<b>Repères et notations</b>	<b>93</b>
6.1.1	Les repères	94
6.1.2	Organes de contrôle	95
6.1.3	Récapitulatif	96
<b>6.2</b>	<b>Mécanique du vol</b>	<b>97</b>
6.2.1	Equations générales	97
6.2.2	Modèles d'efforts aérodynamiques	97
6.2.3	Spécificités liées aux moteurs	98
6.2.4	Notion de facteur de charge	99
6.2.5	Notion de centrage	100
6.2.6	Equations longitudinales	100
6.2.7	Equations latérales	101
<b>6.3</b>	<b>Coefficients aérodynamiques</b>	<b>101</b>
6.3.1	Coefficients longitudinaux	102
6.3.2	Coefficients latéraux	102
6.3.3	Modèles précis pour quelques cas particuliers	102

---

CETTE partie a pour but de familiariser le lecteur avec le fonctionnement d'un avion au travers d'une description des notations utilisées pour décrire son comportement. Nous donnerons quelques éléments de la mécanique du vol de l'avion. Enfin, nous donnerons quelques notions et quelques chiffres relatifs aux retours d'expérience Airbus quant aux incertitudes rencontrées sur certains coefficients aérodynamiques. Ces quantités vont nous permettre par la suite de valider notre méthode.

### 6.1 Repères et notations

Pour mettre en œuvre les lois de la mécanique, il faut calculer les accélérations absolues et choisir un repère de projection [Boiffier, 2001]. L'expérience montre que le système d'équations final a tendance à être plus simple lorsqu'il est exprimé dans un repère relatif et de projection proche de l'avion. Nous allons définir ces repères ainsi que leur position par rapport au repère terrestre. Deux repères proches de l'avion sont considérés : un repère lié à la vitesse aérodynamique de l'avion et le repère lié à l'avion.

### 6.1.1 Les repères

**Le repère terrestre** Il est noté :  $\mathcal{R}_o(O, \vec{x}_o, \vec{y}_o, \vec{z}_o)$ . L'axe  $\vec{z}_o$  de ce repère est orienté suivant la direction descendante de la force de gravitation locale en G, centre de gravité de l'avion. Cet axe est ainsi la direction de la force de gravitation vue par l'avion. L'axe  $\vec{x}_o$  est orienté librement mais de façon usuelle, il est orienté en direction du nord géographique ou magnétique. L'origine du repère est un point fixe choisi à la surface de la terre.

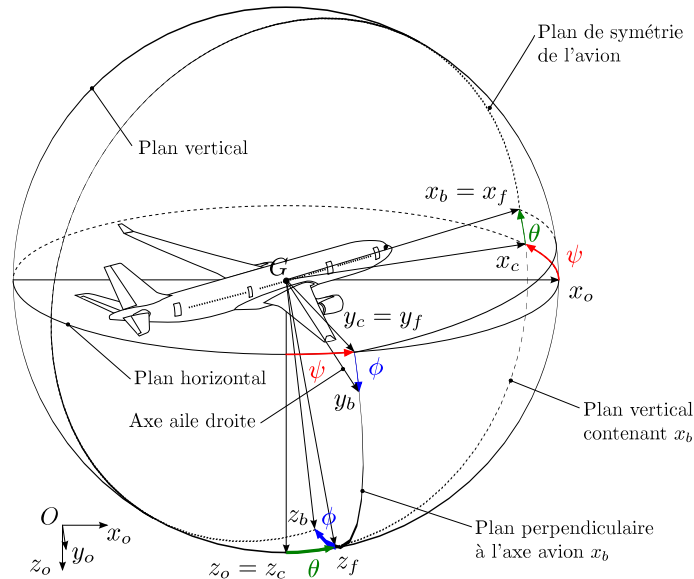


FIG. 6.1: Repère avion  $\mathcal{R}_b$  par rapport au repère terrestre  $\mathcal{R}_o$

**Le repère avion** Il est noté :  $\mathcal{R}_b(G, \vec{x}_b, \vec{y}_b, \vec{z}_b)$ . Ce repère est lié à l'avion. L'origine est le centre de gravité de l'avion, bien qu'il ne soit pas fixe dans l'absolu pendant un vol. L'axe  $\vec{x}_b$ , orienté vers l'avant, appartient au plan de symétrie de l'avion. L'axe  $\vec{z}_b$  est dans le plan de symétrie de l'avion et orienté vers le « ventre » de l'avion. L'axe  $\vec{y}_b$  est orthogonal au plan de symétrie et orienté tel que le repère soit dans le sens direct. La figure 6.1 donne la position de ce repère par rapport à  $\mathcal{R}_o$ . Les angles  $\psi$ ,  $\theta$  et  $\phi$  sont appelés respectivement azimuth, assiette longitudinale et angle de gîte.

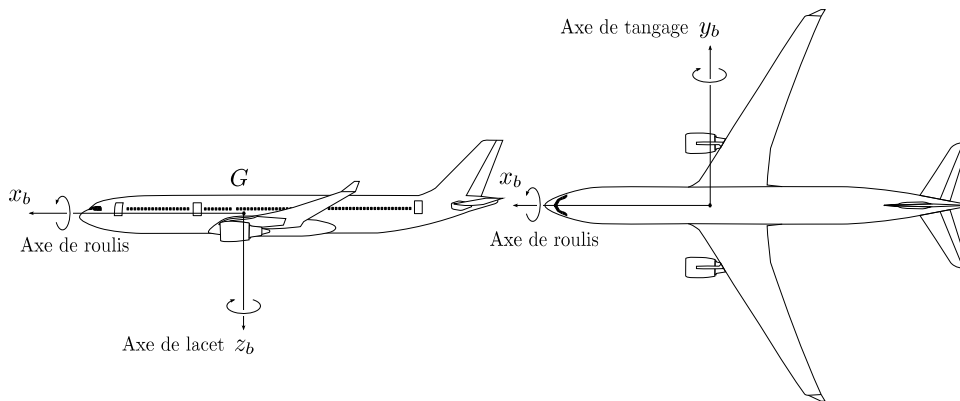


FIG. 6.2: Repère avion  $\mathcal{R}_b$  : vues de profil et de dessus

**Le repère aérodynamique** Il est noté :  $\mathcal{R}_a(G, \vec{x}_a, \vec{y}_a, \vec{z}_a)$ . De même origine que le repère avion  $\mathcal{R}_b$ , le repère aérodynamique  $\mathcal{R}_a$  est défini par  $\vec{x}_a$  porté et orienté par le vitesse aérodynamique de l'avion  $\vec{V}_a$  et deux rotations de passage  $\alpha$  (incidence) et  $\beta$  (dérapage), de  $\mathcal{R}_b$  à  $\mathcal{R}_a$  (cf. figure 6.3). On montre alors que  $\vec{z}_a$  est dans le plan de symétrie de l'avion  $(\vec{x}_b, \vec{y}_b)$ .

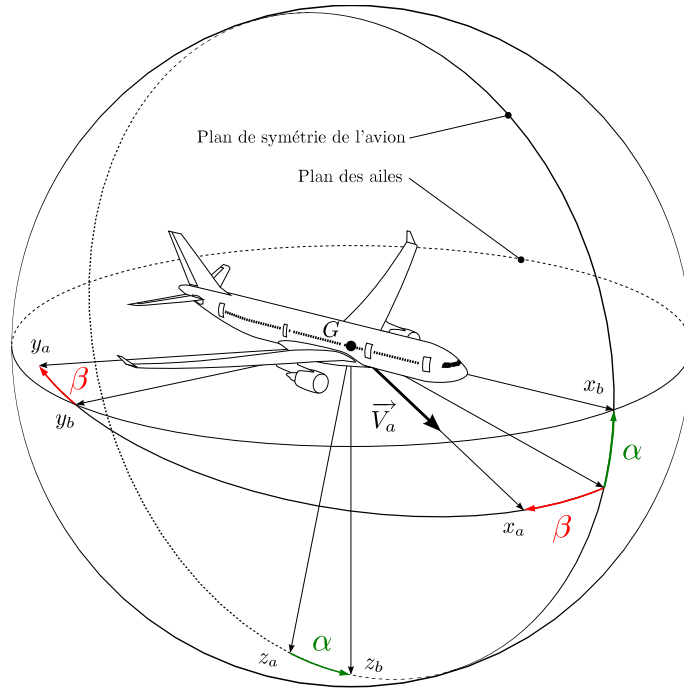


FIG. 6.3: Repère aérodynamique  $\mathcal{R}_a$  par rapport au repère avion  $\mathcal{R}_b$

### 6.1.2 Organes de contrôle

L'avion est constitué d'un certain nombre d'organes de contrôle permettant au pilote d'agir sur l'attitude et la vitesse de l'avion (cf. figure 6.4).

Dans le plan longitudinal, le pilote peut agir sur les moteurs pour venir modifier la poussée, sur la gouverne de profondeur  $\delta q$  pour modifier l'assiette et sur l'ensemble becs/volets pour permettre la sustentation de l'avion aux basses vitesses. Il existe plusieurs positions préprogrammées de becs/volets : elles sont nommées « 0 » (ou lisse), « 1 », « 1+F », « 2 », « 3 » et « FULL ». Nous utiliserons la notation « conf » pour y faire référence. Elles sont utilisées lors des différentes phases de vol (décollage, approche et atterrissage notamment) afin d'optimiser les performances de l'avion. Par la suite, nous appellerons ces positions *configurations aérodynamiques*.

Le pilote peut également utiliser les spoilers  $\delta p_{sp}$  de façon symétrique afin de créer un freinage aérodynamique (cf. chapitre 8). Notons que sur les Airbus, dans un cas nominal, le plan horizontal réglable  $i_h$  n'est pas géré directement par le pilote mais par les lois de pilotage qui vont venir modifier son inclinaison afin d'équilibrer l'avion de façon à ce que l'angle  $\delta q$  entre la gouverne de profondeur et le plan soit nul.

Dans le plan latéral, le pilote peut agir sur les ailerons  $\delta p$  et les spoilers  $\delta p_{sp}$  pour modifier le roulis et sur la dérive  $\delta r$  pour modifier le lacet de l'avion. Les spoilers sont utilisés via une cinématique, lorsque les ailerons n'ont plus l'autorité suffisante pour répondre à la

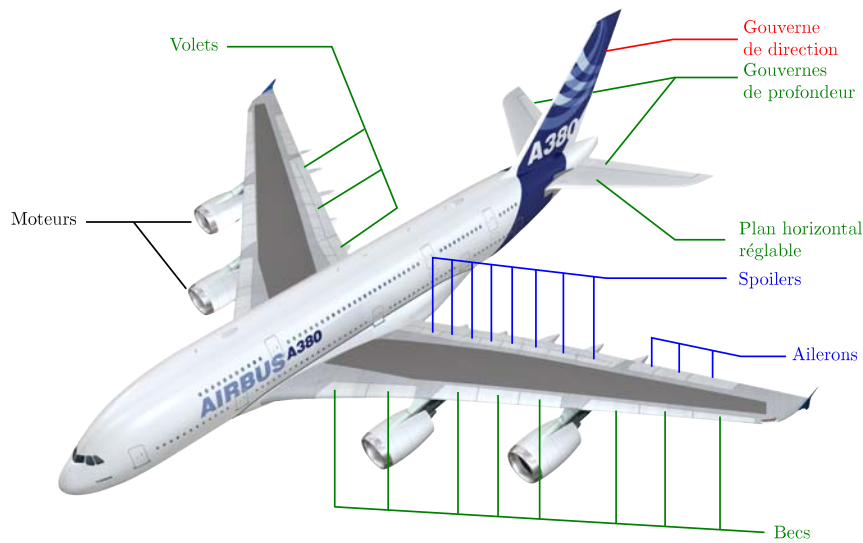


FIG. 6.4: Organes de contrôle

demande du pilote. Notons également que sur certains avions, il y a deux paires d'ailerons : une « interne »  $\delta p_{\text{int}}$  et une « externe »  $\delta p_{\text{ext}}$ .

### 6.1.3 Récapitulatif

La figure 6.5 vient résumer l'ensemble des notations utilisées par la suite ainsi que les conventions de signes pour les divers angles et vitesses de rotation. Elle introduit également quelques notations utiles pour la suite : la pente  $\gamma$  ainsi que les trois composantes du facteur de charge  $N_x$ ,  $N_y$  et  $N_z$  (cf. §6.2.4).

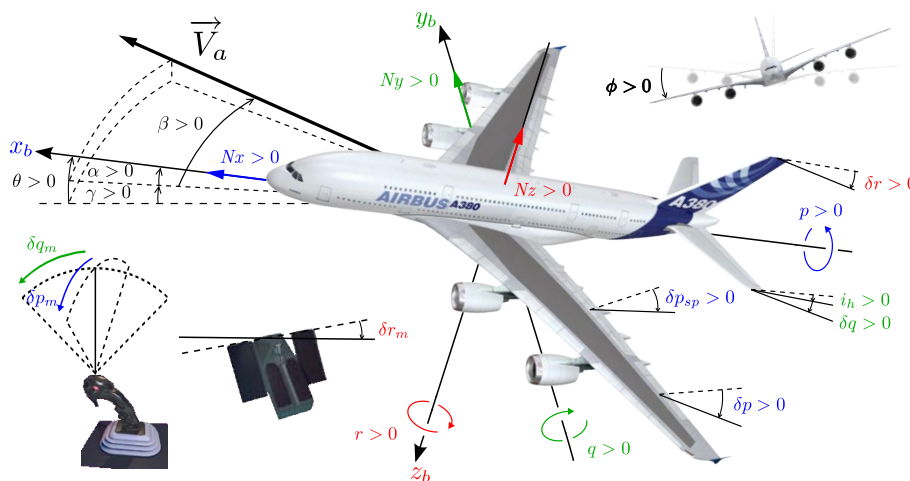


FIG. 6.5: Récapitulatif des notations et conventions de signe

## 6.2 Mécanique du vol

Dans cette partie nous posons les équations générales de la mécanique du vol d'un avion. Toutefois, les démonstrations permettant d'aboutir à ces équations ne sont pas présentées mais le lecteur intéressé pourra les trouver dans [Boiffier, 1998].

### 6.2.1 Equations générales

D'après les équations de la mécanique des solides, nous pouvons exprimer l'accélération  $\mathbf{\Gamma}_{o,G}$  du centre de gravité  $G$  de l'avion dans le repère  $\mathcal{R}_o$  :

$$m\mathbf{\Gamma}_{o,G} = m \frac{d\mathbf{V}^o}{dt} = \mathbf{F}_{\text{aéro}} + \mathbf{F}_{\text{mot}} + m\mathbf{g} \quad (6.1)$$

$$(I_G^b \frac{d\mathbf{\Omega}^o}{dt}) = \mathbf{M}_{\text{aéro},G} + \mathbf{M}_{\text{mot},G} \quad (6.2)$$

avec les notations suivantes :

- $\mathbf{F}_{\text{aéro}}$  : vecteur des forces aérodynamiques
- $\mathbf{F}_{\text{mot}}$  : vecteur des forces liées aux moteurs
- $\mathbf{M}_{\text{aéro},G}$  : vecteur des moment aérodynamiques
- $\mathbf{M}_{\text{mot},G}$  : vecteur des moments liés aux moteurs
- $I_G^b$  : inertie de l'avion par rapport au centre de gravité  $G$  et définie dans  $\mathcal{R}_b$

$$I_G^b = \begin{pmatrix} A & 0 & -E \\ 0 & B & 0 \\ -E & 0 & C \end{pmatrix} \quad (6.3)$$

- vecteur de rotation de l'avion par rapport au référentiel terrestre  $\mathcal{R}_o$ , défini dans  $\mathcal{R}_b$

$$\mathbf{\Omega}^o = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (6.4)$$

où  $p$  est la vitesse de roulis,  $q$  la vitesse de tangage et  $r$  la vitesse de lacet

### 6.2.2 Modèles d'efforts aérodynamiques

Considérons à présent le modèle aérodynamique suivant afin de définir les forces et les moments :

$$\mathbf{F}_{\text{aéro}} = \frac{1}{2} \rho S V_a^2 \mathbf{C}_F \quad (6.5)$$

$$\mathbf{M}_{\text{aéro},G} = \frac{1}{2} \rho S \ell V_a^2 \mathbf{C}_M \quad (6.6)$$

avec les notations suivantes :

- $V_a$  : norme de la vitesse de l'avion par rapport à la masse d'air
- $\rho$  : densité volumique de l'air, fonction de l'altitude
- $S$  : surface aérodynamique de référence (en général celle de l'aile)
- $\ell$  : corde aérodynamique moyenne de la voilure de l'avion définie par :

$$\ell = \frac{2}{S} \int_0^{\frac{b}{2}} c^2(y) dy$$

où  $S$  est la surface de la voilure,  $b$  son envergure,  $y$  la coordonnée le long de l'envergure et  $c(y)$  la corde (distance du bord d'attaque au bord de fuite) en  $y$ .

Nous notons dans le référentiel aérodynamique  $\mathcal{R}_a$  :

$$\mathbf{C}_F = \begin{pmatrix} C_x^a \\ C_y^a \\ C_z^a \end{pmatrix} = \begin{pmatrix} -Cx \\ Cy \\ -Cz \end{pmatrix} \quad (6.7)$$

où  $Cx$  est appelé le *coefficient de traînée*,  $Cy$  le *coefficient de force latérale* et  $Cz$  le *coefficient de portance*

Nous notons dans le référentiel avion  $\mathcal{R}_b$  :

$$\mathbf{C}_M = \begin{pmatrix} C_l^b \\ C_m^b \\ C_n^b \end{pmatrix} = \begin{pmatrix} Cl \\ Cm \\ Cn \end{pmatrix} \quad (6.8)$$

Ces coefficients dépendent essentiellement des angles aérodynamiques  $(\alpha, \beta)$  pour l'orientation générale et des vitesses  $(p, q, r)$  pour les incidences locales. Il dépendent également du nombre de Mach lorsque les effets de compressibilité deviennent sensibles en haut subsonique. Le nombre de Mach dépend quant à lui de la vitesse aérodynamique  $\mathbf{V}_a$  et de la vitesse du son  $a$ , elle-même dépendante de l'altitude.

### 6.2.3 Spécificités liées aux moteurs

Le terme  $\mathbf{F}_{\text{mot}}$  contient plusieurs composantes : un effet principal étant la poussée  $\mathbf{T}$  et deux effets mineurs dus au dérapage  $\beta$  et à l'angle de calage  $\kappa$  des moteurs sur la voilure (cf. figure 6.6)<sup>1</sup> :

- la traînée de captation : il s'agit de la traînée induite par les nacelles moteurs (ici au nombre de 4) ;

$$D_{\text{captation}} = \sum_{i=1}^{N_{\text{mot}}} \|\mathbf{D}_i\| \cdot \sin \beta \quad (6.9)$$

- l'effet de calage moteur : il s'agit de la projection de la poussée sur l'axe  $\vec{y}_b$  ;

$$T_{\text{calage}} = \sum_{i=1}^{N_{\text{mot}}} \|\mathbf{T}_i\| \cdot \sin \kappa_k \quad (6.10)$$

où  $\mathbf{D}_i$  la traînée induite par le  $i^{\text{ème}}$  moteur et  $\mathbf{T}_i$  la poussée générée par le  $i^{\text{ème}}$  moteur. Toutefois, nous pouvons émettre quelques remarques :

- sous l'hypothèse que  $\beta$  est suffisamment faible, l'expression de la traînée de captation est linéaire en  $\beta$ . Nous pouvons alors inclure cet effet dans  $Cy_\beta$ .
- le calage des moteurs ne se remarque qu'en cas de panne moteur asymétrique. De plus, l'angle  $\kappa$  est très faible (environ  $1^\circ$ ). Nous pourrions négliger donc cet effet.

Le terme  $\mathbf{M}_{\text{mot},G}$  provient du fait que sur un avion civil de type Airbus, les moteurs sont situés en dessous du centre de gravité. La force de poussée  $\mathbf{T}$  génère donc un moment cabreur.

<sup>1</sup>L'angle de calage moteur  $\kappa$  a été exagéré sur le schéma pour des raisons de clarté.

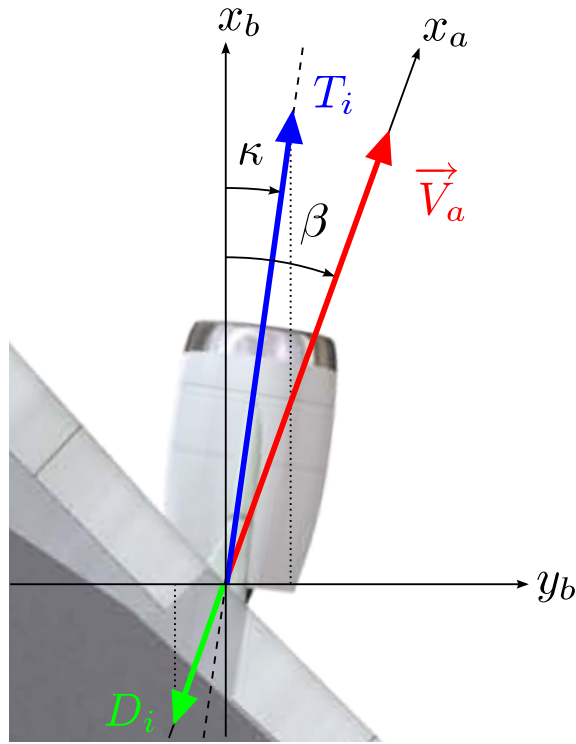


FIG. 6.6: Forces latérales liées aux moteurs

#### 6.2.4 Notion de facteur de charge

Nous introduisons cette notion car elle sera régulièrement utilisée par la suite et elle constitue une donnée mesurable à bord d'un avion, via les IRS. De façon simplifiée, le facteur de charge traduit l'accélération de l'avion, ramenée à l'accélération de gravité.

En reprenant l'équation (6.1), il est possible de regrouper les termes proportionnels à la masse, que nous noterons  $\mathbf{F}_{\text{masse}}$  :

$$m \left( \mathbf{g} - \frac{d\mathbf{V}^o}{dt} \right) + \mathbf{F}_{\text{aéro}} + \mathbf{F}_{\text{mot}} = 0 \quad (6.11)$$

$$\mathbf{F}_{\text{masse}} = m \left( \mathbf{g} - \frac{d\mathbf{V}^o}{dt} \right) \quad (6.12)$$

Ces forces de masse sont égales à la somme du poids et de la force d'inertie. La forme suivante est proposée pour ces forces de masse, qui vont être exprimées proportionnellement au poids.

$$\mathbf{F}_{\text{masse}} = mg\mathbf{N} = m \left( \mathbf{g} - \frac{d\mathbf{V}^o}{dt} \right) \quad (6.13)$$

d'où la définition du facteur de charge (au centre de gravité)

$$\mathbf{N} = \frac{\mathbf{g} - \mathbf{\Gamma}_{o,G}}{g} \quad (6.14)$$



L'équation (6.14) constitue la définition du facteur de charge, que nous exprimerons généralement dans le repère avion  $\mathcal{R}_b$  :

$$\mathbf{N} = \begin{pmatrix} Nx \\ Ny \\ Nz \end{pmatrix}$$

### 6.2.5 Notion de centrage

Cette notion permet de situer le centre de gravité  $G$  de l'avion par rapport à un point fixe  $F$  de celui-ci, le *foyer*. Le foyer d'un avion est le point où s'appliquent les *variations* de portance liées à des *variations* d'incidence. De plus, le centre de gravité varie au cours d'un vol selon le chargement de l'avion, la consommation de carburant, etc. La position du centre de gravité par rapport au foyer va affecter à la fois la stabilité et la manœuvrabilité de l'avion : si  $G$  est devant  $F$ , l'avion est alors statiquement stable mais peu manœuvrant ; si au contraire,  $G$  est derrière  $F$ , l'avion est alors statiquement instable mais très manœuvrant. Les ingénieurs responsables des qualités de vol définissent alors une plage de centrage pour laquelle l'avion est naturellement stable et manœuvrant.

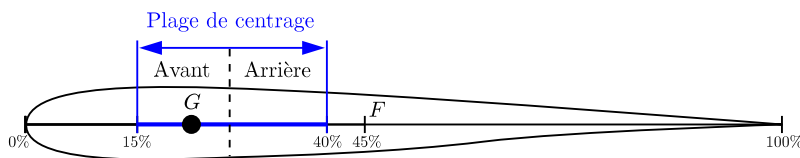


FIG. 6.7: Position du centre de gravité par rapport au foyer et plage de centrage

La position est exprimée en % de la corde aérodynamique moyenne depuis le bord d'attaque (cf. figure 6.7).

### 6.2.6 Equations longitudinales

Ces équations sont projetées dans le repère aérodynamique  $\mathcal{R}_a$  et sont posées pour le cas du vol longitudinal pur. Ce dernier est obtenu lorsque les forces extérieures appliquées à l'avion appartiennent au plan de symétrie de l'avion et que les moments sont perpendiculaires à ce plan. Les diverses forces en jeu sont représentées à la figure 6.8. Plus concrètement, nous considérons que  $\phi = 0$ ,  $\beta = 0$  et  $p = r = 0$ .

– Equation de propulsion, sur l'axe  $\vec{x}_a$

$$m\dot{V} = -\frac{1}{2}\rho S V_a^2 C_x + T \cos \alpha - mg \sin \gamma \quad (6.15)$$

– Equation de sustentation, sur l'axe  $\vec{z}_a$

$$-mV\dot{\gamma} = -\frac{1}{2}\rho S V_a^2 C_z - T \sin \alpha + mg \cos \gamma \quad (6.16)$$

– Equation de moment, autour de l'axe  $\vec{y}_a$

$$B\dot{q} = \frac{1}{2}\rho S \ell V_a^2 C_m + M_{\text{mot},G} \quad (6.17)$$

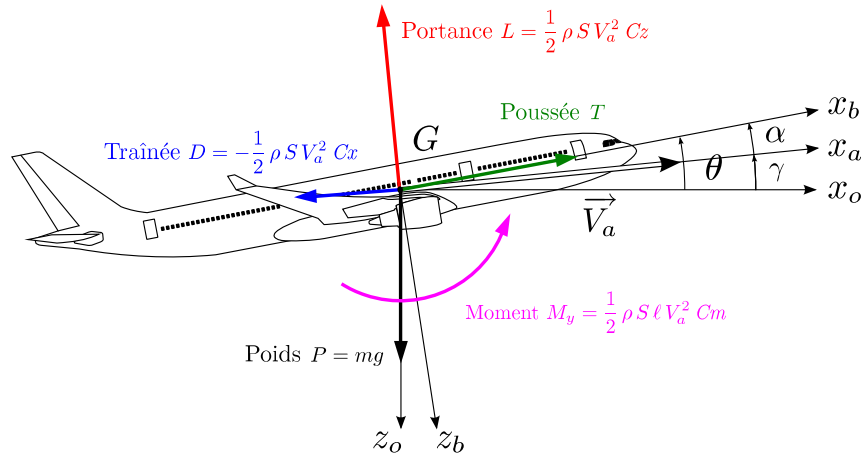


FIG. 6.8: Equations longitudinales – Schéma des forces en jeu

- Equation de cinématique de vitesse de tangage, sur l'axe  $\vec{y}_a$

$$\dot{\alpha} + \dot{\gamma} = q \quad (6.18)$$

- Equation de cinématique d'altitude, sur l'axe  $\vec{z}_o$

$$\dot{h} = V \sin \gamma \quad (6.19)$$

### 6.2.7 Equations latérales

Ces équations sont projetées dans le repère avion  $\mathcal{R}_b$  et nous faisons l'hypothèse que la vitesse  $V_e$ , l'incidence  $\alpha_e$  et la vitesse de tangage  $q_e$  sont constantes.

- Equation de moment de roulis autour de l'axe  $\vec{x}_b$

$$A\dot{p} - (B - C)q_e r - E(\dot{r} + pq_e) = \frac{1}{2}\rho S l V_a^2 C_l \quad (6.20)$$

- Equation de moment de lacet autour de l'axe  $\vec{z}_b$

$$C\dot{r} - (A - B)q_e p - E(\dot{p} + rq_e) = \frac{1}{2}\rho S l V_a^2 C_n \quad (6.21)$$

- Equation de force latérale sur l'axe  $\vec{y}_b$

$$mV_e \left( \dot{\beta} + r \cos \alpha_e + p \sin \alpha_e \right) = mg \sin \phi \cos \theta_e + \frac{1}{2}\rho S V_a^2 C_y \quad (6.22)$$

- Equation cinématique de gîte autour de l'axe  $\vec{x}_b$

$$\dot{\phi} = p + \tan \theta_e (r \cos \phi + q_e \sin \phi) \quad (6.23)$$

## 6.3 Coefficients aérodynamiques

Nous allons à présent rentrer davantage dans le détail des coefficients aérodynamiques, en proposant une décomposition générale, puis une modélisation plus précise utilisée dans les moyens de simulation. Ce niveau de détail va nous permettre de donner des ordres de grandeurs des incertitudes observées sur certaines composantes des coefficients, basés sur le retour d'expérience des ingénieurs chargés de l'identification des avions.

### 6.3.1 Coefficients longitudinaux

Il s'agit des  $Cx$ ,  $Cz$  et  $Cm$ . En première approximation, ils sont modélisés de la façon suivante [Boiffier, 2001] :

$$Cz = Cz_\alpha(\alpha - \alpha_0) + Cz_{\delta q} \cdot \delta q \quad (6.24)$$

$$Cx = Cx_0 + k_i Cz^2 \quad (6.25)$$

$$Cm = Cm_0 + Cm_\alpha(\alpha - \alpha_0) + Cm_{\delta q} \cdot \delta q \quad (6.26)$$

Les  $Cz$  et  $Cm$  sont ici considérés comme des fonctions affines<sup>2</sup> de l'incidence  $\alpha$  et du braquage de la gouverne de profondeur  $\delta q$ . Les  $Cz_\alpha$  et  $Cm_\alpha$  (respectivement  $Cz_{\delta q}$  et  $Cm_{\delta q}$ ) sont donc les gradients de  $Cz$  et  $Cm$  par rapport à l'incidence  $\alpha$  (respectivement le braquage  $\delta q$ ).  $\alpha_0$  est l'incidence pour laquelle le  $Cz$  est nul et le  $Cm_0$  est le moment pur de l'avion.

Le  $Cx$  est une fonction paire de l'incidence et le modèle présenté en (6.25) se nomme « la polaire ».  $k_i$  est le coefficient de traînée induite et  $Cx_0$  le coefficient de traînée à portance nulle.

### 6.3.2 Coefficients latéraux

Il s'agit des  $Cy$ ,  $Cl$  et  $Cn$ . Ils dépendent d'abord des incidences : l'incidence latérale générale  $\beta$  de l'avion et les incidences locales  $(p, q, r)$ . Les commandes de roulis  $\delta p$  et de lacet  $\delta r$  auront également une influence sur ces coefficients. Ces derniers, liés aux axes avion  $\vec{x}_b$ ,  $\vec{y}_b$  et  $\vec{z}_b$ , sont linéarisés en première approximation par rapport aux incidences de la façon suivante [Boiffier, 2001] :

$$Cy = Cy_\beta \cdot \beta + \frac{\ell}{V_a} (Cy_p \cdot p + Cy_r \cdot r) + Cy_{\delta p} \cdot \delta p + Cy_{\delta r} \cdot \delta r \quad (6.27)$$

$$Cl = Cl_\beta \cdot \beta + \frac{\ell}{V_a} (Cl_p \cdot p + Cl_r \cdot r) + Cl_{\delta p} \cdot \delta p + Cl_{\delta r} \cdot \delta r \quad (6.28)$$

$$Cn = Cn_\beta \cdot \beta + \frac{\ell}{V_a} (Cn_p \cdot p + Cn_r \cdot r) + Cn_{\delta p} \cdot \delta p + Cn_{\delta r} \cdot \delta r \quad (6.29)$$

Par exemple, le  $Cy_\beta$  représente le gradient de force latérale par rapport à  $\beta$ . Les vitesses de rotation  $p$  et  $q$  sont adimensionnées par le facteur  $\ell/V_a$  afin de garder les coefficients aérodynamiques sans dimension.

### 6.3.3 Modèles précis pour quelques cas particuliers

Les équations de (6.24) à (6.29) donnent les dépendances dominantes de ces coefficients aérodynamiques. Ces modèles sont essentiellement utilisés pour des raisonnements qualitatifs. Dès lors que l'on souhaite établir un modèle plus précis, il faut tenir compte de tous les effets aérodynamiques jusque là négligés : modélisation des décrochages, effets des becs et volets, des spoilers, de présence des mâts moteurs sur le bord d'attaque, du givre, etc. Nous présentons dans les paragraphes suivants deux cas particuliers que nous allons être amenés à utiliser dans les chapitres 7 et 8.

---

<sup>2</sup>Ce modèle n'est valable que pour une certaine plage d'incidence. En effet, aux fortes incidences, apparaît le phénomène de décrochage induisant une perte de portance par décollement de l'écoulement sur l'extrados de l'aile

### 6.3.3.1 Cas du $C_y$

Le  $C_y$  donné en (6.27), peut être affiné de la façon suivante :

$$C_y = C_{y_\beta} \cdot \beta + \Delta C_{y_\beta} \cdot \delta_{hl} + \frac{\ell}{V_a} (C_{y_p} \cdot p + C_{y_r} \cdot r) + \Delta C_{y_{\delta p}} + \Delta C_{y_{\delta r}} \\ + \frac{\ell}{V_a} C_{y_{\dot{\beta}}} \cdot \dot{\beta} + \Delta C_{y_{\text{spoilers}}} + \Delta C_{y_\phi} + \Delta C_{y_{\delta q}} + \Delta C_{y_{\text{divers}}} \quad (6.30)$$

avec

$$\delta_{hl} = \begin{cases} 0 & \text{si l'avion est en configuration lisse} \\ 1 & \text{si l'avion est hyper-sustenté.} \end{cases}$$

Par la suite nous allons négliger les termes de la deuxième ligne car leur contribution est soit faible par rapport aux termes prédominants donnés (6.27), soit très spécifique (effet des trains, des inverseurs de poussée, etc.). L'intérêt principal de cette décomposition est d'introduire la notation  $\Delta C_{y_\star}$  : il s'agit d'un effet non-linéaire additif sur le coefficient  $C_y$ , dû à la grandeur  $\star$ . Toutefois, nous verrons à la fin du chapitre 7 qu'il est envisageable de conserver le terme en  $\dot{\beta}$ .

Cette approximation peut être encore affinée si nous tenons compte du fait que le  $C_{y_r}$  peut être décomposé sous la forme :

$$C_{y_r} = C_{y_{r,25\%}} + \frac{x_{cg} - 25}{100} C_{y_\beta} \quad (6.31)$$

où  $x_{cg}$  représente la position (en %) du centre de gravité de l'avion sur la corde aérodynamique moyenne.

### 6.3.3.2 Effet des spoilers sur les coefficients longitudinaux

Pour les coefficients longitudinaux, nous avons une décomposition du type :

$$C_{\star_{\text{global}}} = C_\star + \sum_k \Delta C_{\star_k}, \quad \star = x, z \text{ ou } m \quad (6.32)$$

où  $C_\star$  est un des coefficients donnés en (6.24), (6.25) et (6.26) et  $\Delta C_\star$  un effet non-linéaire additif de  $C_\star$  dû à la grandeur  $k$  (spoilers, becs/volets, givre, etc.). Nous allons ici nous intéresser aux effets de spoilers car ce sont les coefficients sur lesquels nous allons raisonner dans le chapitre 8. L'effet est décomposé de la façon suivante :

$$\Delta C_{m_{\text{sp}}} = \Delta C_{m_{\text{sp},\alpha=0}} + \frac{\partial \Delta C_{m_{\text{sp}}}}{\partial \alpha} \alpha + \Delta C_{m_{\text{sp},\text{nl}}} \quad (6.33)$$

$$\Delta C_{m_{\text{sp},\alpha=0}} = f_1(\text{conf}, \text{Mach}, \delta p_{\text{sp}}) \quad (6.34)$$

$$\frac{\partial \Delta C_{m_{\text{sp}}}}{\partial \alpha} = f_2(\text{conf}, \text{Mach}, \delta p_{\text{sp}}) \quad (6.35)$$

$$\Delta C_{m_{\text{sp},\text{nl}}} = f_3(\alpha, \text{conf}, \text{Mach}, \delta p_{\text{sp}}) \quad (6.36)$$

Regardons dans un premier temps sur la figure 6.9, le coefficient de tangage à incidence nulle et le gradient de moment par rapport à l'incidence, en fonction de leurs dépendances respectives.

Nous remarquons que le gradient de  $C_m$  par rapport à  $\alpha$  varie fortement à haut Mach ce qui peut nous indiquer les zones où nous devons augmenter la densité de données

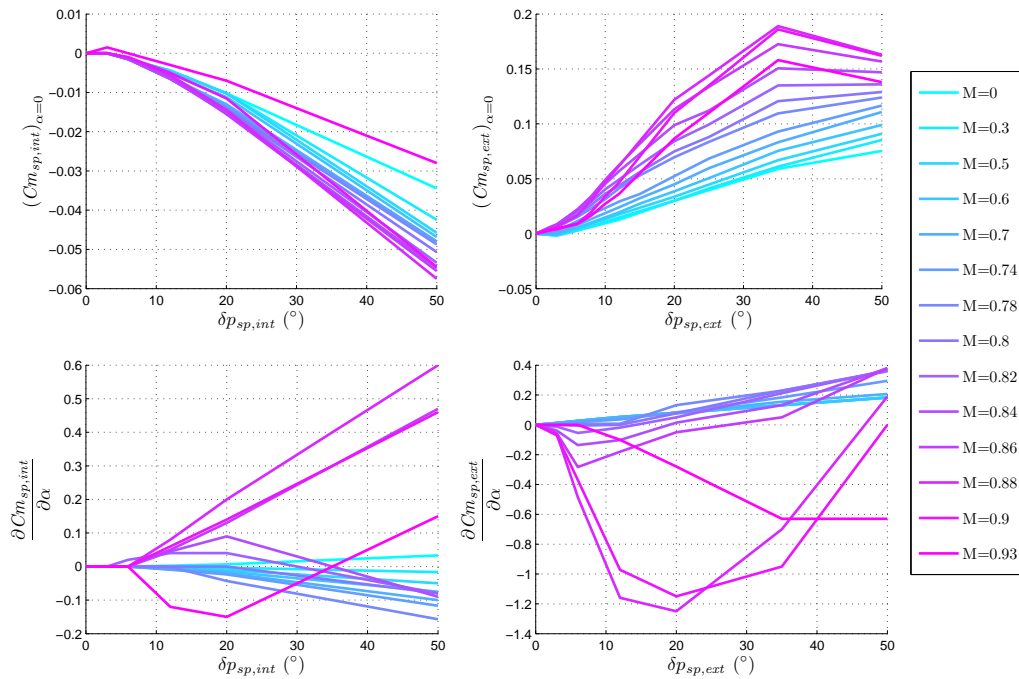


FIG. 6.9:  $C_m$  spoilers interne et externes à  $\alpha = 0$  et  $\frac{\partial C_{m_{sp,i}}}{\partial \alpha}$

d'apprentissage, même si ce sont des zones en dehors de l'enveloppe de vol nominales (Mach  $> 0,86$ ), afin de couvrir certains cas de vol particuliers.

Afin de tester nos méthodes, nous avons besoin d'ordres de grandeur réalistes sur les incertitudes de ces coefficients aérodynamiques.

Nous avons donc profité des retours d'expériences des précédents programmes afin de quantifier le pourcentage de recalage des modèles prévisionnels permettant un recoupement satisfaisant avec les enregistrements d'essais. Ces recalages sont effectués à la suite de vols d'essais spécifiques où la réponse de l'avion à l'extension des A/F est enregistrée pour toutes les configurations aérodynamiques. Les résultats sont également recoupés avec des vols d'identification du comportement en roulis. D'une manière générale, avant identification, les coefficients sont sous-estimés. Cela est dû à une variation d'effet Reynolds<sup>3</sup> lors des essais en soufflerie.

<sup>3</sup>Le nombre de Reynolds (Re) est un nombre sans dimension utilisé en mécanique des fluides. Il caractérise un écoulement, en particulier la nature de son régime (laminaire, transitoire, turbulent).

# Chapitre 7

## Estimateur de dérapage

### Sommaire

---

<b>7.1</b>	<b>Génération de données</b>	<b>107</b>
7.1.1	Calcul du dérapage	107
7.1.2	Coefficients aérodynamiques	109
7.1.3	Génération des données hors ligne	109
7.1.4	Génération des données en ligne	110
<b>7.2</b>	<b>Apprentissage</b>	<b>112</b>
7.2.1	Apprentissage hors ligne	112
7.2.2	Apprentissage en ligne	119
7.2.3	Localisation des essais	119
7.2.4	Résultats d'apprentissage	120
<b>7.3</b>	<b>Conclusion</b>	<b>124</b>

---

DEPUIS l'A380, Airbus utilise une loi de commande latérale basée sur une régulation du dérapage (angle  $\beta$ ). Jusque là, cet angle n'était pas mesuré sur les avions de série et le retour en  $\beta$  utilisé dans les lois de pilotage était obtenu par des estimateurs dont la précision n'atteignait pas un niveau suffisant. Sur l'A380, pour des raisons de qualité de vol et charges structurales, il est nécessaire d'obtenir une valeur précise de cette information. Par conséquent des sondes spécifiques ont été implantées, comme le montre la figure 7.1 et fournissent la valeur réel du dérapage, c'est-à-dire la valeur permanente mais aussi toutes les variations dues à la turbulence. Il faut donc filtrer cette donnée. Afin de limiter la perte d'information liée à ce filtrage, la valeur mesurée est hybridée avec des valeurs estimées, moins perturbées par la turbulence.

L'introduction de cette valeur estimée dans une loi de pilotage nécessite donc un degré de précision suffisant qui peut être mis en défaut au début du développement d'un nouvel avion. En effet, comme nous le verrons par la suite, l'estimateur utilisé repose sur les équations de la mécanique du vol, présentées au chapitre précédent et nécessite donc la connaissance de certains coefficients aérodynamiques, spécifiques à l'avion. Comme nous l'avons mentionné au début de ce mémoire, ces coefficients sont mal connus à cette étape du développement. Nous allons donc appliquer les méthodologies proposées à la partie précédente au recalage d'un estimateur de dérapage embarqué.

Nous allons aborder au paragraphe §7.1, la question de la génération des données d'apprentissage. Nous justifierons également le modèle utilisé pour estimer de dérapage et comment les MEX interviennent. Nous allons ensuite détailler au paragraphe §7.2 les résultats



FIG. 7.1: *Emplacement des sondes de dérapiage sur l'A380*

liés à l'apprentissage. Dans un premier temps, nous donnerons une étude sur l'influence des divers paramètres d'un MEX (utilisation des données, choix du maillage, de la norme, etc.). Nous donnerons également les résultats obtenus pour les apprentissages hors ligne et en ligne, en comparant la performance de notre estimateur à celui utilisé dans les calculateurs. La méthode est résumée par la figure 7.2.

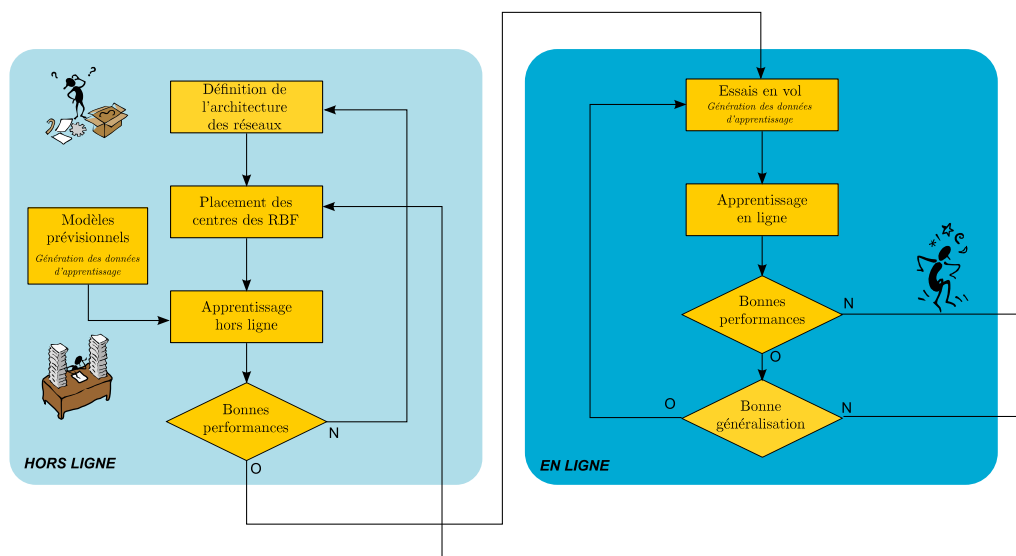


FIG. 7.2: *Les différentes étapes de l'apprentissage*

## 7.1 Génération de données

Le dérapage est une grandeur de la mécanique du vol liée à l'orientation du repère aérodynamique par rapport au repère avion, comme nous l'avons vu au paragraphe §6.1.1. Nous pouvons donc utiliser les équations de la mécanique de vol afin de déterminer une façon de calculer le dérapage. Pour cela nous allons utiliser l'équation de force latérale (6.22) et la décomposition du coefficient aérodynamique de force latérale proposée en (6.27), que nous allons venir modifier et reformuler afin de les rendre plus représentatives.

### 7.1.1 Calcul du dérapage

Tout d'abord, reprenons l'équation de force latérale (6.22), projetée sur l'axe  $\vec{y}_b$  :

$$mV \left( \dot{\beta} + r \cos \alpha_e + p \sin \alpha_e \right) = mg \sin \phi \cos \theta_e + \frac{1}{2} \rho S V_a^2 C_y \quad (7.1)$$

Si nous regroupons les termes fonctions de la masse, nous pouvons introduire le facteur de charge latéral (présenté au paragraphe §6.2.4) :

$$m \left( g \sin \phi \cos \theta_e - V \left( \dot{\beta} + r \cos \alpha_e + p \sin \alpha_e \right) \right) = -\frac{1}{2} \rho S V_a^2 C_y \quad (7.2)$$

$$\iff mgNy = -\frac{1}{2} \rho S V_a^2 C_y \quad (7.3)$$

De plus, d'après le paragraphe §6.3.3, le coefficient de force latérale  $C_y$  peut être décomposé selon l'expression suivante :

$$C_y = C_{y_\beta} \cdot \beta + \Delta C_{y_\beta} + \frac{\ell}{V_a} (C_{y_p} \cdot p + C_{y_r} \cdot r) + \Delta C_{y_{\delta p}} + \Delta C_{y_{\delta r}} \quad (7.4)$$

avec les notations suivantes :

- $C_{y_\beta}$  : le gradient de  $C_y$  par rapport au dérapage  $\beta$
- $\Delta C_{y_\beta}$  : la partie non-linéaire de  $C_y$  due à  $\beta$
- $C_{y_p}$  : le gradient de  $C_y$  par rapport au taux de roulis  $p$
- $C_{y_r}$  : le gradient de  $C_y$  par rapport au taux de lacet  $r$
- $\Delta C_{y_{\delta p}}$  : la partie de  $C_y$  due au braquage des ailerons  $\delta p$
- $\Delta C_{y_{\delta r}}$  : la partie de  $C_y$  due au braquage de la gouverne de direction  $\delta r$

En injectant l'expression (7.4) du  $C_y$  dans (7.3), nous obtenons une première estimation du dérapage :

$$\beta \approx -\frac{1}{C_{y_\beta}} \left( \frac{mg}{P_d S} Ny + \Delta C_{y_\beta} + \Delta C_{y_{\delta r}} + \Delta C_{y_{\delta p}} + \frac{\ell}{V_a} (C_{y_r} \cdot r + C_{y_p} \cdot p) \right) \quad (7.5)$$

où  $P_d$  est la pression dynamique exercée sur l'avion.

$$P_d = \frac{1}{2} \rho V_a^2$$

De plus, nous devons tenir compte de la localisation de la sonde inertielle mesurant le facteur de charge latérale  $Ny$ . En effet, cette sonde est située à l'arrière de l'appareil et mesure  $Ny_{ar}$  et non  $Ny$ . Nous devons donc ramener cette accélération au centre de gravité en utilisant la composition des accélérations. Pour tout point  $M$  de l'avion rigide, nous



pouvons exprimer le facteur de charge au centre de gravité  $G$  de l'appareil en fonction de celui du point  $M$  dans le même repère.

$$\begin{aligned} \mathbf{N}_G &= \mathbf{N}_M + \boldsymbol{\Omega} \wedge \left( \boldsymbol{\Omega} \wedge \frac{1}{g} \overrightarrow{GM} \right) + \left( \frac{d\boldsymbol{\Omega}}{dt} \right) \wedge \frac{1}{g} \overrightarrow{GM} \\ &= \mathbf{N}_M + \frac{1}{g} \begin{pmatrix} q \cdot p \cdot y_{\overrightarrow{GM}} - (q^2 + r^2)x_{\overrightarrow{GM}} + r \cdot p \cdot z_{\overrightarrow{GM}} \\ r \cdot q \cdot z_{\overrightarrow{GM}} - (r^2 + p^2)y_{\overrightarrow{GM}} + p \cdot q \cdot x_{\overrightarrow{GM}} \\ p \cdot r \cdot x_{\overrightarrow{GM}} - (p^2 + q^2)z_{\overrightarrow{GM}} + q \cdot r \cdot y_{\overrightarrow{GM}} \end{pmatrix} \\ &\quad + \frac{1}{g} \begin{pmatrix} \dot{q} \cdot z_{\overrightarrow{GM}} - \dot{r} \cdot y_{\overrightarrow{GM}} \\ \dot{r} \cdot x_{\overrightarrow{GM}} - \dot{p} \cdot z_{\overrightarrow{GM}} \\ \dot{p} \cdot y_{\overrightarrow{GM}} - \dot{q} \cdot x_{\overrightarrow{GM}} \end{pmatrix} \end{aligned}$$

soit, en négligeant les termes du deuxième ordre

$$Ny_G \approx Ny_M + \frac{1}{g} \left( \dot{r} \cdot x_{\overrightarrow{GM}} - \dot{p} \cdot z_{\overrightarrow{GM}} \right) \quad (7.6)$$

En notant  $Ny_{ar} = Ny_M$ , nous pouvons déduire une nouvelle expression du dérapage :

$$\begin{aligned} \beta \approx & -\frac{1}{Cy_\beta} \left( \frac{mg}{P_d S} \left( Ny_{ar} + \frac{1}{g} \left( \dot{r} \cdot x_{ar} - \dot{p} \cdot z_{ar} \right) \right) + \Delta Cy_\beta + \Delta Cy_{\delta r} \right. \\ & \left. + \Delta Cy_{\delta p} + \frac{\ell}{V_a} \left( Cy_{r,25\%} \cdot r + Cy_p \cdot p \right) - \frac{\ell}{V_a} \left( \frac{x_{cg} - 25}{100} \right) \cdot r \right) \quad (7.7) \end{aligned}$$

Cette équation va nous permettre d'aborder le problème en utilisant une approche « boîte grise ». Nous allons donc réécrire l'équation précédente de façon à isoler les paramètres mal connus à modéliser.

$$\begin{aligned} \hat{\beta} = & -\boxed{Cy_\beta^{-1}} \frac{mg}{P_d S} \left( Ny_r + \frac{1}{g} \left( \dot{r} \cdot x_{ar} - \dot{p} \cdot z_{ar} \right) \right) - \delta_{HL} \boxed{\Delta \widetilde{Cy}_\beta} \\ & - \boxed{\Delta \widetilde{Cy}_{\delta p}} - \boxed{\Delta \widetilde{Cy}_{\delta r}} - \frac{\ell}{V_a} \left( \boxed{\widetilde{Cy}_p} \cdot p + \left( \boxed{\widetilde{Cy}_{r,25\%}} + \frac{x_{cg} - 25}{100} \right) \cdot r \right) \quad (7.8) \end{aligned}$$

avec

$$\begin{aligned} \Delta \widetilde{Cy}_\beta &= \frac{\Delta Cy_\beta}{Cy_\beta}, & \Delta \widetilde{Cy}_{\delta p} &= \frac{\Delta Cy_{\delta p}}{Cy_\beta}, \\ \Delta \widetilde{Cy}_{\delta r} &= \frac{\Delta Cy_{\delta r}}{Cy_\beta}, & \widetilde{Cy}_p &= \frac{Cy_p}{Cy_\beta}, \\ \widetilde{Cy}_{r,25\%} &= \frac{Cy_{r,25\%}}{Cy_\beta}. \end{aligned}$$

Les termes encadrés sont des rapports de coefficients aérodynamiques et donc sujets à des imprécisions avant la phase d'identification de l'avion. Nous avons choisi de les modéliser par des MEX. Les autres termes du modèle sont des grandeurs mesurables directement

à partir du modèle de l'avion ou de l'avion réel. Signalons toutefois que les grandeurs  $\dot{p}$  et  $\dot{r}$  ne sont pas mesurées à bord de l'avion réel et seront calculées indirectement comme nous le verrons au paragraphe §7.1.4.

La principale raison pour laquelle nous avons choisi de modéliser des rapports de coefficients, et non les coefficients directement, est pour obtenir une structure d'estimateur explicite linéaire en les paramètres des MEX. En effet, cela a permis d'exploiter la structure déjà linéaire des MEX. Cependant, nous aurions pu envisager de ne pas diviser par  $Cy_\beta$  et ainsi résoudre le problème de façon implicite.

### 7.1.2 Dépendances des coefficients aérodynamiques

Nous présentons ici les dépendances des coefficients aérodynamiques utilisés pour estimer  $\beta$ , telles qu'elles ont été identifiées par des spécialistes en aérodynamique.

$$\begin{aligned} Cy_\beta &= f(\alpha, \text{Mach}, P_d, \text{conf}) & \Delta Cy_\beta &= f(|\beta|, \alpha, \text{conf}) \\ Cy_{r,25\%} &= f(\alpha, \text{Mach}, P_d, \text{conf}) & Cy_p &= f(\alpha, \text{Mach}, P_d, \text{conf}) \\ \Delta Cy_{\delta r} &= f(\alpha, \text{Mach}, \delta r, \text{conf}) & \Delta Cy_{\delta p} &= f(\alpha, \text{Mach}, \delta p_{\text{ext}}, \delta p_{\text{int}}, \text{conf}) \end{aligned}$$

Chaque coefficient est calculé de manière différente selon la configuration aérodynamique de l'avion, à cause d'effets aérodynamiques induits par l'extension des becs et volets. Nous déduisons donc de ces dépendances celles des rapports utilisés pour estimer le dérapage.

Ici, la variable « conf » est une variable discrète dont chaque valeur correspond à une configuration aérodynamique donnée. Nous allons donc construire un estimateur par configuration aérodynamique afin que cette variable discrète ne viennent pas perturber l'apprentissage.

Notons que pour la configuration lisse,  $\Delta Cy_\beta$  est nul et que pour les autres configurations,  $\Delta Cy_{\delta p}$  ne dépend plus du Mach.

Nous voyons apparaître une difficulté qui est que le coefficient  $\Delta Cy_\beta$  dépend du dérapage, grandeur que nous cherchons à estimer ! Pour l'apprentissage, cela ne pose pas de problème étant donné que la mesure est accessible. Le problème se posera donc lorsque l'estimateur sera utilisé de façon autonome. Il ne serait pas cohérent d'utiliser la mesure du dérapage non plus car nous souhaitons utiliser cet estimateur comme source indépendante de la mesure. Nous avons donc rebouclé la grandeur estimée, retardée d'un pas de calcul, sur l'entrée du réseau. Une étude de stabilité devrait être effectuée pour prouver que l'estimateur converge bien qu'aucune instabilité n'ait été remarquée pendant l'utilisation de l'estimateur.

### 7.1.3 Génération des données hors ligne

Dans le contexte hors ligne, les coefficients sont accessibles directement depuis le modèle numérique. Nous pourrions donc effectuer un apprentissage individuel de chaque réseau. Les données sont alors générées aléatoirement à l'aide du modèle numérique pour chaque configuration aérodynamique, selon une loi uniforme affectant le point de vol (le Mach, l'altitude  $z_p$ , la masse, le centrage  $x_{cg}$ ), les variables d'état ( $\alpha$ ,  $\beta$ ,  $p$ ,  $r$ ) ainsi que les gouvernes de l'avion ( $\delta p$ ,  $\delta r$ ,  $\delta p_{sp}$ ). Remarquons que nous ne générons pas directement les variables dont dépendent les coefficients. Cela est dû au formalisme de l'outil utilisé pour générer les données. Nous pourrions récupérer les variables manquantes (les rapports de coefficients,  $P_d$ ,  $Ny_{ar}$ ,  $\dot{p}$ ,  $\dot{r}$  et  $V_a$ ) en sortie.

Une fois les paramètres d'entrée de la simulation générés, nous vérifions la validité du point de vol en comparant la vitesse calculée aux limites imposées par la configuration aérodynamique. Ensuite, nous appliquons au modèle les variables d'état et les entrées de gouvernes pour effectuer une simulation sur un pas de calcul. Pour simplifier, nous pouvons considérer que le modèle d'avion sur un point de vol a la représentation d'état suivante :

$$\begin{aligned}\dot{\mathbf{x}} &= A(\text{Mach}, z_p, \text{masse}, x_{cg})\mathbf{x} + B(\text{Mach}, z_p, \text{masse}, x_{cg})\mathbf{u} \\ \mathbf{y} &= h(\text{Mach}, z_p, \text{masse}, x_{cg}, \dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) \\ \mathbf{x} &= (\alpha \quad \beta \quad p \quad r)^\top \\ \mathbf{u} &= (\delta p \quad \delta r \quad \delta p_{\text{sp}})^\top,\end{aligned}$$

à laquelle nous appliquons les variables précédemment générées pour calculer  $\dot{\mathbf{x}}$  et  $\mathbf{y}$ .

L'outil nous permet de récupérer toutes les grandeurs désirées dans le vecteur  $\mathbf{y}$ , notamment les facteurs de charge  $Nz$  et  $Ny$ , qui nous permettent d'effectuer une deuxième validation du point de vol par rapport aux accélérations subies par l'avion.

Les figures 7.3 et 7.4 représentent les domaines en  $(V_c, \alpha)$  et  $(V_c, \beta)$  générés pour chaque configuration<sup>1</sup>. Les points verts représentent les points validés, les points bleus et rouges sont ceux dehors du domaine en  $(Nz, Ny)$ .

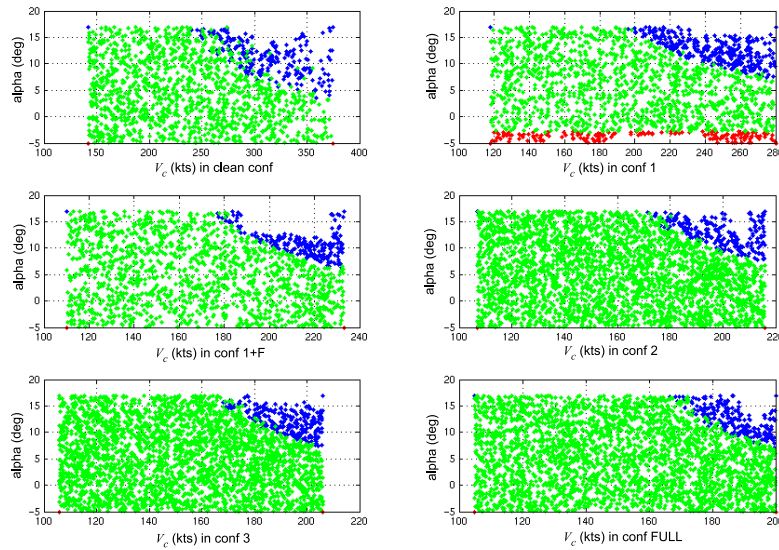


FIG. 7.3: Domaine  $(V_c, \alpha)$  défini par le modèle numérique

#### 7.1.4 Génération des données en ligne

Plusieurs problèmes se posent lorsque nous souhaitons apprendre à partir de données d'essais en vol. D'une part, les coefficients aérodynamiques mis en évidence pour le calcul du dérapage ne sont pas mesurables sur l'avion réel. Il n'est donc plus possible de recalculer directement chaque MEX sur une référence individuelle. Nous allons donc effectuer un apprentissage en parallèle des MEX en utilisant un critère sur l'erreur commise en dérapage.

<sup>1</sup>La  $V_c$  ou vitesse corrigée ou encore CAS, est la vitesse indiquée par l'anémomètre d'un aéronef, corrigée des erreurs de position et d'instruments. Elle est égale à la vitesse vraie dans des conditions standards d'atmosphère et au niveau de la mer (définition tirée de la réglementation de l'aviation civile).

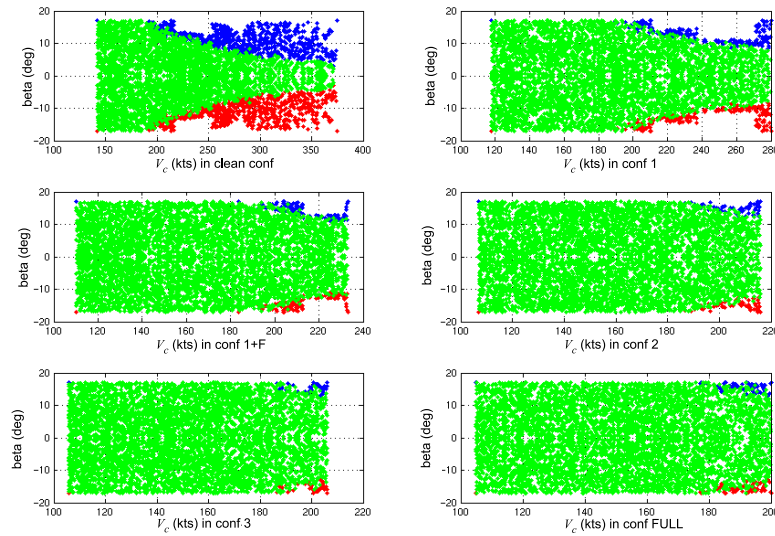


FIG. 7.4: Domaine  $(V_c, \beta)$  défini par le modèle numérique

La structure linéaire posée en (7.8) va alors grandement faciliter la mise en place de cet apprentissage « indirect ».

D'autre part, l'estimation du dérapage utilise des paramètres inertiels  $(Ny_{ar}, p, r)$  qui sont sensibles aux modes souples de l'avion. En effet, pour des avions de grande taille, il n'est plus possible de considérer l'avion comme un solide indéformable. Il faut donc tenir compte de l'existence des modes structuraux dans les lois de pilotage. Afin d'éviter de les exciter, ces paramètres sont filtrés par les filtres suivants :

- Sur  $Ny_{ar}$  et  $r$  :

$$F(s) = \frac{1}{0,0253s^2 + 0,2252s + 1}, \quad f_c = 1\text{Hz}, \xi = 0,7$$

- Sur  $p$  :

$$F(s) = \frac{1}{0,0113s^2 + 0,1501s + 1} \times \frac{0,0083s^2 + 0,01286s + 1}{0,0083s^2 + 0,1286s + 1}$$

$$f_{c,1} = 1,5\text{Hz}, \xi_1 = 0,7$$

$$f_{c,2} = 1,75\text{Hz}, \xi_{2,n} = 0,07, \xi_{2,d} = 0,7$$

Ces filtres ont été calculés lors de la conception de l'estimateur de dérapage utilisé à l'heure actuelle dans les calculateurs.

Enfin nous avons souligné au paragraphe §7.1.1 que les grandeurs  $\dot{p}$  et  $\dot{r}$  ne sont pas mesurées mais calculées indirectement par pseudo-dérivation de  $p$  et  $r$ . Le filtre utilisé est le suivant :

$$F(s) = \frac{s}{0,3s + 1}$$

Les données qui ont été utilisées sont des enregistrements d'essais en vol au nombre de 11, en configuration lisse. Ces données ont servi à identifier le comportement latéral en dérapage de l'avion. Ce sont des vols au cours desquels le pilote fait voler l'avion dans une attitude de dérapage stabilisé.

## 7.2 Apprentissage

Nous allons présenter ici la mise en pratique de l'apprentissage et les résultats obtenus. En parallèle, nous donnons au paragraphe §7.2.1, une étude qualitative sur l'influence des divers paramètres sur la performance d'apprentissage du MEX : la taille de la base de données d'apprentissage utilisée, le placement des centres et le choix de la norme.

### 7.2.1 Apprentissage hors ligne

Le critère de performance considéré dans cette partie est

$$\mathcal{J} = 100 \cdot \frac{\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|}{\|\boldsymbol{\theta}\|} \quad (7.9)$$

$$\boldsymbol{\theta} = \left( \Delta \widetilde{C}y_{\beta} \quad \widetilde{C}y_p \quad \widetilde{C}y_{r,25\%} \quad \Delta \widetilde{C}y_{\delta p} \quad \Delta \widetilde{C}y_{\delta r} \right)^T \quad (7.10)$$

où  $\boldsymbol{\theta}$  est le vecteur de toutes les valeurs générées de la grandeur à estimer  $\theta$  et  $\hat{\boldsymbol{\theta}}$  est le vecteur de toutes les valeurs estimées par un MEX donné.

#### 7.2.1.1 Influence de la taille de la base de données

Les données générées précédemment sont divisées en un ensemble d'apprentissage et un ensemble de validation. Nous avons donc regardé l'influence du nombre de points d'apprentissage sur la performance de validation sur tous les points disponibles. Pour cela, nous avons considéré le rapport de coefficient  $\frac{\Delta C y_{\delta p}}{C y_{\beta}}$  et tiré aléatoirement un certain pourcentage de données sur lesquelles l'apprentissage a été effectué. Pour un pourcentage donné, nous effectuons 100 tirages aléatoires et calculons la moyenne, l'écart-type et le maximum du critère  $\mathcal{J}$  sur ces 100 tirages. Les résultats sont présentés à la figure 7.5. Nous voyons donc

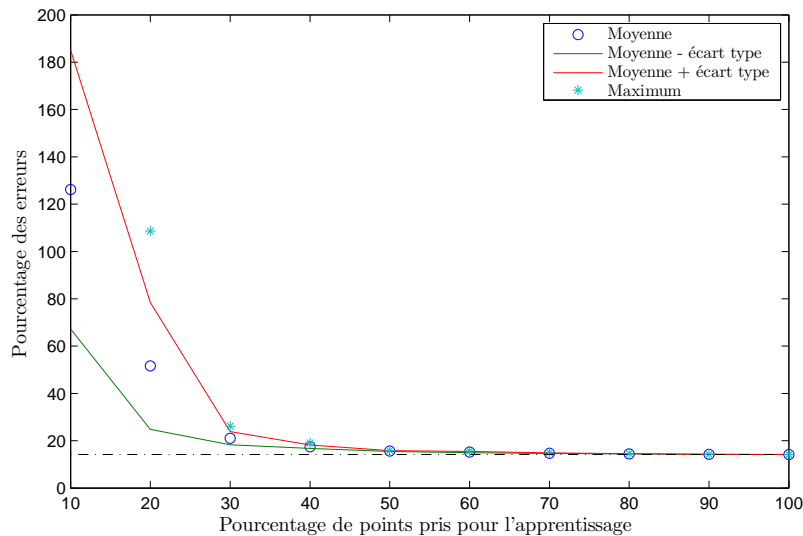


FIG. 7.5: Influence de la taille de la base de données sur la performance de validation

qu'à partir de 40% du nombre total de points, la performance de validation n'est quasiment plus modifiée. Nous garderons donc cette valeur afin d'alléger les calculs par la suite. Ce résultat est toutefois spécifique à la complexité de la fonction à identifier.

### 7.2.1.2 Placement des centres et choix de la norme

Nous avons comparé le placement régulier des centres à celui calculé par l'algorithme des  $k$ -moyennes. De même pour chaque placement, nous avons regardé l'influence de la norme utilisée ( $\|\cdot\|_2$  ou  $\|\cdot\|_\infty$ ). La RBF utilisée pour chaque MEX est une multiquadrique généralisée telle que :

$$\varphi_r(\mathbf{a}) = \frac{1}{\|\mathbf{a} - \mathbf{c}_r\|_{\Sigma_r}^2 + 1}.$$

La raison de ce choix est que nous souhaiterions implémenter cet estimateur dans un calculateur donc il est nécessaire de tenir compte des limites imposées par le vocabulaire d'implémentation (i.e. les fonctions  $\sqrt{(\cdot)}$  et  $\exp(\cdot)$  ne sont pas accessibles).

Dans un premier temps, nous avons élaboré un maillage régulier de l'espace d'entrée de chaque MEX modélisant un rapport de coefficient. C'est une approche au coup par coup qu'il est difficile de systématiser mais nous pouvons mettre en évidence les contraintes qui nous ont orientés vers les structures choisies. Tout d'abord, nous venons d'évoquer la potentielle implémentation de l'estimateur dans un calculateur embarqué. Il faut donc minimiser tant que possible la complexité. Sachant qu'un maillage régulier va générer une complexité exponentielle, nous sommes limités à des maillages « grossiers » (5 mailles grand maximum). Ensuite, nous pouvons utiliser quelques connaissances *a priori* sur les entrées afin de déterminer un maillage approprié. Par exemple, les grandeurs  $\delta r$  et  $\delta p$  ont des domaines symétriques d'évolution. Etant donné que nous utilisons des modèles locaux linéaires, il semble judicieux d'utiliser un nombre de mailles paire sur ces grandeurs, afin de conserver la parité. Nous donnons dans les tableaux 7.1 et 7.2, le nombre de mailles affecté à chaque dépendance pour chacun des MEX.

	$\alpha$	Mach	$P_d$	$\delta p_{\text{ext}}$	$\delta p_{\text{int}}$	$\delta r$
$Cy_\beta^{-1}$	3	3	1	-	-	-
$\widetilde{C}y_p$	3	3	1	-	-	-
$\widetilde{C}y_{r,25\%}$	3	3	1	-	-	-
$\Delta \widetilde{C}y_{\delta p}$	2	1	2	2	2	-
$\Delta \widetilde{C}y_{\delta r}$	3	3	1	-	-	2

TAB. 7.1: Maillages réguliers proposés en configuration lisse

	$\alpha$	Mach	$P_d$	$\delta p_{\text{ext}}$	$\delta p_{\text{int}}$	$\delta r$	$\beta$
$Cy_\beta^{-1}$	3	3	1	-	-	-	-
$\Delta \widetilde{C}y_\beta$	3	2	-	-	-	-	3
$\widetilde{C}y_p$	3	3	1	-	-	-	-
$\widetilde{C}y_{r,25\%}$	3	3	1	-	-	-	-
$\Delta \widetilde{C}y_{\delta p}$	2	-	2	2	2	-	-
$\Delta \widetilde{C}y_{\delta r}$	3	3	1	-	-	2	-

TAB. 7.2: Maillages réguliers proposés en configuration lisse

Pour chaque MEX, nous avons ensuite calculé la matrice  $\Sigma_r$  telle que suggérée en (4.8) :

$$\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_M)$$

$$\sigma_m = \frac{1}{\Delta_m \ln 2} \quad \forall m = 1, \dots, M,$$

où  $\Delta_m$  est le nombre de maille pour la  $m^{\text{ème}}$  dimension de l'espace d'entrée du MEX. Cette matrice  $\Sigma_r$  est identique pour chaque modèle local du MEX.

En parallèle, nous avons élaboré d'autres MEX avec l'algorithme des  $k$ -moyennes, en imposant un nombre de modèles locaux identique de leurs homologues élaborés avec un maillage régulier. Les centres ont été positionnés en utilisant la fonction `kmeans` de MATLAB<sup>®</sup>. La maillage étant ici non régulier, nous avons calculé  $\Sigma_r$  à l'aide de (4.7) :

$$\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_M)$$

$$\sigma_m = \frac{d_{\max}}{\sqrt{2} \cdot R} \quad \forall m = 1, \dots, M,$$

où  $d_{\max}$  est la distance maximale entre les centres choisis.

Nous présentons dans les tableaux suivants les résultats de l'apprentissage de chaque rapport de coefficient ainsi que la performance d'estimation sur le dérapage en configuration lisse. Rappelons que l'apprentissage est réalisé sur un tirage aléatoire de 40% des données générées puis vérifié sur l'ensemble des données. Pour les deux solutions de placement de centres, nous regardons l'effet de l'utilisation de la norme  $\|\cdot\|_2$  et de la norme  $\|\cdot\|_\infty$ .

	Maillage régulier		K-moyennes	
	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$
$Cy_\beta^{-1}$	0,7%	0,71%	0,74%	0,98%
$\widetilde{C}y_p$	1,21%	1,4%	1,5%	2,12%
$\widetilde{C}y_{r,25\%}$	0,96%	1,05%	1,12%	1,45%
$\Delta \widetilde{C}y_{\delta p}$	10,55%	12,66%	11,1%	12,9%
$\Delta \widetilde{C}y_{\delta r}$	3,58%	7,02%	3,45%	5,37%
$\widehat{\beta}$	4,2%	4,6%	4,2%	4,4%

TAB. 7.3: Apprentissage direct des coefficients – Erreur relative

De manière générale, nous pouvons conclure que l'apprentissage est satisfaisant dans la mesure où la généralisation est bonne, que ce soit sur les valeurs des coefficients ou sur le dérapage. Ensuite, les performances entre les différentes structures de MEX sont équivalentes. Nous pouvons quand même conclure que la combinaison maillage régulier/norme  $\|\cdot\|_2$  semble la meilleure. Nous présentons également en figure 7.7 la répartition des erreurs pour chacun des rapports de coefficients et figure 7.6 celle des erreurs en dérapage pour la structure de MEX maillage régulier/norme  $\|\cdot\|_2$ .

Afin de ne pas délaissier les configurations hyper-sustentées, nous présentons la répartition des erreurs sur les coefficients et le dérapage pour la configuration « 1 ».

### 7.2.1.3 Régularisation

Nous avons ici abordé la régularisation des MEX. Pour cela, nous avons appliqué l'heuristique donné au paragraphe §5.4.2 à chacun des MEX afin de déterminer un paramètre

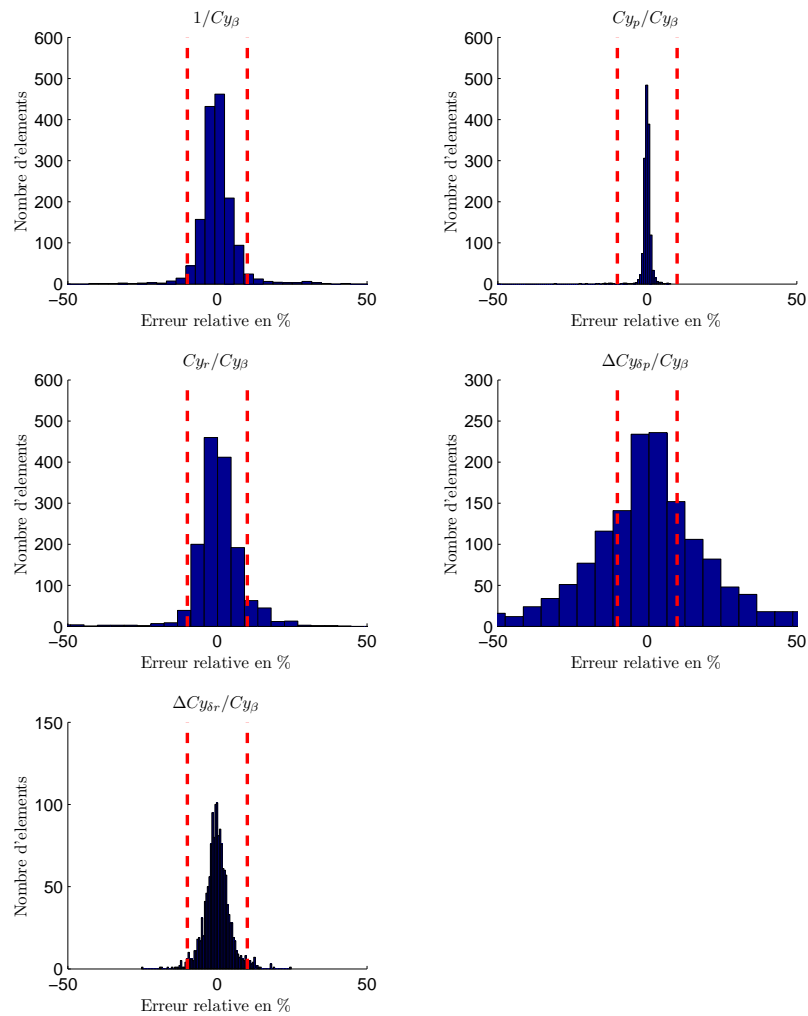


FIG. 7.6: Répartition des erreurs de validation (configuration lisse)

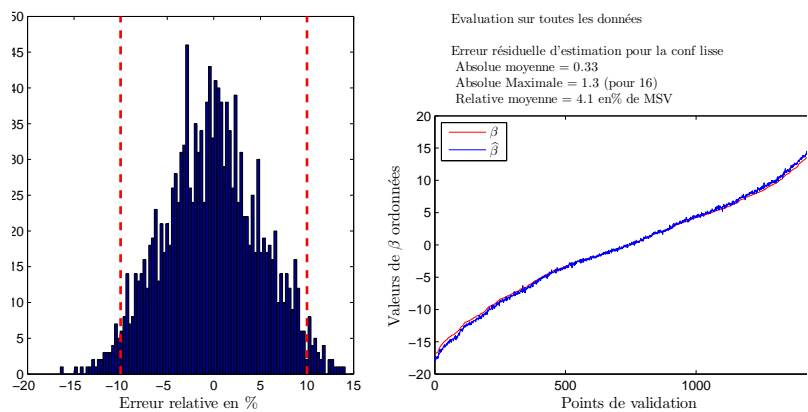


FIG. 7.7: Performance d'estimation du dérapage (configuration lisse)

$\lambda$  optimal. La régularisation devrait nous permettre d'améliorer non pas la performance d'apprentissage mais la performance de validation. Nous comparons dans le tableau suivant, les résultats régularisés à ceux obtenus au paragraphe précédent pour une structure



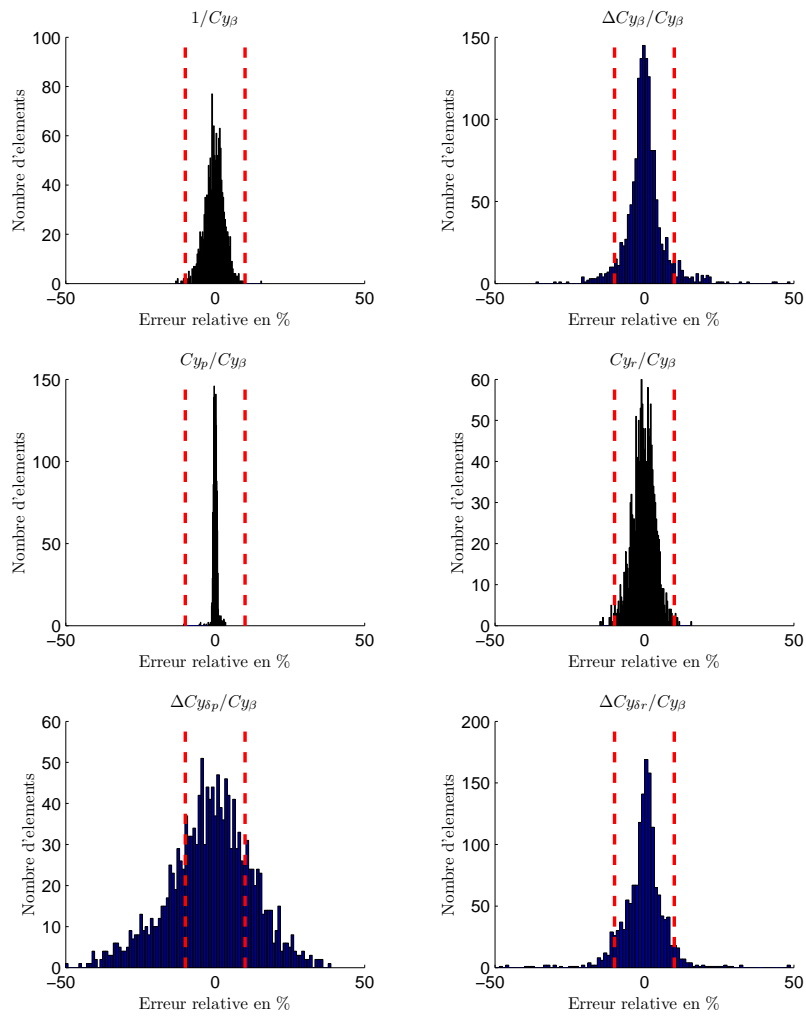


FIG. 7.8: Répartition des erreurs de validation (configuration « 1 »)

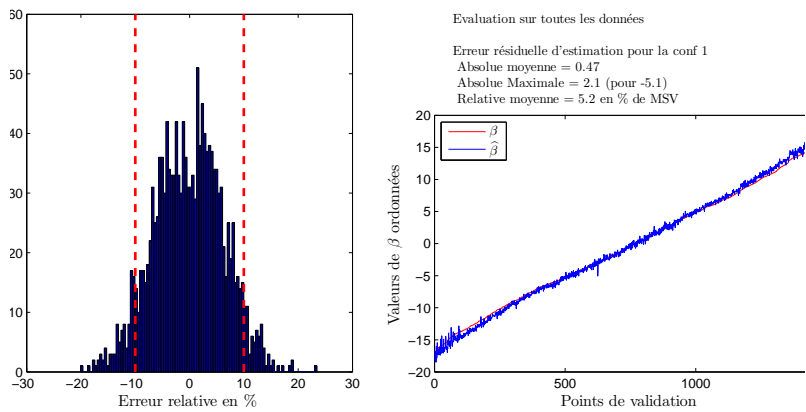


FIG. 7.9: Performance d'estimation du dérapage (configuration « 1 »)

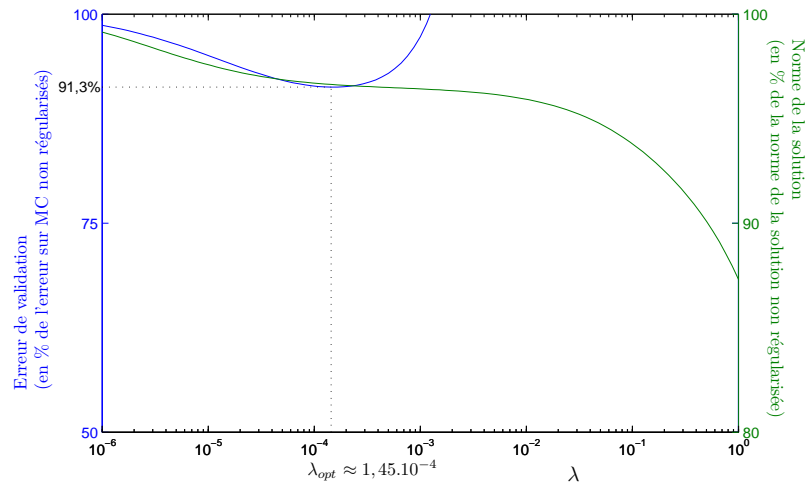
de MEX maillage régulier/norme  $\|\cdot\|_2$  et nous donnons la valeur de  $\lambda$  calculée.

La figure 7.10 montre un exemple de détermination de  $\lambda$  pour le coefficient  $\Delta\widetilde{C}y_{\delta\rho}$ .

Nous pouvons voir que la régularisation n'a pas un effet vraiment significatif dans ce

	Non-régularisé	Régularisé	$\lambda_{opt}$
$Cy_{\beta}^{-1}$	0,7%	0,69%	$3,8 \cdot 10^{-4}$
$\widetilde{C}y_p$	1,21%	1,18%	$4,2 \cdot 10^{-5}$
$\widetilde{C}y_{r,25\%}$	0,96%	0,96%	×
$\Delta \widetilde{C}y_{\delta p}$	10,55%	9,83%	$1,45 \cdot 10^{-4}$
$\Delta \widetilde{C}y_{\delta r}$	3,58%	3,61%	$4,8 \cdot 10^{-5}$
$\widehat{\beta}$	4,2%	4,2%	N/A

TAB. 7.4: Apprentissages non-régularisé et régularisé – Erreur relative

FIG. 7.10: Détermination du  $\lambda_{opt}$  pour  $\Delta \widetilde{C}y_{\delta p}$ 

cas. Si nous pouvons constater une légère amélioration sur la performance individuelle de chaque MEX, la performance globale d'estimation n'est pas modifiée. Nous choisirons donc de ne pas régulariser les MEX par la suite.

#### 7.2.1.4 Apprentissage direct sur le dérapage

Dans la perspective d'effectuer l'apprentissage en ligne de l'estimateur, nous avons regardé la performance d'un apprentissage indirect des MEX : le critère d'apprentissage n'est plus l'erreur de sortie des réseaux mais l'erreur entre le dérapage estimé et le vrai dérapage. En effet, nous considérons que nous n'avons plus accès aux valeurs des coefficients mais seulement à la mesure du dérapage. L'équation (7.8) nous donne une expression linéaire du dérapage<sup>2</sup>, par rapport aux quotients de coefficients aérodynamiques (entre crochets). Il suffit pour cela de remplacer le  $Cy$  par la décomposition proposée par (7.4).

Afin de faciliter la lecture, nous introduisons quelques notations. Soit  $N_o$  le nombre de rapports de coefficients inconnus,  $\widehat{\theta}_n$  le  $n^{\text{ème}}$  rapport de coefficient normalisé avec  $n =$

<sup>2</sup>Cela simplifie considérablement l'expression de  $\widehat{\beta}$  (7.11) l'optimisation du procédé lancé pour identifier les paramètres inconnus

$1, \dots, N_o$  et  $v_n$  sa mesure auxiliaire rattachée. Nous pouvons donc écrire :

$$\hat{\beta} = \sum_{n=1}^{N_o} v_n \cdot \hat{\theta}_n \quad (7.11)$$

Le  $n^{\text{ème}}$  coefficient est modélisé par un réseau RBF avec  $R_n$  estimateurs locaux linéaires  $\hat{f}_{r,n}(\mathbf{a}_n)$  où  $\mathbf{a}_n$  est le vecteur de dépendances de ce coefficient, identifiées au paragraphe §7.1.2.

D'après (2.52), le dérapage estimé peut s'écrire :

$$\begin{aligned} \hat{\beta} &= \sum_{n=1}^{N_o} v_n \psi_n^T \mathbf{x}_n \\ &= \sum_{n=1}^{N_o} \zeta_n^T \mathbf{x}_n \\ &= \zeta^T \mathbf{X} \end{aligned} \quad (7.12)$$

Nous pouvons donc conclure que l'apprentissage indirect des MEX se résume à problème aux moindres carrés linéaires. Nous avons testé cet apprentissage sur des MEX à maillages réguliers pour les normes  $\|\cdot\|_2$  et  $\|\cdot\|_\infty$ . La régularisation a aussi été étudiée. Les résultats sont résumés dans le tableau suivant.

	Non régularisé	Régularisé	$\lambda_{opt}$
$\ \cdot\ _2$	13,2%	8,2%	0,018
$\ \cdot\ _\infty$	12,1%	8,7%	0,0046

TAB. 7.5: Résultats sur l'apprentissage en  $\beta$

Les figures 7.11 et 7.12 montrent le calcul des  $\lambda_{opt}$  dans chacun des cas.

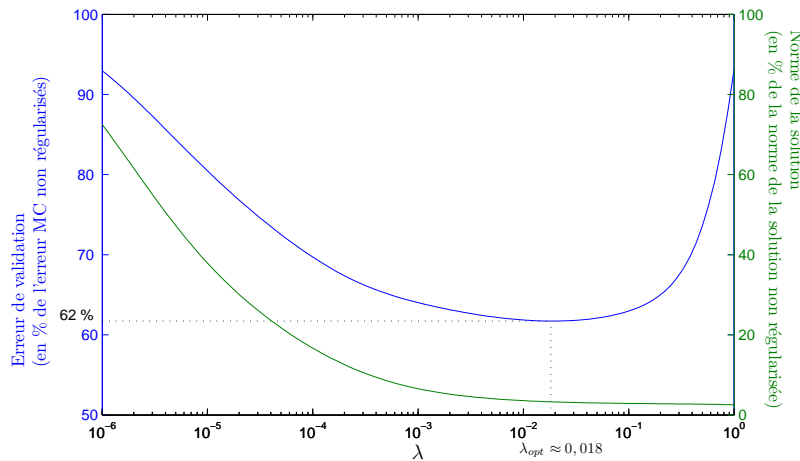


FIG. 7.11: Calcul de  $\lambda_{opt}$  pour la norme  $\|\cdot\|_2$

Nous pouvons voir que contrairement au paragraphe précédent, la régularisation a un effet significatif sur la performance de validation. Toutefois, nous constatons que la performance de validation est moins bonne que pour un apprentissage direct.

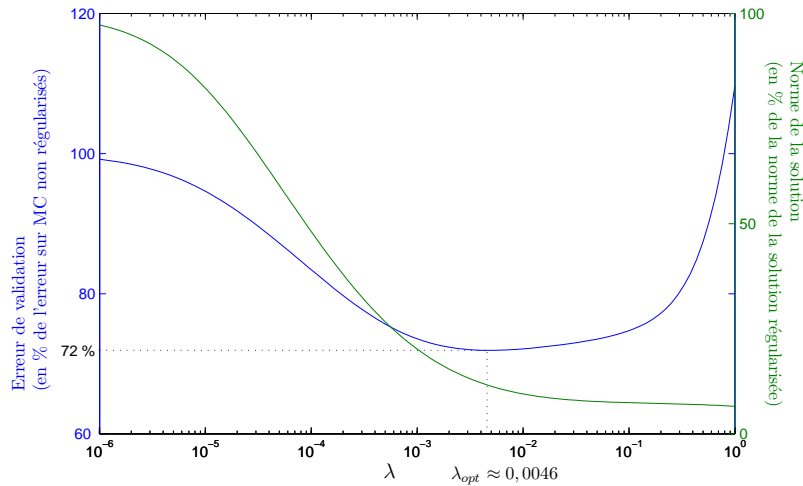


FIG. 7.12: Calcul de  $\lambda_{opt}$  pour la norme  $\|\cdot\|_{\infty}$

### 7.2.1.5 Conclusion

Cette partie a été l'occasion d'observer sur une application concrète l'influence des divers paramètres constituant un MEX (ou un ensemble de MEX) sur la performance d'apprentissage. Nous avons donc conclu pour cette application, qu'un maillage régulier de l'espace d'entrée associé à un calcul de distance par la norme  $\|\cdot\|_2$ , sur critère d'apprentissage non-régularisé offrait les meilleures performances d'apprentissage et de généralisation. A complexité équivalente, le positionnement des centres par algorithme des  $k$ -moyennes offre des performances similaires.

## 7.2.2 Apprentissage en ligne

Nous allons à présent nous intéresser au recalage d'un estimateur ayant fait l'objet d'un apprentissage direct hors-ligne. Comme nous l'avons annoncé au paragraphe §7.1.4, nous allons utiliser des données d'essais en vol. A ce titre, nous devrons effectuer l'apprentissage sur l'erreur en dérapage car les coefficients aérodynamiques ne sont plus mesurables. Dans un premier temps, nous allons donner un aperçu de la localisation des données utilisées dans les espaces d'entrée des MEX. Ensuite, nous allons appliquer les méthodes présentées au paragraphe §4.2 sur un estimateur dont les MEX ont un maillage régulier et utilisent la norme  $\|\cdot\|_2$ . Nous comparerons la performance de cet estimateur à celle d'un estimateur implémenté dans les calculateurs de bord au moment des essais. Enfin, nous vérifierons la performance d'un estimateur à maillage régulier, après un parcours d'apprentissage sur tous les essais en vérifiant la localité de l'apprentissage. Nous n'avons considéré ici que les essais en configuration lisse afin d'alléger le propos. La démarche suivie peut s'appliquer de la même manière aux estimateurs utilisés pour les autres configurations aérodynamiques.

## 7.2.3 Localisation des essais

Nous donnons à la figure 7.13, la localisation de ces essais dans le domaine  $(\alpha, \text{Mach})$  normalisé car il s'agit d'entrées communes à tous les MEX. Nous représentons également le maillage réalisé pour ce domaine.

Les symboles situent le point de vol et les ellipses délimitent la zone explorée pendant

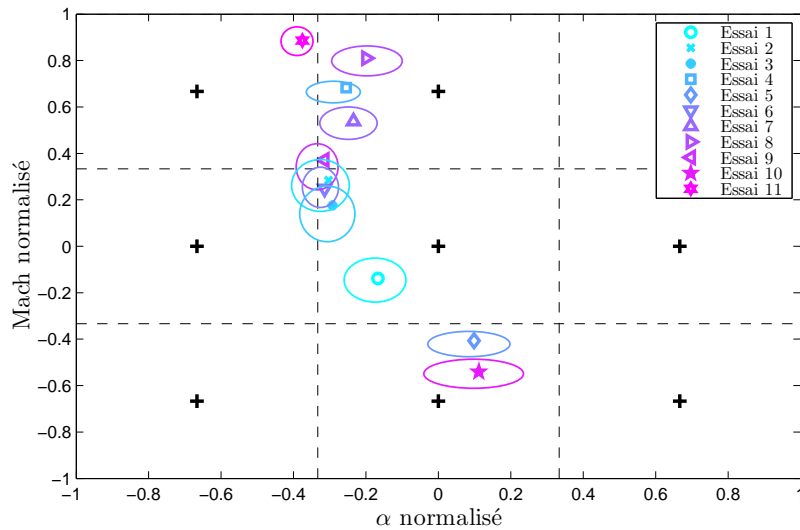


FIG. 7.13: Localisation des essais utilisés dans l'espace  $(\alpha, Mach)$  normalisé

l'essai. Nous pouvons voir que le domaine n'est pas entièrement recouvert. D'une part, les zones de haut Mach et fortes incidences ne seront jamais explorées et d'autre part, les zones à faible Mach sont davantage couvertes avec les configurations hyper-sustentées.

Pour évaluer la performance des méthodes, nous n'allons considérer dans un premier temps que deux points de vol suffisamment distants. Nous pourrions ainsi valider le caractère local de l'apprentissage.

	Mach	$z_p$ (ft)	Centrage	Masse (t)
Essai 10	0.37	9 200	38%	260
Essai 11	0.85	30 000	30%	256,4

TAB. 7.6: Points de vol

## 7.2.4 Résultats d'apprentissage

### 7.2.4.1 Recalage sur un seul point

Nous commençons par observer la performance initiale de l'estimateur sur l'essai n° 10 puis sa performance après apprentissage à l'aide des MCR. Nous allons également comparer la performance de l'estimateur existant à celle de notre estimateur. Ils se distinguent par :

- le calcul du dérapage : la modélisation du  $Cy$  sur laquelle est basée l'ancien estimateur est moins complexe que celle donnée par (7.4). Les effets non-linéaires de  $\Delta Cy_\beta$  ne sont pas pris en compte ni l'effet de traînée de captation (cf. §6.2.3).
- la modélisation des coefficients aérodynamiques : alors que nous avons utilisé des MEX à 7 entrées, l'autre estimateur utilise des tables d'interpolation linéaire, limitées à deux entrées.

Nous avons initialisé la matrice de covariance  $\Pi(0)$  des MCR avec un gain relativement faible ( $\Pi(0) = 10 \cdot I$ ) car le vecteur de paramètres initial  $\mathbf{X}(0)$  n'est pas quelconque. En effet, dans la mesure où nous récupérons un estimateur ayant effectué un apprentissage hors ligne, nous pouvons considérer que cette valeur du vecteur paramètres est proche d'une

valeur optimale. Les résultats sont présentés aux figures 7.14 et 7.15 où la courbe bleue représente le dérapage mesuré ramené au centre de gravité, la ligne magenta en pointillés le dérapage estimé par l'estimateur embarqué, les courbes rouge et verte le dérapage estimé par l'estimateur à base de MEX avant et après l'apprentissage.

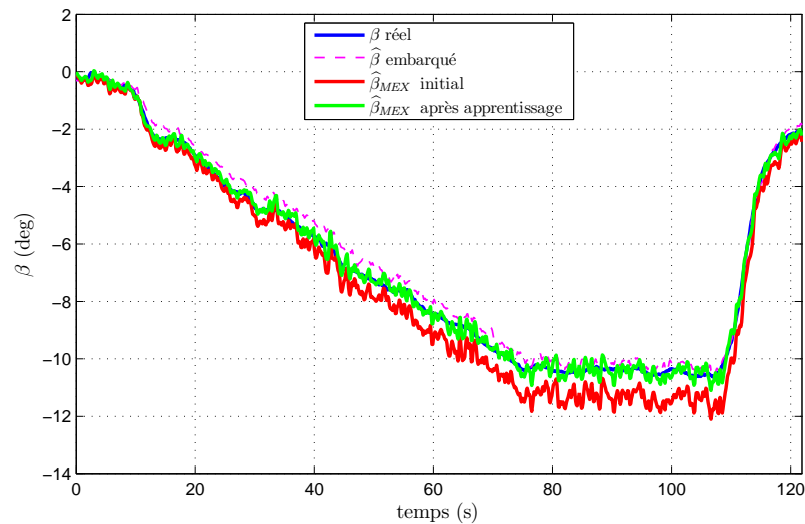


FIG. 7.14: Essai 10 - Estimateur avant et après apprentissage

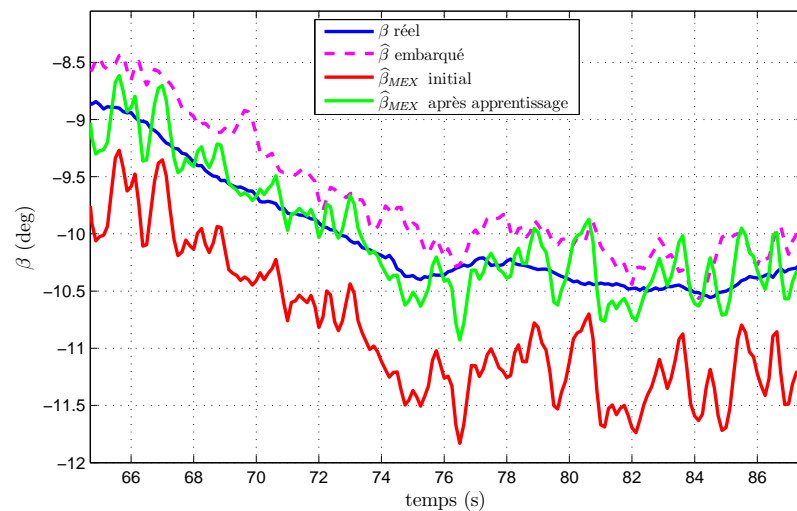


FIG. 7.15: Essai 10 - Estimateur avant et après apprentissage (détail)

Remarquons que pour l'estimateur non recalé par MCR, des erreurs apparaissent, dues à la fois à l'imperfection de l'apprentissage hors ligne et au fait que les valeurs des ensembles d'apprentissage sont inexacts.

De plus, les MCR améliorent sensiblement la performance locale bien que l'estimation soit bruitée, à cause du signal provenant de la sonde de  $Ny$  et de la pseudo-dérivation des taux de roulis  $p$  et de lacet  $r$ .

Les résultats sont satisfaisants dans la mesure où notre estimation est meilleure que celle de l'ancien estimateur, malgré la présence d'un comportement oscillatoire. Cependant, les données que nous utilisons ont été lissées lors du traitement effectué sur les enregistrements d'essais, ce qui explique la propreté du dérapage estimé par l'estimateur embarqué comparé à celui de notre estimateur.

### 7.2.4.2 Généralisation

Nous allons à présent vérifier la performance de généralisation et le caractère local de la retouche. Pour cela, nous allons commencer par une évaluation simple : nous allons régler l'estimateur issu de l'apprentissage hors-ligne successivement sur deux points de vol suffisamment distants dans l'espace d'entrée puis nous allons vérifier si la performance sur le premier point de vol est conservée après l'adaptation sur le deuxième point de vol.

- Les figures 7.16 et 7.17 montrent l'estimation du dérapage sur l'essai 11 après avoir réglé l'estimateur sur l'essai 10. Nous avons également initialisé la matrice de covariance avec la dernière valeur calculée lors de l'apprentissage sur l'essai 10. Nous conservons ainsi l'information de l'apprentissage précédent.

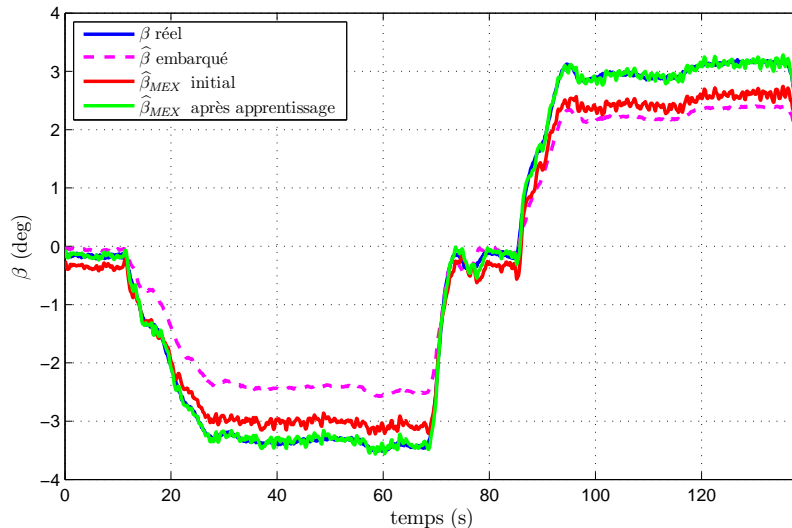


FIG. 7.16: Essai 11 - Estimateur avant et après apprentissage

Nous pouvons voir qu'après l'apprentissage par MCR, la performance locale (en vert) a bien été améliorée par rapport à la performance initiale (en rouge). Encore une fois, l'estimateur à base de MEX est meilleur que l'estimateur embarqué (en magenta).

- La figure 7.18 montre la différence en degrés entre la réponse de l'estimateur sur l'essai 10 avant et après l'adaptation sur l'essai 11. Nous pouvons voir que l'écart est très faible et par conséquent, que l'apprentissage s'est bien fait de façon locale.

Nous pouvons résumer les résultats précédents dans le tableau 7.7. Nous donnons dans ce tableau le critère donné en (7.9), à savoir la norme de l'erreur entre la mesure et l'estimée, ramenée sur la norme de la mesure.

Nous avons également testé les MCR avec oubli directionnel (pour  $\lambda = 0,997$ ) avec une même initialisation de la matrice  $\Pi$ . Cependant, les résultats sont quasiment identiques.

Nous avons ensuite vérifié ce comportement sur un ensemble de onze enregistrements d'essais en vol. La figure 7.19 se lit de la manière suivante : nous commençons par observer

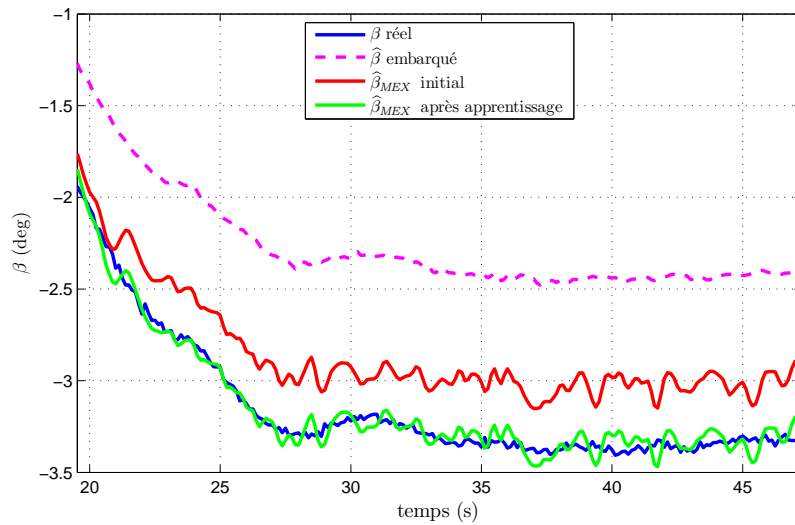


FIG. 7.17: Essai 11 - Estimateur avant et après apprentissage (détail)

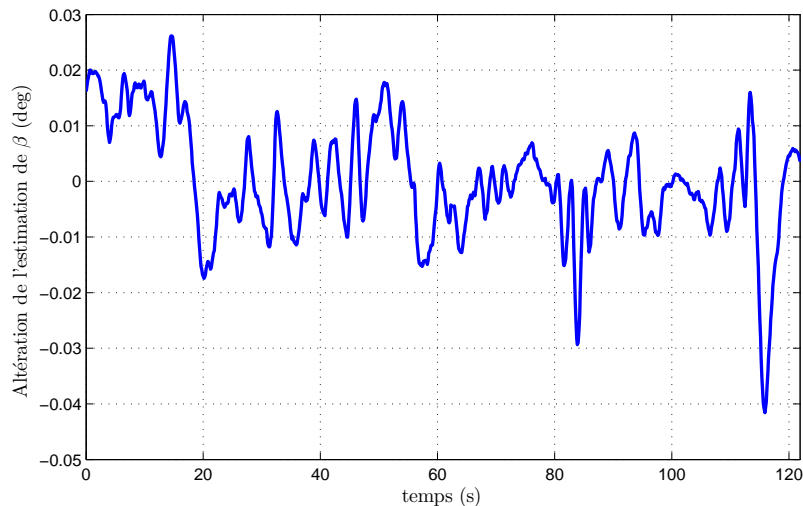


FIG. 7.18: Désadaptation de l'estimateur sur l'essai 10 après apprentissages successifs

la performance de l'estimateur issu de l'apprentissage hors-ligne (Essais appris = 0) puis nous faisons apprendre sur l'essai n° 1 et nous vérifions l'estimateur sur les dix autres essais ; puis nous faisons apprendre l'essai n° 2 et nous vérifions également sur les dix autres essais, et ainsi de suite. A chaque apprentissage, nous initialisons la matrice de covariance avec la dernière valeur calculée lors de l'apprentissage sur l'essai précédent.

Les cercles noirs représentent l'erreur initiale pour chaque essai. Les cercles rouges et verts ont un diamètre proportionnel au rapport entre l'erreur observée après apprentissage sur un essai donné et l'erreur initial : si un de ces cercles est à l'intérieur du cercle noir, la performance a été améliorée et inversement. Les cercles rouges mettent en évidence les essais qui n'ont pas été appris et les cercles verts ceux qui ont été appris.

Nous pouvons voir que l'apprentissage s'effectue bel et bien de manière locale. En effet, lors de l'apprentissage de l'estimateur sur un des essais, nous voyons qu'à la fois les essais appris et non appris sont peu ou pas modifiés. De plus dans le cas d'une modification, elle



	Erreur
Evaluation initiale sur essai 10	8,97%
<i>Apprentissage sur essai 10</i>	
Evaluation finale sur essai 10	2,54%
Evaluation initiale sur essai 11	14,35%
<i>Apprentissage sur essai 11</i>	
Evaluation finale sur essai 11	2,47%
Vérification sur essai 10	2,54%

TAB. 7.7: Première vérification de la localité de l'apprentissage

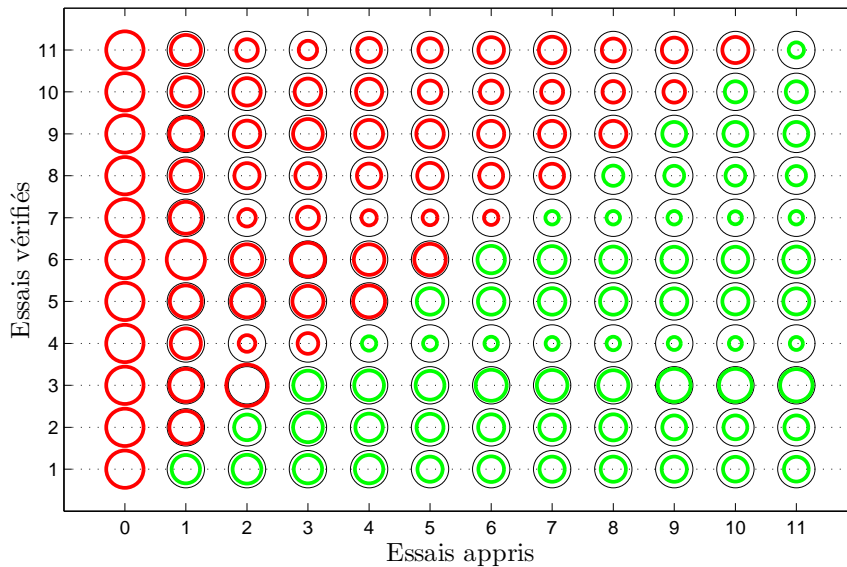


FIG. 7.19: Vérification sur l'ensemble des enregistrements, par apprentissages successifs

est en générale « dans le bon sens », à savoir une réduction de l'erreur. Ces interactions peuvent s'expliquer par la proximité des essais dans l'espace d'entrée de MEX (cf. figure 7.13).

### 7.3 Conclusion

Nous avons pu voir sur une application assez simple, la mise en pratique d'une méthode de recalage locale, que ce soit hors ligne ou bien en temps-réel. Si les résultats sont prometteurs, il reste que la solution proposée présente une certaine complexité d'implémentation, assez peu compatible avec les possibilités d'un ordinateur embarqué. Ce problème pourra faire l'objet d'un travail ultérieur.

# Chapitre 8

## Compensation de spoilers

### Sommaire

---

<b>8.1 De l'utilisation des spoilers . . . . .</b>	<b>125</b>
8.1.1 But de la compensation . . . . .	128
<b>8.2 Expression du besoin . . . . .</b>	<b>128</b>
8.2.1 Objectifs et contraintes . . . . .	129
8.2.2 Comportement de l'avion . . . . .	129
8.2.3 Conception de la compensation . . . . .	130
8.2.4 Conclusion . . . . .	131
<b>8.3 Génération de données . . . . .</b>	<b>132</b>
8.3.1 Hors-ligne . . . . .	132
8.3.2 En-ligne . . . . .	140
<b>8.4 Apprentissage . . . . .</b>	<b>142</b>
8.4.1 Hors ligne . . . . .	142
8.4.2 En ligne . . . . .	145
<b>8.5 Conclusion . . . . .</b>	<b>148</b>

---

CETTE partie a pour but de présenter les utilisations des surfaces extradors d'un avion de transport civil type Airbus ainsi que la fonction de compensation du moment de tangage généré par leur extension ou rétraction. Nous donnons également les objectifs et contraintes liés à leur réglage, qui vont nous servir de cahier des charges par la suite. Il s'agit d'élaborer une loi de compensation en boucle ouverte. Nous allons donc suivre la démarche présentée à la partie II. La partie hors ligne consistera en l'identification d'une structure de loi à partir d'une inversion numérique, suivant le cahier des charges donné. Cette partie sera effectuée sur un modèle de *synthèse* de l'avion. Une fois la synthèse effectuée sur le domaine de vol de l'avion, nous interpolerons les paramètres de la loi à l'aide d'un MEX. Nous irons ensuite vérifier et éventuellement ajuster la loi en temps-réel sur un modèle de *validation* de l'avion, qui est en fait le modèle de synthèse dont les coefficients aérodynamiques ont été recalés d'après les modifications observées après identification de l'avion.

### 8.1 De l'utilisation des spoilers

Sur les Airbus, la fonction aérofreinage est assurée par les spoilers, qui sont les surfaces situées sur l'extrados des ailes de l'appareil, comme indiqué en figure 8.1. Toutefois, ce

n'est pas l'unique solution et c'est à l'avionneur de choisir leur emplacement en fonction de ses contraintes.



FIG. 8.1: Spoilers A380

Lorsqu'elles sont déployées, ces surfaces génèrent deux effets aérodynamiques directs : une augmentation de la traînée et une perte locale de portance par une dégradation du profil de l'aile, d'où le terme « spoiler » issu de « spoil » (abîmer, gâcher). Le phénomène est illustré à la figure 8.2, sur laquelle est représentée la répartition (approximative) de la portance lorsqu'un spoiler est rentré ou déployé.

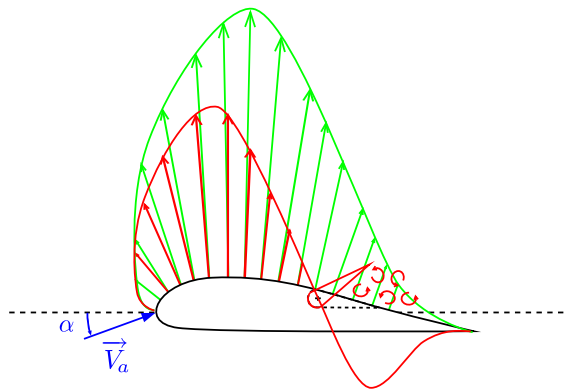
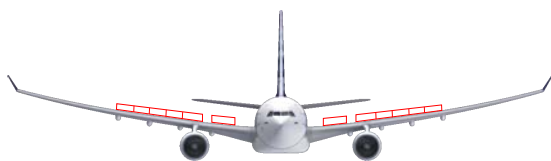
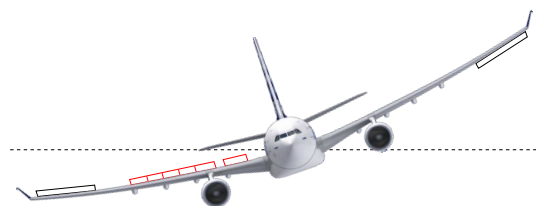


FIG. 8.2: Répartition de la portance avec et sans braquage de spoilers

Ils sont utilisés de différentes manières :

- *aérofreins* (A/F) : les surfaces sont déployées symétriquement en vol (cf. figure 8.3) et servent à ralentir l'avion en augmentant la traînée aérodynamique. Il s'agit d'une commande de décélération. Par extension, les spoilers permettent donc d'augmenter la pente de descente de l'avion tout en gardant sa vitesse constante.
- *roulis* : le braquage est alors dissymétrique (cf. figure 8.4). Cela permet de restaurer de l'autorité en roulis lorsque les ailerons arrivent à leurs limites, par l'intermédiaire d'un répartiteur d'ordre.
- « *ground spoilers* » : les spoilers sont alors braqués au maximum dès que l'avion touche

FIG. 8.3: *Fonction A/F*FIG. 8.4: *Fonction roulis*

le sol à l'atterrissage (cf. figure 8.5). Ils servent avant tout à plaquer l'avion au sol par destruction de la portance. La traînée aérodynamique engendrée ne représente qu'un faible pourcentage de la force de freinage globale, assurée par les freins de roues et les inverseurs de poussée.

FIG. 8.5: *Fonction ground spoilers*

- *PTA* (Passive Turbulence load Alleviation : allègement passif de charges en turbulence) : le braquage est symétrique. Le but est ici d'alléger les charges sur la voilure, dues aux turbulences.
- *DLC* (Direct Lift Control : contrôle direct de portance) : les spoilers sont utilisés de façon symétrique autour d'un braquage « d'équilibre » et vont venir contrôler directement la portance de l'avion. C'est une fonction qui permet un meilleur contrôle de l'avion pendant les phases d'approche et fait encore l'objet de recherches.

Du point de vue des lois de pilotage, ces surfaces sont réparties en trois groupements : un groupement interne (localisés à l'emplanture de l'aile) et deux groupements externes. Les raisons de cette subdivision sont multiples. Tout d'abord, il a fallu isoler les spoilers internes car ils génèrent des perturbations sur le plan horizontal réglable. Il n'est donc pas possible d'affecter les mêmes déflexions qu'aux autres spoilers. L'autre raison est liée à l'architecture du système de commandes de vol électrique. Il est à noter que les futurs programmes s'orientent vers un contrôle spécifique à chaque surface.

De façon opérationnelle, les A/F sont utilisés par les pilotes pour augmenter la vitesse verticale de descente de l'avion. Leur utilisation n'est pas systématique et dépend du trafic aérien alentour. Par rapport au plan de vol, la première utilisation rencontrée correspond au tout début de la descente et est située entre le niveau de croisière (40000 ft) et une altitude de 25000 ft, et un Mach compris entre sa valeur de croisière (0,86 ou 0,89 selon les avions) et 0,65. Cependant, leur utilisation à ce moment est évitée pour des raisons de sécurité et de confort des passagers liées au *buffeting*<sup>1</sup>. Ils sont davantage utilisés dans la zone intermédiaire de la descente, entre 25000 ft et 10000 ft. A des altitudes inférieures, les pilotes disposent de moyens plus efficaces de ralentir l'avion (trains, volets). Enfin, les A/F sont utilisés la plupart du temps en position intermédiaire du levier de commande.

### 8.1.1 But de la compensation

Le but de cette fonction est de compenser le moment de tangage induit par le braquage des spoilers, à l'aide de la gouverne de profondeur (cf. figure 8.6). Du fait de leur position par rapport au centre de gravité, la perte locale de portance  $\Delta L_{sp}$  induite par la déflexion des spoilers va générer un moment de tangage  $\Delta C m_{sp} > 0$  à cabrer (le nez de l'avion part vers le haut)<sup>2</sup>. Une déflexion  $\delta q_{comp}$  sur la gouverne de profondeur est alors appliquée de façon à générer un moment de tangage à piquer  $C m_{\delta q} \delta q_{comp} < 0$  qui compense celui généré par les spoilers.

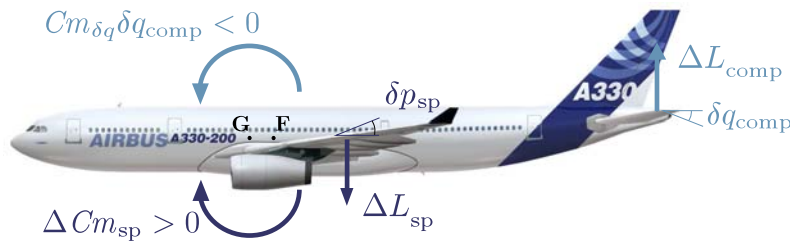


FIG. 8.6: Illustration du phénomène à compenser

Cette fonction est active autant sur un braquage symétrique des spoilers (A/F) que sur un braquage asymétrique (roulis). Elle compense également les braquages symétriques des ailerons lorsqu'ils sont utilisés pour alléger les charges en bout d'ailes dues aux manœuvres.

Il est à noter qu'il est possible de faire deux compensations séparées pour les braquages symétriques et dissymétriques.

## 8.2 Expression du besoin

Nous donnons dans ce paragraphe un cahier des charges industriel ainsi qu'un exemple de mise au point de la compensation.

<sup>1</sup> Il s'agit du tremblement de l'avion susceptible d'apparaître lors de manœuvres à la vitesse de croisière. Selon l'incidence, l'écoulement peut comporter des décollements, qui constituent une excitation aérodynamique. La structure de l'avion, du fait de cette excitation, tremble, et ce tremblement met en danger la stabilité du vol.

<sup>2</sup> Il se peut qu'individuellement, les spoilers internes génèrent un moment contraire mais globalement, l'effet est cabreur.

### 8.2.1 Objectifs et contraintes

En fonction A/F, la compensation doit aider la loi engagée à minimiser les variations de la trajectoire à court terme. De façon opérationnelle, les A/F sont utilisés pendant la descente de l'avion. Les pilotes ont tendance à l'utiliser afin d'augmenter la pente  $\gamma$  tout en gardant une vitesse longitudinale constante. Aussi, un comportement cabreur sera à éviter sur une extension d'A/F et il faudra chercher une compensation légèrement piqueuse. La compensation devra également minimiser la vitesse verticale  $V_z$  induite (le critère est de  $\pm 200$  ft/mn, correspondant à une perte maximale d'altitude de 10 ft en PA, en mode maintien d'altitude.)

Sur une mise en roulis brusque, la compensation doit minimiser les écarts d'assiette longitudinale  $\theta$ .

Enfin, à la sortie des déporteurs au sol, la compensation devra contrer le moment cabreur des spoilers.

La principale contrainte est de faire simple, c'est-à-dire une compensation proportionnelle au braquage commandé à chaque groupe de gouverne mais avec néanmoins une prise en compte d'éventuelles pannes hydrauliques.

Cette compensation est ajoutée à l'ordre gouverne calculé par la loi  $Nz$  et est limitée à  $7,5^\circ$  en loi normale<sup>3</sup> et en loi alternée<sup>4</sup>. En loi directe<sup>5</sup>, cette limitation est supprimée.

### 8.2.2 Comportement de l'avion

Nous observons dans cette partie, les réponses d'un avion, manche au neutre, au cours de deux phases d'utilisation standard des aérofreins : en approche ( $\gamma_0 = -3^\circ$ ) à vitesse modérée et basse altitude (220 kts, 5000 ft, configuration « 1 », 180 t et centrage à 25%) et en croisière ( $\gamma_0 = 0^\circ$ ) à haute vitesse et haute altitude (Mach 0.86, 35000 ft, configuration lisse, 220 t et centrage à 30%). Les réponses considérées sont la pente  $\gamma$ , l'assiette longitudinale  $\theta$ , la vitesse longitudinale  $V_c$  et le facteur de charge vertical  $Nz$ .

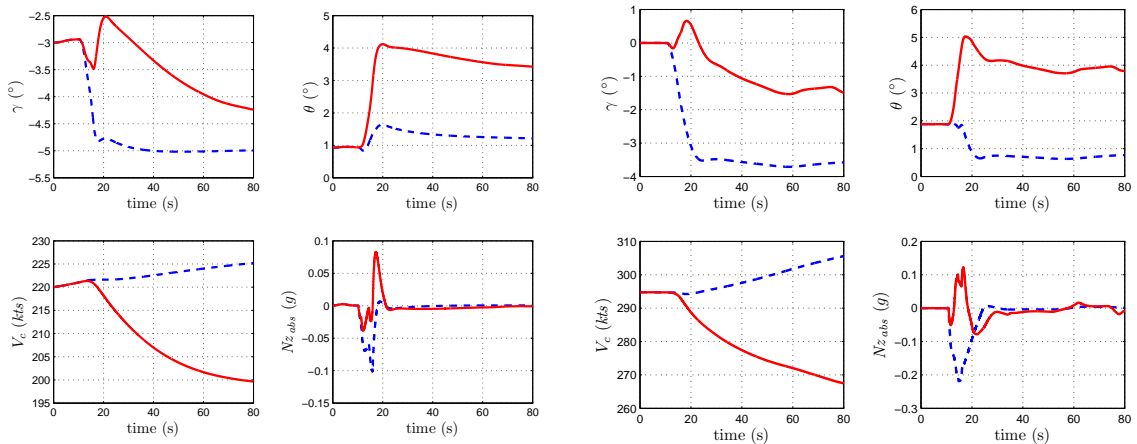


FIG. 8.7: Comportement type en réponse à une sortie full A/F en approche (vitesse modérée et basse altitude)

FIG. 8.8: Comportement type en réponse à une sortie full A/F en croisière (vitesse élevée et haute altitude)

<sup>3</sup>Il s'agit de la loi de pilotage utilisée en fonctionnement nominal

<sup>4</sup>Il s'agit d'une loi dégradée suite à un certain nombre de pannes

<sup>5</sup>Il s'agit d'une loi boucle ouverte, utilisée si de nombreux systèmes sont en panne

Les courbes en pointillés bleus représentent le comportement de l'avion sans la compensation et les courbes en trait plein rouge avec la compensation existante. Nous pouvons constater que l'objectif de la compensation pour cet avion a été de minimiser les écarts d'assiette  $\theta$  au détriment des écarts de pente et de vitesse verticale.

### 8.2.3 Conception de la compensation

Pour la suite, l'avion est considéré comme étant symétrique et les interactions entre les différents spoilers ne sont pas prises en compte.

Les dépendances principales, mises en évidence par de travaux effectués lors de précédents programmes, sont le centrage  $x_{cg}$ , la vitesse corrigée  $V_c$  (la définition est donnée au paragraphe §7.1.3), l'altitude  $z_p$  et la configuration aérodynamique de l'avion. Un autre paramètre sensible est le braquage du spoiler dont l'effet est non-linéaire. Remarquons que ces dépendances ne sont pas exactement celles des coefficients aérodynamiques associés (cf. paragraphe §6.3). En effet, ces derniers dépendent du Mach, et non de la  $V_c$ . Cependant, ces deux paramètres sont liés par une relation mathématique. De plus, cette tabulation a été conservée dans un souci de cohérence avec les autres lois de pilotage.

Nous noterons la dépendance linéaire du  $Cm$  en fonction du centrage. En effet, en première approche, nous pouvons considérer que le moment généré est égal à la perte locale de portance que multiplie le bras de levier, i.e. la distance du centre de gravité au point d'application de la force de déportance.

Le spoiler interne a un effet plutôt à piquer dans la plupart des cas de vol. Le spoiler le plus externe a quant à lui un effet cabreur quelles que soient les conditions de vol. L'effet des autres spoilers dépendra davantage du centrage de l'avion et de la configuration aérodynamique.

**Exemple type** Nous illustrons le propos de la conception avec un exemple type pour un avion Airbus. La figure 8.9 présente l'architecture de la compensation existante.

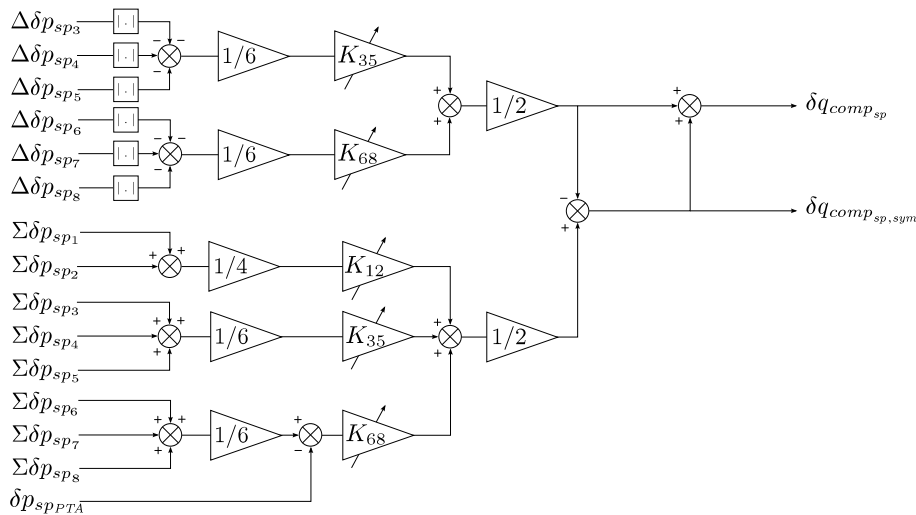


FIG. 8.9: Structure type de compensation

Comme il a été dit plus haut, la compensation doit avoir une structure simple, proportionnelle au braquage commandé des spoilers/ailerons. Les gains multiplicatifs sont alors

calculés dans un premier temps comme des rapports d'efficacité, représentant le braquage de gouverne de profondeur équivalent nécessaire pour compenser le moment de tangage induit par un braquage de 1° de spoiler/aileron. Nous pouvons les exprimer de la façon suivante :

$$K_i = \frac{\Delta C m_{\delta p_{sp_i}} / \delta p_{sp_i}}{C m_{\delta q}} \quad (8.1)$$

Ces coefficients sont multipliés par leur braquage commandé respectif puis sommés pour générer la compensation désirée. L'utilisation de tels coefficients donne un sens physique à la compensation ce qui permet un réglage plus aisé pendant la phase de test.

De façon à simplifier la structure, nous pouvons ignorer les effets de certains paramètres d'entrée. Tout d'abord, l'efficacité des spoilers ( $\Delta C m_{\delta p_{sp_i}} / \delta p_{sp_i}$ ) ne dépend pas de la déflexion de façon significative pour de petites déflexions. En d'autres termes, le moment de tangage est considéré comme proportionnel au braquage des spoilers.

En lisse<sup>6</sup>, le Mach a un effet sensible. En efficacité relative, des travaux réalisés lors des précédents développements, ont montré une dispersion fonction de l'altitude pour une vitesse donnée. Le centrage modifie également l'amplitude du moment de tangage induit par les spoilers. Par conséquent, la tabulation en vitesse et altitude est réalisée à centrage avant et des corrections additives sont apportées selon le centrage.

Lorsque l'avion est hyper-sustenté, le domaine de vol est restreint en altitude (< 20000 ft) et en vitesse. Nous ne tenons alors plus compte des effets du Mach donc la tabulation s'en trouve simplifiée : elle ne dépend plus que du centrage et de la configuration aérodynamique.

La compensation prend également en compte les pannes hydrauliques. L'amplitude de la compensation est alors ajustée en fonction du nombre de spoilers disponibles.

Au final, les gains s'écrivent de la façon suivante :

$$K_i = K_{fail} \left[ g_1(V_c, z_p) + f(x_{cg}) \right] \cdot \overline{\delta_{hl}} + g_2(x_{cg}, conf) \cdot \delta_{hl} \quad (8.2)$$

avec  $\delta_{hl}$  un booléen qui détermine si l'avion est hyper-sustenté.

#### 8.2.4 Conclusion

Forts de cette connaissance quant au phénomène à compenser et à la conception de la compensation, nous allons appliquer la méthodologie présentée à l'introduction de la partie II (p.31), afin de permettre une synthèse systématique de cette compensation. Pour cela, nous devons identifier les divers signaux que nous avons utilisés pour développer cette méthode. La figure 8.10 nous donne la structure de la loi de pilotage longitudinal, ainsi que l'intégration de la compensation au sein de cette loi.

Nous pouvons ainsi facilement faire le parallèle avec la figure II.4 (p. 34) où nous aurions comme signal de perturbation  $w$  l'ordre  $\delta p_{sp}$  envoyé aux spoilers, le signal  $\delta q_{comp}$  comme signal de commande  $u$  et les sorties considérées  $\mathbf{z}$  seraient la pente, la vitesse verticale et le facteur de charge vertical, soit :

$$\mathbf{z}(t) = (\gamma(t) \quad V_z(t) \quad Nz(t))^T \quad (8.3)$$

---

<sup>6</sup>Becs et volets rentrés



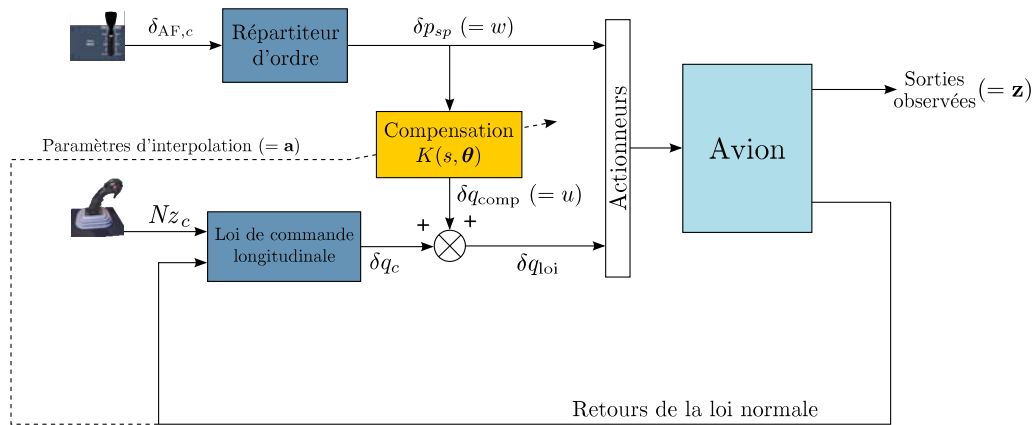


FIG. 8.10: Intégration de la compensation dans la loi longitudinale

### 8.3 Génération de données

Nous nous trouvons dans le cas où nous n'avons pas accès directement aux données d'apprentissage des MEX. Nous allons donc appliquer la méthode décrite à la partie II pour essayer d'améliorer la loi existante.

#### 8.3.1 Hors-ligne

Dans la pratique, cette partie de la méthode peut être réalisée en bureau d'étude à partir de données de simulations ou d'essais en vol. L'interprétation des résultats est donnée pour chacune des deux parties du réglage : l'inversion et la synthèse de la loi, sur plusieurs points de vol. La figure 8.11 résume les étapes de la synthèse hors-ligne de la loi.

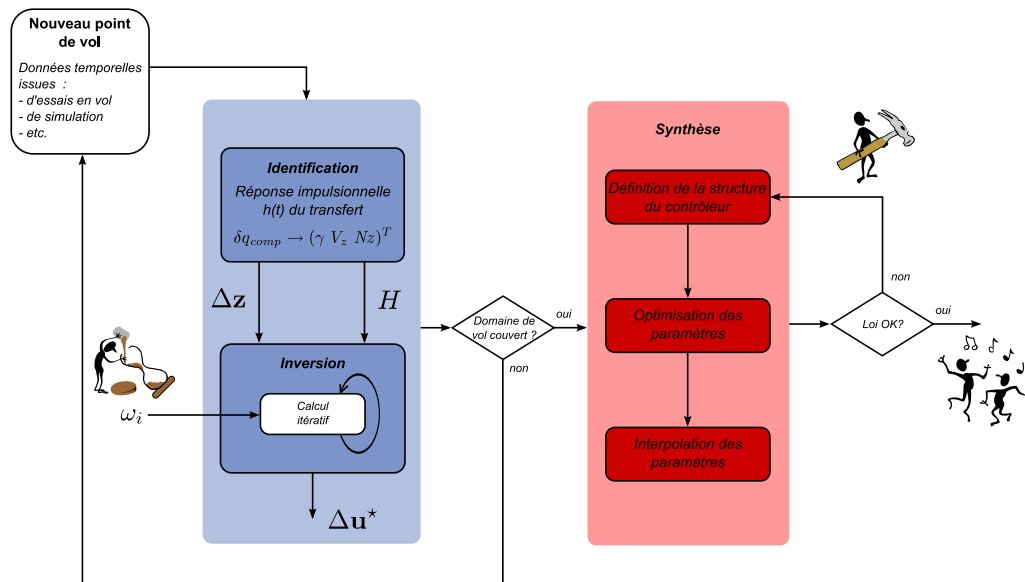


FIG. 8.11: Méthode de synthèse de loi par inversion

Tout d'abord, nous avons généré un certain nombre de cas de vol dans le domaine

$(V_c, z_p)$ . Toutefois, nous ne pouvons pas uniquement délimiter ce domaine entre deux bornes inférieure et supérieure car l'avion est limité de par sa conception, en vitesse, en altitude mais également en Mach, qui est une fonction non-linéaire de la  $V_c$  et de l'altitude  $z_p$ . Nous faisons apparaître cette limite en Mach sur un domaine  $(V_c, z_p)$  à la figure 8.12.

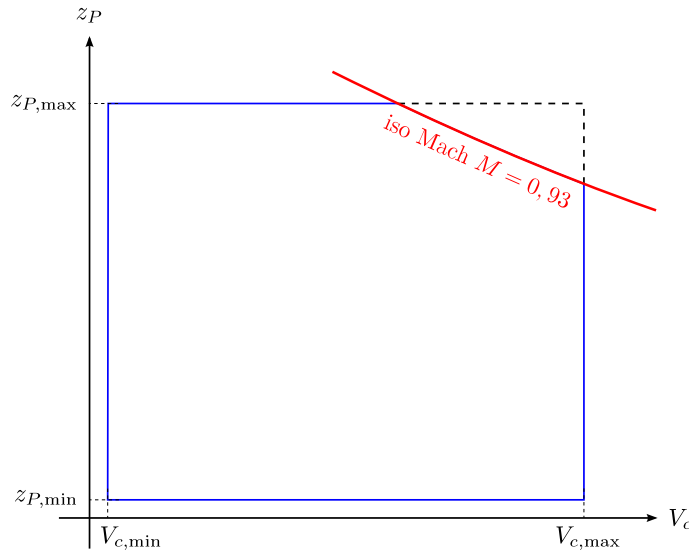


FIG. 8.12: Domaine  $(V_c, z_p)$  - Illustration de la limitation en Mach

Il faudra donc veiller à générer un nombre suffisant de points d'apprentissage sur cette courbe iso-Mach afin de réduire les problèmes aux limites du domaine.

### 8.3.1.1 Inversion

Pour l'inversion, nous allons étudier les performances de la méthode sur le domaine de vol ainsi que sa robustesse à d'éventuels bruits de mesure lors de l'acquisition des données. Tout d'abord, nous générons des points de vol (aléatoirement ou selon un maillage régulier) dans le domaine défini à la figure 8.12, tout en concentrant davantage de données dans les zones à haut Mach (entre 0,79 et 0,93) où nous avons vu au paragraphe §8.2.3, que les coefficients aérodynamiques considérés présentent de fortes variations.

Pour chaque point de vol, nous allons calculer le signal optimal  $\delta q_{\text{comp}}$  qui compense une extension des A/F. Le signal  $w_o$  correspondant à cette perturbation est une rampe saturée en amplitude. L'extension dure environ 5 secondes pour atteindre une amplitude égale aux trois-quarts de la déflexion maximum autorisée pour chaque groupe de spoilers.

Soulignons qu'il est important de rester dans le voisinage du point de vol, le paramètre le plus critique étant la vitesse. En effet, la sortie des aérofreins va induire un ralentissement de l'avion qu'il faudra minimiser afin de rester dans le voisinage du point. Pour cela, nous avons réalisé toutes les simulations nécessaires à l'inversion en activant l'auto-manette<sup>7</sup>. Remarquons qu'il ne s'agit pas d'une situation opérationnelle, puisque la fonction des A/F est de ralentir l'avion.

Nous présentons tout d'abord des résultats temporels (figures 8.14 et 8.15) pour deux points de vol distincts et suffisamment distants dans l'espace d'entrée. Les sorties observées

<sup>7</sup>Dispositif permettant de réguler la vitesse de l'avion via les moteurs

sont la pente  $\gamma$ , la vitesse verticale  $V_z$ , le facteur de charge  $Nz$ , le signal de compensation calculé par inversion  $\delta q_{\text{comp}}$ , la vitesse  $V_c$  et l'altitude  $z_p$ .

L'inversion est effectuée sur le seul critère en pente pour deux raisons : pour des faibles valeurs de  $\gamma$ , la  $V_z$  est proportionnelle à la  $V_c$  et le  $Nz$  est une fonction linéaire de  $\dot{\gamma}$  (cf. §6.2.6).

$$V_z \approx V_c \cdot \sin \gamma \approx V_c \cdot \gamma \propto \gamma$$

$$Nz = \cos \gamma + \frac{V_c}{g} \dot{\gamma} \approx 1 + \frac{V_c}{g} \dot{\gamma}$$

De plus, nous effectuons une régularisation de la solution en utilisant l'identité comme opérateur de régularisation et en calculant le paramètre de régularisation  $\lambda$  en utilisant l'heuristique proposée au paragraphe §5.4.2 (p. 87). Nous donnons à la figure 8.13 un exemple de résultat de calcul du paramètre de régularisation par cette méthode :

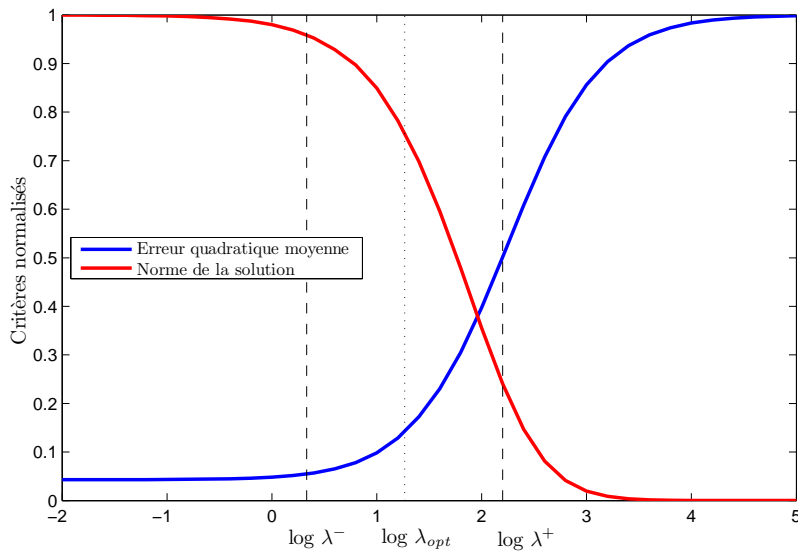


FIG. 8.13: Calcul de  $\lambda_{\text{opt}}$  par heuristique

Sur les figures 8.14 et 8.15, les courbes vertes représentent les réponses avion sans la compensation, les courbes bleues les réponses avec compensation et les pointillés rouges les limites en  $\gamma$  et  $V_z$  à ne pas dépasser.

Nous pouvons voir que l'inversion donne de très bons résultats vis-à-vis des critères en  $\gamma$  et en  $V_z$ . Remarquons que sur les dernières secondes de simulation, le comportement de la solution est quelque peu erratique. C'est un effet dû à la régularisation et à l'initialisation de l'algorithme de Landweber.

En effet, au vu de la forme de la solution à partir de  $t \approx 20$  s, nous avons préféré utiliser un paramètre de régularisation fenêtré plutôt qu'une constante sur tout l'horizon temporel considéré. Ce dernier choix donnait des résultats assez médiocres dans la mesure où, si  $\lambda$  était trop grand, il pénalisait la partie statique en privilégiant une faible norme pour la solution. S'il était trop faible, la partie dynamique devenait trop oscillante. Nous avons donc choisi de garder la valeur de  $\lambda$  calculée par l'heuristique pendant la phase dynamique puis de la ramener à zéro sur la partie statique (cf. figure 8.16).

L'absence de régularisation à la fin de l'horizon temporel explique les oscillations observées sur la solution étant donné que celle-ci est initialisée à zéro. Toutefois, cet effet de bord

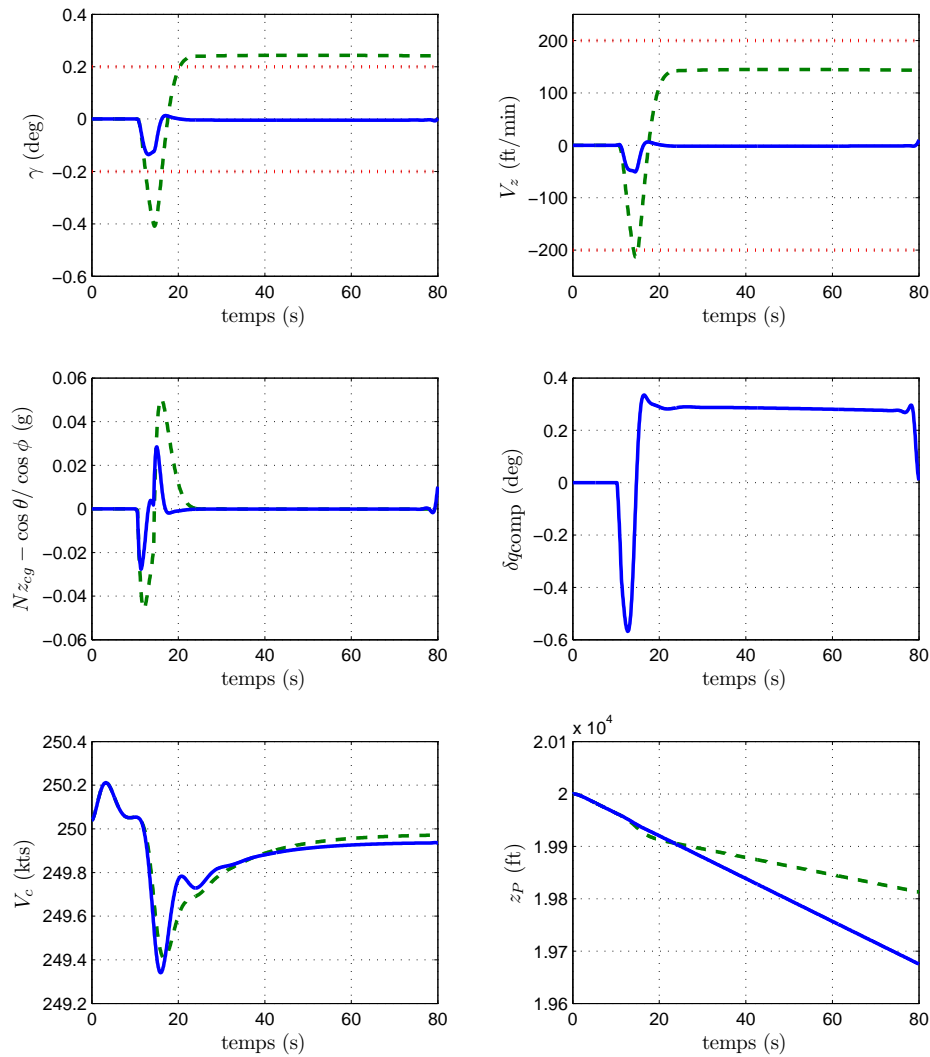


FIG. 8.14:  $V_c = 250$  kts,  $z_p = 20000$  ft (--- Sans compensation, — Avec  $\Delta u^*$ , ... Contraintes)

n'est pas gênant pour la suite car nous allons tronquer la solution de façon à supprimer ces oscillations.

Dans le tableau 8.1, nous donnons à titre indicatif le pourcentage de l'erreur résiduelle sur la sortie prévu à l'aide du modèle à RIF identifié comparé au pourcentage de l'erreur réelle obtenue en simulation.

Nous constatons une légère différence entre les taux d'erreurs prévus et réels, certainement liée à l'approximation de transfert à RIF. De plus ce modèle ne tient pas compte des variations paramétriques ayant lieu pendant les simulations de vérification sur le système réel.

Enfin, nous donnons à la figure 8.17 la performance des signaux de compensation calculés par inversion sur l'ensemble du domaine de vol. Nous avons représenté pour chaque point de vol un cercle rouge dont le diamètre est proportionnel à l'erreur maximale admissible en  $\gamma$  ( $\pm 0,2$  deg) et un cercle bleu dont le diamètre est proportionnel à l'erreur maximale observée en simulation avec un signal de compensation calculé par inversion.

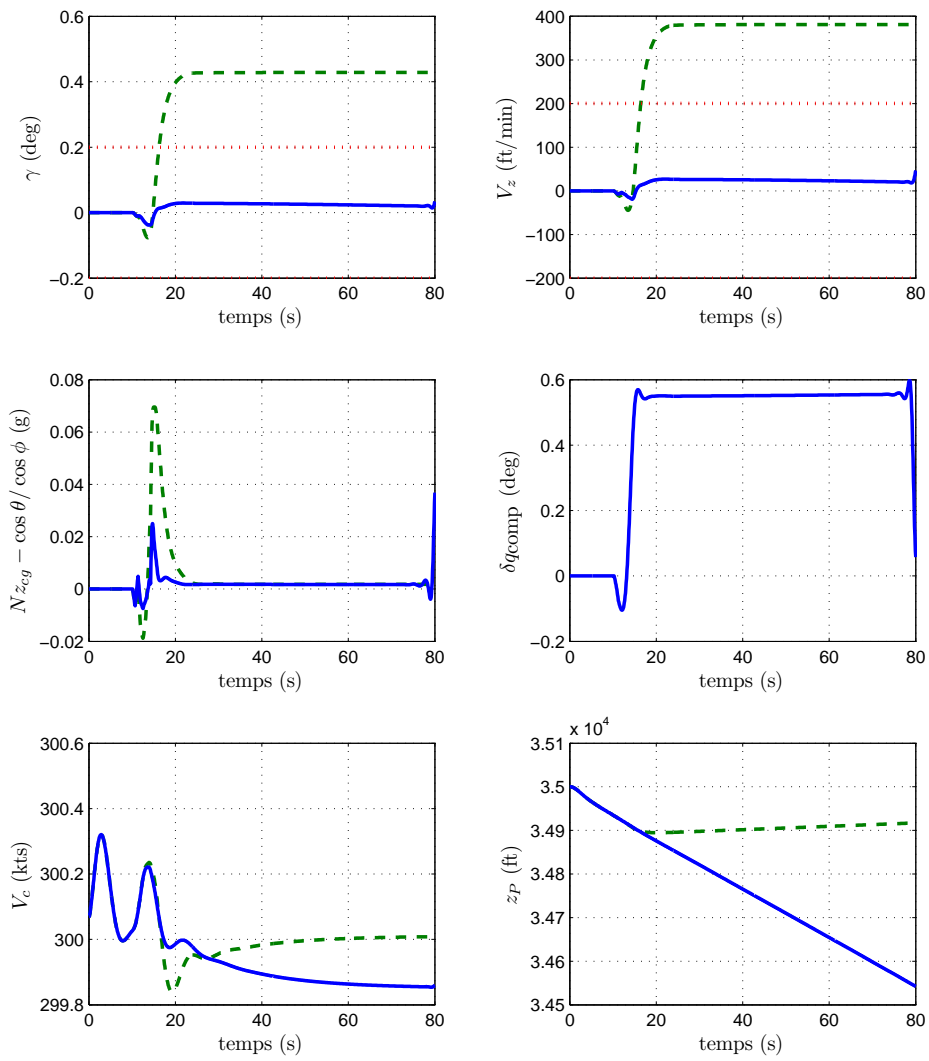


FIG. 8.15:  $V_c = 300$  kts,  $z_p = 35000$  ft (--- Sans compensation, - Avec  $\Delta u^*$ , ... Contraintes)

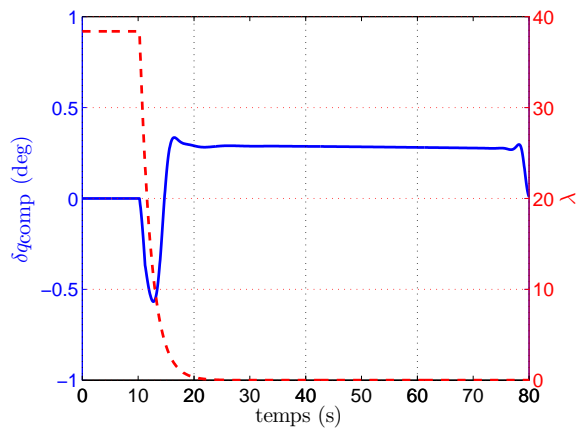
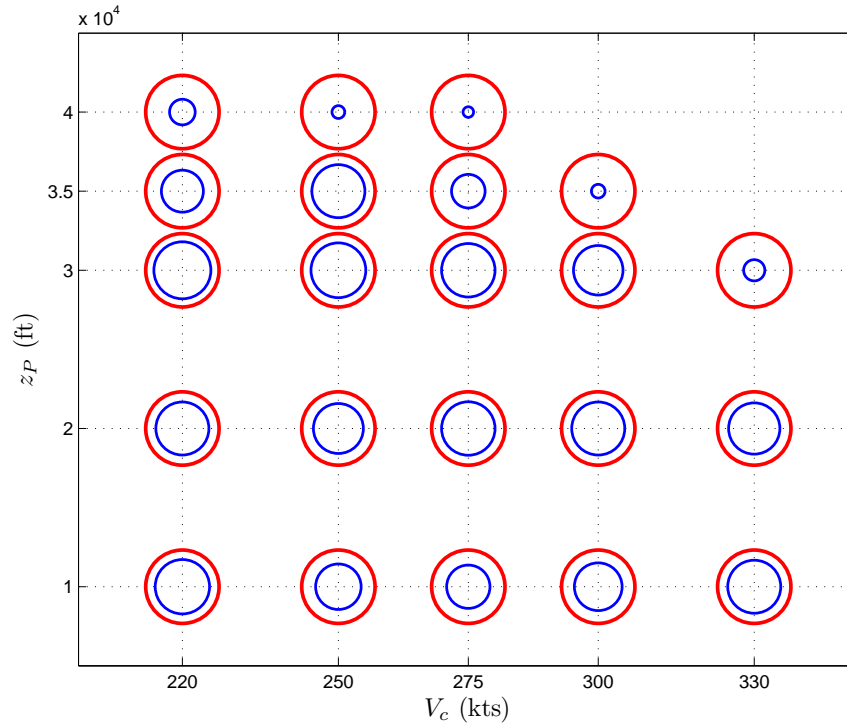


FIG. 8.16: Régularisation fenêtrée

$V_c, z_p$	$\gamma$		$V_z$		$Nz$	
	Prévu	Réel	Prévu	Réel	Prévu	Réel
250 kts, 20k ft	8,2%	11%	6,4%	10,3%	74,8%	75,5%
250 kts, 35k ft	8,9%	13,2%	6,5%	13,2%	68,2%	68,6%
300 kts, 20k ft	5,8%	7,3%	2,8%	7%	70,6%	71,7%
300 kts, 35k ft	12,4%	11,6%	12%	12,1%	78%	78%

TAB. 8.1: Performances de la méthode d'inversion

FIG. 8.17: Résultats de l'inversion sur le domaine de vol -  $\bigcirc$  Erreur maximale observée avec  $\Delta u^*$ ,  $\bigcirc$  Contraintes

Nous pouvons donc conclure de la bonne performance de l'inversion à laquelle nous pouvons ajouter un temps de calcul suffisamment court. En effet, il faut compter environ une quinzaine de secondes par point de vol, en incluant les trois simulations nécessaires (simulation initiale, simulation d'identification et simulation de vérification). L'inversion en elle-même ne dure qu'entre 1 et 2 secondes (soit une centaine d'itérations de l'algorithme de Landweber), dans la mesure où elle est arrêtée prématurément lorsque la norme de la différence entre deux itérés est inférieure à  $10^{-7}$ .

**Robustesse aux bruits de mesure** Enfin, pour tester la robustesse de l'inversion aux bruits de mesure, nous avons ajouté un bruit blanc gaussien  $\nu(t)$  centré et de variance de 20% de  $\|\mathbf{z}\|^2$ . La figure 8.18 compare les résultats de l'inversion pour le cas bruité (courbes bleu clair en trait plein) et le cas non bruité pour un même point de vol (courbes bleues en pointillés).

Nous pouvons voir que les résultats sont toujours bons, grâce à la régularisation qui filtre

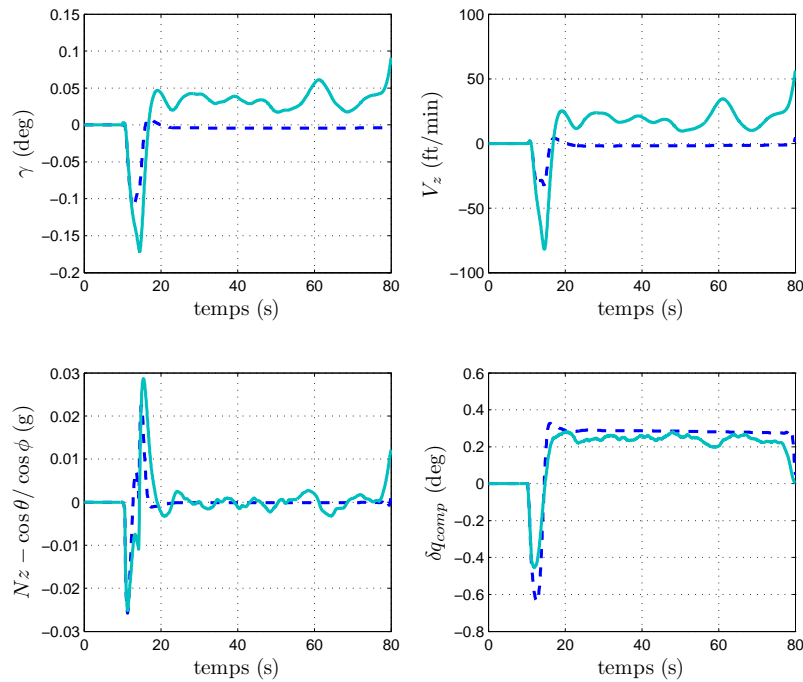


FIG. 8.18: Influence du bruit -  $V_c = 250$  kts,  $z_p = 20000$  ft (--- Inversion de référence, - Inversion avec régularisation de la norme)

les hautes fréquences en minimisant la norme de la solution. Cependant, nous remarquons un biais sur la pente et la vitesse verticale. Cela s'explique par le fait qu'à cause du bruit, nous avons dû régulariser la solution sur tout l'horizon temporel et non juste sur une fenêtre comme dans le cas non-bruité. Nous pouvons toutefois envisager une régularisation sur la dérivée discrète de la solution à l'aide de l'opérateur de régularisation suivant :

$$D = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 1 \\ 0 & \cdots & \cdots & 0 & -1 \end{pmatrix}$$

Nous présentons les résultats à la figure 8.19. Les courbes rouges en trait plein donnent les résultats pour une régularisation de la dérivée, les courbes bleu clair en traits-pointillés les résultats pour une régularisation de la norme de la solution et les courbes bleues en pointillés les résultats de l'inversion dans les cas non bruité.

Nous pouvons voir que nous n'avons plus le problème de biais sur la pente ou la vitesse verticale. Cependant la solution est paradoxalement plus agitée, notamment la réponse en  $Nz$ . Un paramètre de régularisation plus fort aurait davantage réduit ces oscillations, au détriment de la performance finale.

### 8.3.1.2 Synthèse

Au paragraphe §8.2.3, nous avons mis en évidence les dépendances de l'effet induit par les spoilers, que nous souhaitons compenser. La loi de compensation devra alors suivre

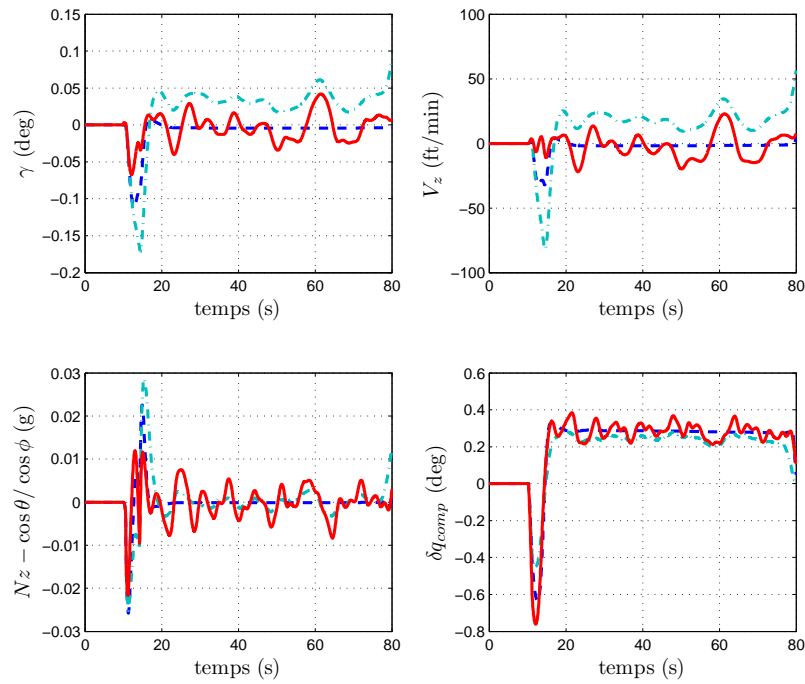


FIG. 8.19: Influence du bruit -  $V_c = 250$  kts,  $z_p = 20000$  ft (- - Inversion de référence, -·- Inversion avec régularisation de la norme, - Inversion avec régularisation de la dérivée)

ces mêmes dépendances. La figure 8.20 met en évidence les résultats de l'inversion sur le domaine de vol.

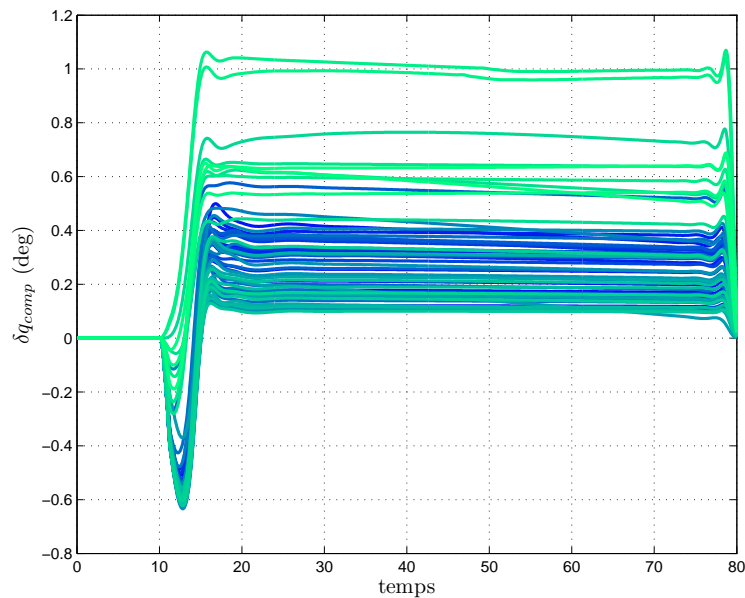


FIG. 8.20: Résultats des inversions sur tout le domaine de vol

Nous notons également une certaine homogénéité des résultats de l'inversion qui peuvent amener à reconsidérer éventuellement la structure de la loi existante. Après avoir observé



les réponses calculées par inversion sur tout le domaine de vol, nous pouvons relever certaines caractéristiques qui nous serviront à concevoir une loi de compensation de façon empirique :

- existence d'un non-minimum de phase variant en fonction de l'altitude (cf. figure 8.21) ; il faudrait inclure un zéro instable de l'ordre de 1 seconde.
- dynamique type 1<sup>er</sup> ordre durant le transitoire
- diverses amplitudes en fonction de  $V_c$  et  $z_p$  (cf. figures 8.21 et 8.22), impliquant l'interpolation des gains de la loi

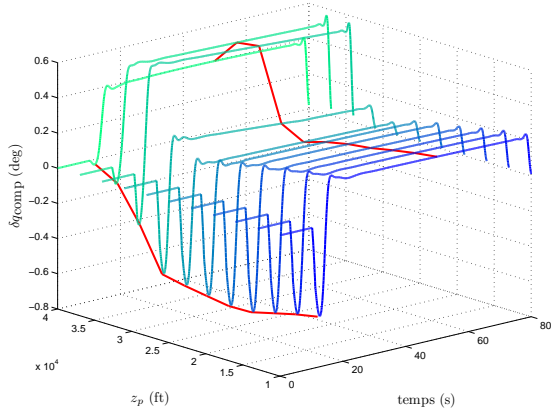


FIG. 8.21: Effet de  $z_p$  sur l'inversion à iso- $V_c$

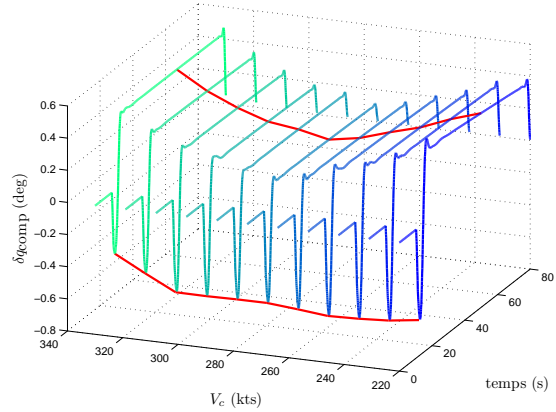


FIG. 8.22: Effet de  $V_c$  sur l'inversion à iso- $z_p$

Ceci nous amène à choisir une expression de la forme suivante pour structure de la loi de compensation :

$$\delta q_{comp} = K_{comp} \left( \frac{1 - \tau_1 s}{1 + \tau_2 s} \right) \cdot w \quad (8.4)$$

$$= K(s, \boldsymbol{\theta}) \cdot w \quad (8.5)$$

où  $s$  est la variable de Laplace et avec

$$\boldsymbol{\theta} = (K_{comp} \quad \tau_1 \quad \tau_2)^T \quad (8.6)$$

où  $w$  est l'ordre envoyé aux spoilers. Pour chaque point de vol, nous allons calculer les paramètres de la loi par identification puis les interpoler à l'aide d'un MEX (cf. §8.4.1). Pour poursuivre l'apprentissage en ligne de la compensation, il peut être préférable de mettre le correcteur  $K(s, \boldsymbol{\theta})$  sous forme de filtre en treillis normalisé afin de contrôler l'évolution de sa stabilité (cf. §8.4.2).

### 8.3.2 En-ligne

Pour générer les données d'apprentissage en ligne, il faut dans un premier temps calculer à chaque instant la valeur de  $\delta q_{comp}$  objectif. Pour ce faire, nous appliquons ici la méthode décrite au paragraphe §3.2.2. Nous avons tout d'abord identifié pour chaque point de vol généré, les matrices  $A$ ,  $B$  et  $C$  du système  $\hat{\Sigma}_2$  (cf. figure 3.4) à l'aide de la méthode des sous-espaces (routine `n4sid` sous MATLAB<sup>®</sup>) avec un ordre du système estimé de 4.

Nous procédons ensuite à la synthèse LQG sur chacun des modèles identifiés en utilisant un horizon de contrôle  $T_c = 1,7$  s et un horizon d'observation  $T_o = 2$  s. Ces valeurs ont fait l'objet d'une recherche manuelle permettant un asservissement rapide, indépendamment du dépassement observé sur une réponse indicielle, tout en conservant des gains peu élevés.

Les modèles et les gains LQG calculés ont ensuite été interpolés avec des réseaux RBF et la stabilité de la boucle fermée ainsi réalisée a été vérifiée sur un grand nombre de point du domaine de vol (cf. figure 8.23).

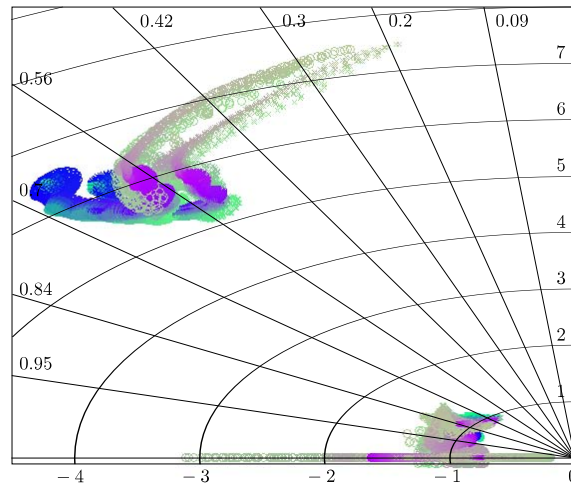


FIG. 8.23: Pôles du retour d'état ( $\times$ ) et de l'observateur ( $\circ$ ) interpolés

Nous pouvons voir que la boucle fermée est stable sur tout le domaine.

Il faut à présent vérifier la qualité de cette inversion en ligne par rapport à celle effectuée hors-ligne. C'est ce que montre la figure 8.24.

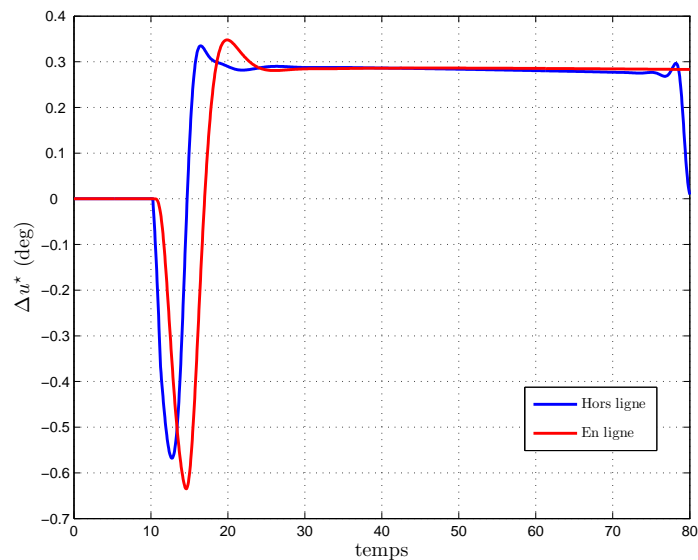


FIG. 8.24: Comparaison inversion en ligne et hors ligne

Nous notons un retard conséquent du signal généré en ligne dû à la dynamique de

la boucle LQG. La performance sera donc moindre mais il ne paraît pas judicieux de vouloir reproduire le signal hors ligne à l'identique pour deux raisons. Premièrement, cela impliquerait de réduire les horizons de commande et d'observation afin de forcer le système à répondre plus rapidement, ce qui se traduirait par une augmentation sensible des gains LQG et une potentielle instabilité de l'inversion. Ensuite, il faut garder à l'esprit l'utilisation que nous souhaitons en faire. En effet, nous comptons utiliser cette référence afin d'ajuster les paramètres d'un filtre en treillis par l'intermédiaire d'un algorithme LMS normalisé. Il ne serait pas raisonnable d'aligner la constante de temps  $\eta'$  de l'algorithme donné en (4.56) et (4.57), sur celle du système, ce qui pourrait provoquer l'instabilité de l'algorithme de mise à jour des paramètres.

## 8.4 Apprentissage

Nous allons à présent donner les performances d'apprentissage des MEX sur les paramètres de la loi de compensation générés précédemment. Nous avons à considérer deux structures différentes : celle proposée au paragraphe §8.3.1.2 et la structure en treillis que nous allons utiliser pour effectuer l'apprentissage en ligne. Nous avons donc trois paramètres à modéliser pour chaque structure :

$$\begin{aligned}\boldsymbol{\theta}_{standard} &= (K_{comp} \quad \tau_1 \quad \tau_2)^\top \\ \boldsymbol{\theta}_{treillis} &= (\phi_1 \quad \nu_0 \quad \nu_1)^\top.\end{aligned}$$

Rappelons que  $\phi_1$  règle la dynamique du filtre et que  $\nu_0$  et  $\nu_1$  jouent sur le gain et le zéro du filtre.

### 8.4.1 Hors ligne

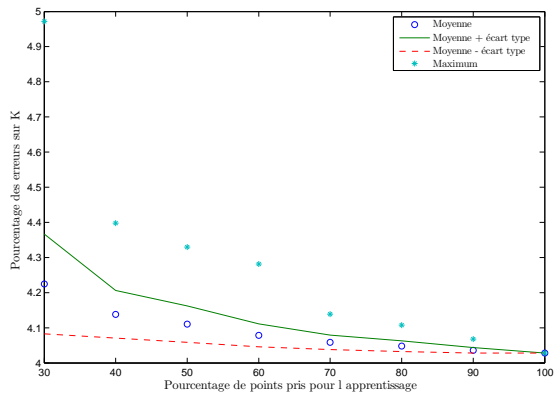
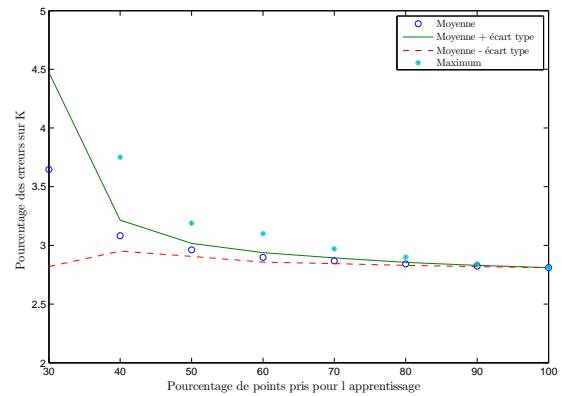
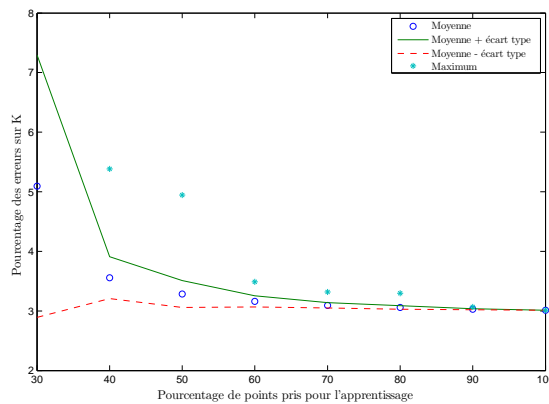
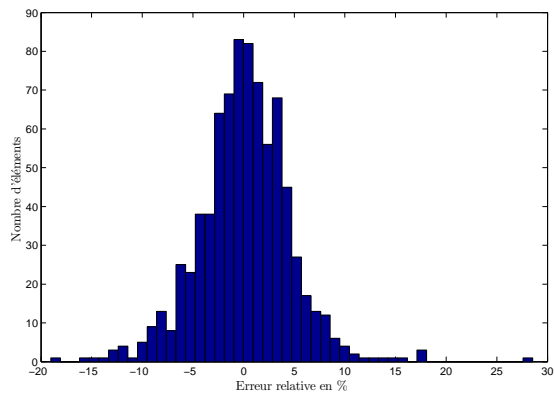
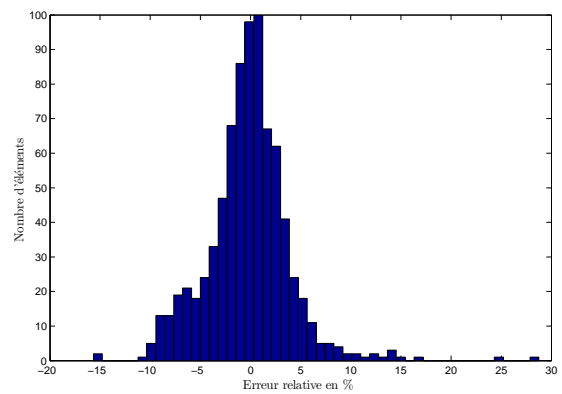
Nous utilisons une partie des résultats du paragraphe §7.2.1 afin de définir les MEX utilisés : nous utiliserons le même critère (7.9) afin de chiffrer nos résultats et nous donnerons aux MEX une structure maillage régulier/norme  $\|\cdot\|_2$ . Il nous reste alors à déterminer le maillage de l'espace d'entrée  $(V_c, z_p)$ . Pour cela, nous avons différentes possibilités pour lesquelles nous avons étudié l'influence de la taille de base de données d'apprentissage afin d'évaluer la capacité de généralisation du MEX (cf. §7.2.1.1, p. 112). Nous donnons les résultats aux figures 8.25 à 8.27 pour trois maillages différents, pour l'apprentissage du paramètre  $K_{comp}$ .

Nous pouvons voir sur ces figures que le meilleur choix semble être le maillage  $3 \times 7$ . En effet, ce dernier présente la meilleure performance ainsi que la plus faible erreur maximale à partir d'une « taille » de 40% de la base de données totale. Des résultats similaires sont observés sur les autres paramètres.

Les figures 8.28 à 8.30 donnent les répartitions des erreurs de validation pour les éléments du vecteur  $\boldsymbol{\theta}_{standard}$ .

Les performances d'apprentissage et de validation du MEX sont satisfaisantes et seront davantage mises en évidence par la performance de la loi interpolée, que nous allons voir par la suite.

La figure 8.31 montre les résultats obtenus par résolution du problème d'identification 3.3.1 (p. 55) sur tout le domaine de vol avec la structure de correcteur présentée en (8.4). Le modèle utilisé pour valider le comportement de la loi est le modèle de synthèse. La symbolique est identique à la figure 8.17 : les cercles rouges représentent l'erreur maximale

FIG. 8.25: Influence de la taille de la base données d'apprentissage - Maillage  $3 \times 3$ FIG. 8.26: Influence de la taille de la base données d'apprentissage - Maillage  $3 \times 7$ FIG. 8.27: Influence de la taille de la base données d'apprentissage - Maillage  $5 \times 5$ FIG. 8.28: Répartition des erreurs de validation pour  $\tau_1$ FIG. 8.29: Répartition des erreurs de validation pour  $\tau_2$ 

admissible sur  $\gamma$ , les cercles bleus l'erreur maximale constatée en utilisant le signal calculé par inversion et les cercles verts l'erreur maximale constatée en utilisant la loi interpolée.

Nous constatons que la performance est toujours satisfaisante quoique légèrement dégradée, en particulier dans le domaine des hautes altitudes et basses vitesses.

A présent, observons la performance de la loi sur le modèle de validation (que nous assimilerons au vrai avion), qui n'est autre que le modèle de synthèse dont le coefficient

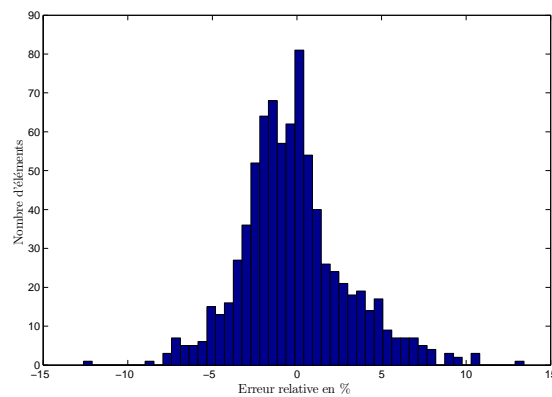


FIG. 8.30: Répartition des erreurs de validation pour  $K_{comp}$

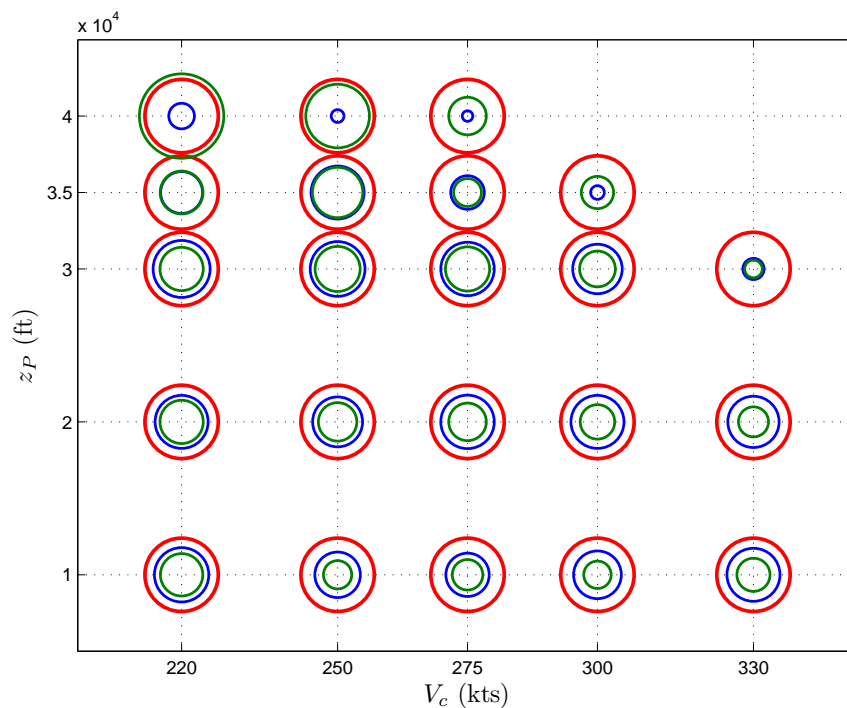


FIG. 8.31: Résultats de la synthèse sur le domaine de vol (modèle initial) –  $\circ$  Erreur maximale observée avec  $\Delta u^*$ ,  $\circ$  Erreur maximale observée avec la loi déterminée,  $\circ$  Contraintes

$\Delta C m_{sp}$  a été augmenté de 20%, le  $\Delta C z_{sp}$ , le  $\Delta C z_{\delta q}$  et le  $\Delta C x_{sp}$  augmentés de 5%. Ces recalages constituent des pire-cas de modification observée sur ces coefficients après identification de l'avion. Sur la figure 8.32, les cercles magenta représentent l'erreur maximale observée avec la loi interpolée sur le modèle de validation. Les cercles verts ont été laissés à titre de comparaison.

Nous constatons une dégradation assez nette de la performance sans être toutefois catastrophique. Il y a en effet davantage de zones en dehors de la tolérance mais l'écart reste faible. Nous envisageons donc de venir retoucher notre loi en ligne, sur « l'avion réel » (i.e. le modèle de validation).

Avant de procéder à l'apprentissage, nous devons mettre la loi sous forme de filtre en treillis normalisé, même si la complexité de la loi ne l'impose pas. En effet, il s'agit

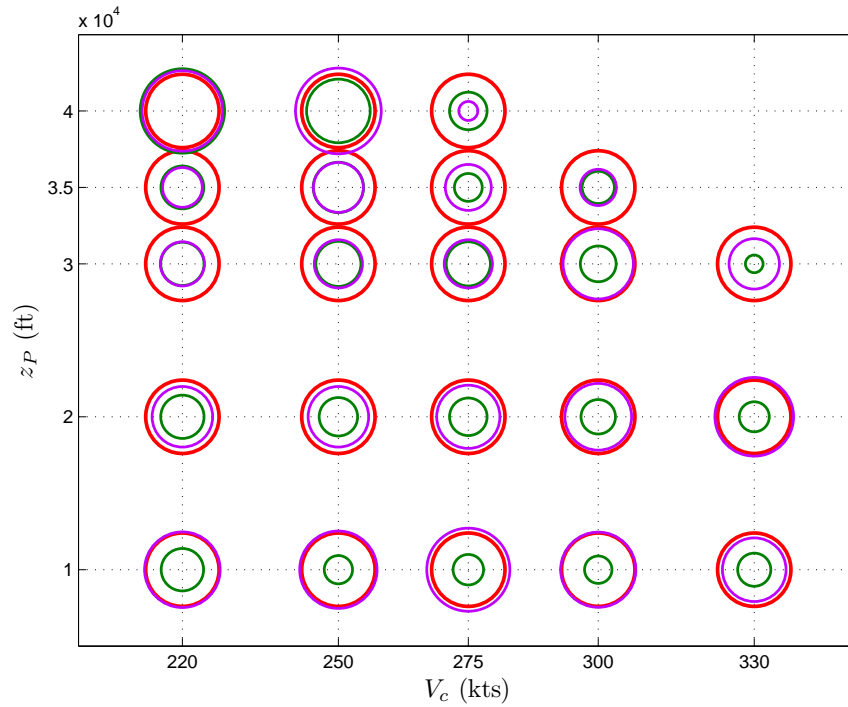


FIG. 8.32: Résultats de la synthèse sur le domaine de vol (modèle pire-cas) –  $\circ$  Erreur maximale observée avec  $\Delta u^*$ ,  $\circ$  Erreur maximale observée avec la loi sur le modèle initial,  $\circ$  Erreur maximale observée avec la loi sur le modèle réel,  $\circ$  Contraintes

d'un filtre du premier ordre dont la stabilité est simple à gérer. Toutefois, dans l'optique de valider la méthode proposée, nous faisons abstraction de cette augmentation de complexité.

La relation  $f : \theta_{standard} \mapsto \theta_{treillis}$  n'est pas triviale mais le calcul de  $\theta_{treillis}$  peut se faire en discrétisant le filtre  $K_c(s, \theta_{standard})$  puis en appliquant l'algorithme de calcul donné dans [Regalia, 1994] aux coefficients du filtre discrétisé.

Nous appliquons donc cet algorithme aux données générées et réalisons l'apprentissage en utilisant un MEX identique au précédent. Les performances obtenues avec la loi en treillis sont équivalentes à celles obtenues avec la forme précédente.

### 8.4.2 En ligne

Dans ce paragraphe, nous rappelons que la loi est sous la forme d'un filtre en treillis normalisé. Nous allons tout d'abord vérifier le caractère local de la retouche, en considérant le point de vol où la performance est la plus mauvaise. Les figures 8.33 et 8.34 montrent les évolutions temporelles de la pente  $\gamma$  (traits pleins) et du facteur de charge  $Nz$  (pointillés) avant (en bleu), pendant (en orange) et après la retouche (en vert), et l'évolution de l'ordre de compensation (trait plein) et la référence (en pointillés) pendant la retouche.

Nous pouvons voir une nette amélioration de l'erreur maximale sur la pente. Il faut à présent s'assurer du caractère local de la retouche, comme nous l'avons fait au chapitre précédent. Pour cela, nous allons observer dans un premier temps la norme de la modification des vecteurs de paramètres des modèles locaux du réseau RBF après la retouche. Nous présentons cette observation à la figure 8.35 dans l'espace d'entrée normalisé, sous forme de cercle centré sur chacun des centres du réseau et dont le diamètre est proportionnel à l'évolution de la norme des paramètres du modèle local associé au centre. La courbe verte

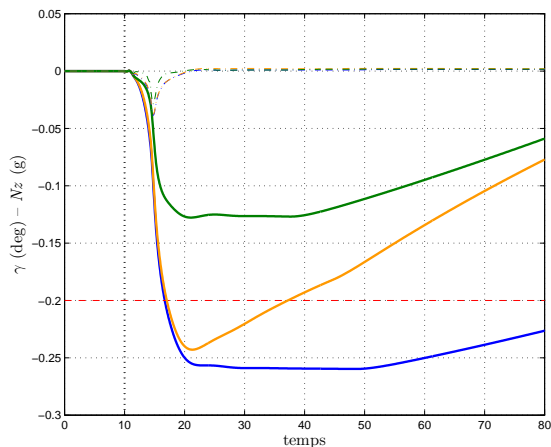


FIG. 8.33: Allures de  $\gamma(t)$  et  $Nz(t)$  avant, pendant et après l'apprentissage

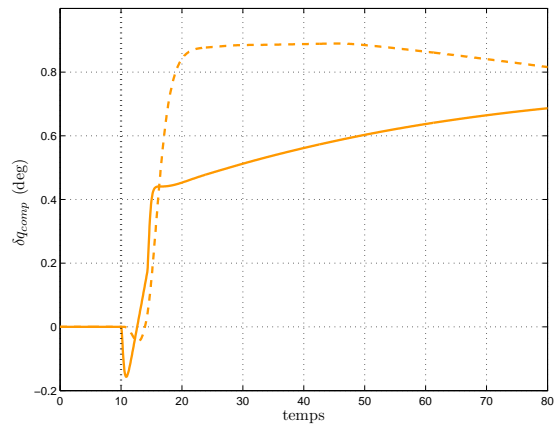


FIG. 8.34: Commande  $\delta q_{comp}(t)$  calculée pendant l'apprentissage et référence  $v(t)$

représente la trajectoire parcourue dans l'espace d'entrée pendant la retouche.

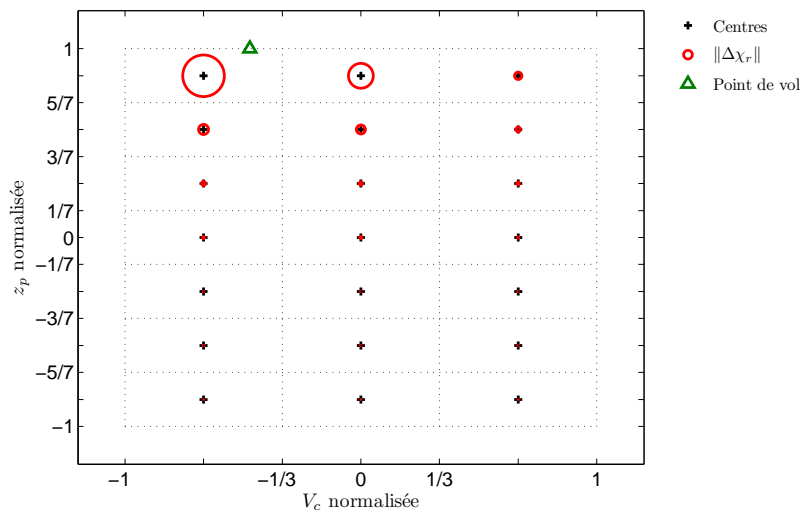


FIG. 8.35: Norme de la modification des modèles locaux

Nous voyons bien que seuls les modèles dont les centres sont à proximité du point courant sont retouchés. Mais il faut également vérifier que les modifications n'ont pas affecté la performance de la loi sur tout le domaine de vol. La figure 8.36 montre cette performance après la retouche (cercles bleu clair), comparée à celle observée avant la retouche (cercles magenta).

Le résultat est presque entièrement satisfaisant, dans la mesure où la performance n'a été dégradée que dans le secteur haute altitude et basse vitesse ( $V_c = 220$  kts et  $z_p = 40000$  ft). Cela n'est pas « inquiétant » car nous avons pu observer que la loi n'est pas bonne initialement. Nous allons donc retoucher ce point-là.

A présent, nous allons effectuer une retouche sur chacun des points présentés sur les figures précédentes, puis vérifier la performance sur l'ensemble des points de validation. La figure 8.37 donne le nombre de points associés à un intervalle d'erreur maximum en pente avant la retouche complète (en bleu) et après la retouche complète (en magenta).

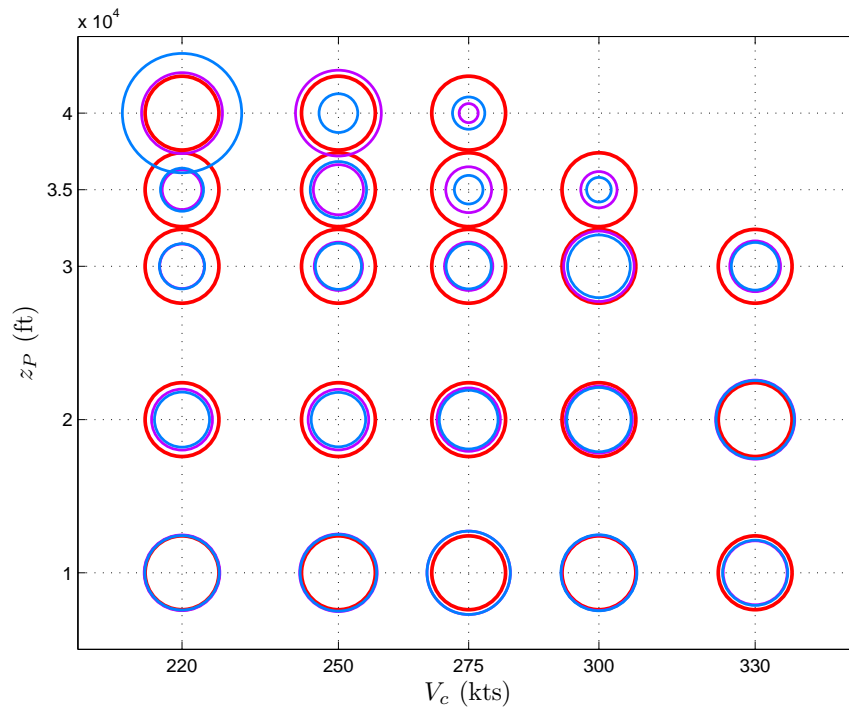


FIG. 8.36: Résultats sur le domaine de vol après apprentissage sur le pire-cas –  $\circ$  Erreur maximale observée avant retouche,  $\circ$  Erreur maximale observée après retouche,  $\circ$  Contraintes

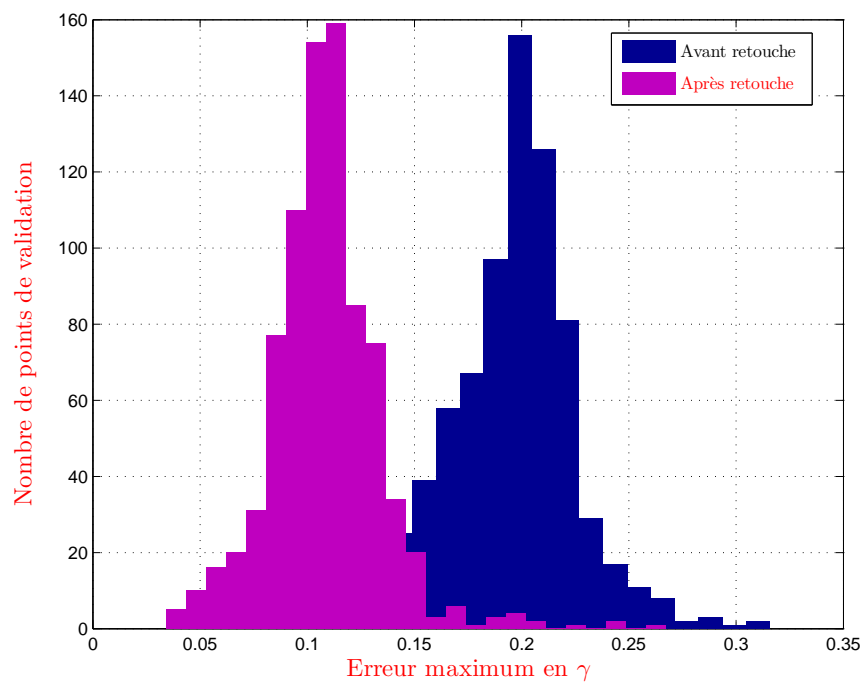


FIG. 8.37: Résultats après apprentissage sur tout le domaine de vol

Nous pouvons voir que la performance a été nettement améliorée et seuls quelques points sont en dehors des performances attendues.



## 8.5 Conclusion

Nous venons de présenter une mise en pratique réussie de la méthodologie développée à la partie II. Nous avons réussi à améliorer la performance de la loi de compensation sur l'ensemble du domaine de vol, pour des niveaux de recalage de coefficients aérodynamiques correspondant à un pire-cas. Nous avons également mis en évidence le caractère local de l'apprentissage, à l'instar du chapitre précédent. Toutefois, dans l'apprentissage en ligne, la condition d'excitation permanente (nécessaire pour la convergence des filtres adaptatifs) n'est pas satisfaite. L'apprentissage est stoppé à la fin d'un échelon d'une centaine de seconde. L'algorithme n'a donc peut-être pas convergé vers la solution optimale. Si la vérification s'avère infructueuse, la procédure devra alors être répétée jusqu'à obtention d'un résultat satisfaisant.

Quatrième partie

Conclusion et perspectives



## Chapitre 9

# Conclusion et perspectives

### Bilan et apports

L'objectif de cette thèse était de développer des méthodologies de réglage automatique pour des lois de pilotages dont le réglage s'avère long et fastidieux à cause des différences de comportement entre les modèles de synthèse et l'avion réel. Les principales causes de ces différences ont été identifiées comme étant les coefficients aérodynamiques de l'avion, régissant son comportement dynamique. En effet, nous avons pu constater que certains coefficients pouvaient être modifiés après identification, d'environ 20% de leur valeur initialement estimée. Nous avons donc cherché des solutions permettant l'utilisation directe de données issues de l'avion réel afin de s'affranchir autant que possible des erreurs inhérentes aux modèles de synthèse.

Nous avons concentré notre travail sur les deux aspects suivants :

- le caractère PV de l'avion. En effet, ce dernier se comporte différemment selon la valeur de différents paramètres internes aussi bien qu'externes : sa masse, son centrage, sa vitesse par rapport à l'air, l'altitude à laquelle il évolue, etc. Ces paramètres définissent le *domaine de vol* de l'avion. Les lois de pilotage sont alors synthétisées avec ces mêmes dépendances paramétriques ou bien rendues insensibles à ces paramètres si la précision de leur mesure n'est pas suffisante pour assurer un niveau de performance satisfaisant. Ces lois ont une structure fixe et sont interpolées par l'intermédiaire de leur paramètres caractéristiques (gain, constante de temps, etc.). Nous pouvons donc considérer que le réglage d'une loi de pilotage passe par l'identification d'une fonction d'interpolation ;
- le caractère local de l'information utilisée. Cette dernière provient d'essais en vol qui sont réalisés sur un ensemble de *points de vol* (i.e. des points situés dans le domaine de vol). Les données récoltées sont donc spécifiques à un point de vol et son voisinage. Nous avons considéré qu'il était essentiel que le réglage soit également local.

Ces deux aspects nous ont fait nous tourner vers la notion d'apprentissage local et plus précisément de mélange d'experts (MEX), qui permet d'intégrer une connaissance locale dans la structure de la solution au problème d'interpolation [Ronceray *et al.*, 2007a]. Ces notions ont fait l'objet de la partie I.

En intégrant cette approche dans un contexte d'identification et de contrôle adaptatif, nous avons développé, au cours de la partie II, une méthode « à géométrie variable », dans le sens où l'utilisateur peut choisir parmi plusieurs possibilités en fonction de l'application visée et des ressources à sa disposition. Nous avons séparé cette méthode en trois principaux

thèmes : la génération de données, l'apprentissage et la régularisation. Les deux premiers thèmes ont été traités à la fois pour un travail hors ligne (en bureau d'études) [Ronceray *et al.*, 2008] et pour un travail en ligne (pendant un essai en vol) [Ronceray *et al.*, 2009]. Bien que nous ayons fait l'effort de systématiser la méthode, nous n'avons pas cherché à la rendre entièrement automatique. Certains aspects doivent malgré tout faire l'objet d'une réflexion de la part de l'utilisateur (structure du mélange d'experts, structure de la loi à régler).

Cette méthode a fait l'objet de deux applications, identifiées par Airbus comme reflétant le problème posé. Elles ont été traitées au cours de la partie III. La première concerne l'estimateur d'une grandeur aérodynamique, l'angle de dérapage  $\beta$ . Nous avons élaboré à partir des équations de la mécanique du vol, une estimation de cet angle faisant intervenir des coefficients aérodynamiques. Ces derniers ont été modélisés à l'aide de MEX ayant appris sur un modèle de synthèse puis recalés à l'aide de données d'essais en vol [Ronceray *et al.*, 2007b]. La seconde application concerne la loi de compensation d'un effet dynamique lié à l'extension de surfaces de contrôle, les spoilers. Nous avons élaboré une loi à l'aide de la méthode de synthèse par inversion [Ronceray *et al.*, 2008] sur un modèle de synthèse puis avons ajusté la loi en temps-réel sur le modèle de synthèse dont les coefficients aérodynamiques ont été modifiés de façon à refléter un pire-cas de recalage après identification.

Nous avons alors illustré la façon de procéder quant au réglage des degrés de liberté à disposition de l'utilisateur. Plus concrètement, la méthode que nous avons proposée a permis de systématiser un processus de réglage fonctionnant jusque là « à tâtons ». Dans le cadre du développement d'un nouvel avion, nous pouvons envisager deux façons de procéder.

La première consisterait à n'utiliser que les aspects hors ligne de la méthode sur des enregistrements d'essais en vol. Nous pensons notamment aux essais dédiés à l'identification de l'avion. Cela permet de synthétiser directement la loi sur le système de référence. Il n'est alors plus nécessaire d'interpoler la loi ainsi déterminée à l'aide de MEX puisqu'elle ne sera vraisemblablement pas retouchée par la suite. Cette approche est apparue applicable directement et rapidement dans le cadre d'un nouveau programme. Elle permet d'accomplir un travail de mise au point de loi de pilotage en parallèle du processus d'identification.

La seconde consisterait à utiliser intégralement la méthode. Une synthèse préalable serait réalisée sur les modèles préliminaires en suivant la partie hors ligne de la méthode. La loi devra alors être interpolée par des MEX dans l'éventualité d'une retouche en temps réel de loi dans les zones du domaine de vol où les écarts de modèles pénalisent trop la performance. Ces retouches seraient réalisées dans le cadre d'essais en vol spécifiques, au cours desquels le processus d'apprentissage pourrait être répété sur un même point de vol. Cette approche est moins évidente à appliquer à court terme dans la mesure où l'implémentation des MEX dans un calculateur semble difficilement envisageable à l'heure actuelle. Elle permet néanmoins de se détacher encore davantage du processus d'identification.

## Perspectives méthodologiques

Dans le cadre de cette thèse, nous avons proposé des méthodes proposant une solution à plusieurs problèmes industriels. Elles sont donc justifiées pour des applications spécifiques mais nous avons formalisé ces problèmes afin de donner un caractère générique à la méthode

proposée. Toutefois, nous n'avons abordé que les cas d'un estimateur statique et des lois feedforward. Il pourrait alors être intéressant d'élargir ce travail aux cas d'un estimateur dynamique et des lois feedback.

Dans [Seren et Hardier, 2008], les auteurs proposent un estimateur dynamique de masse/centrage, utilisant une modélisation de coefficients aérodynamiques par des réseaux de neurones « embarquables » et s'appuyant sur un filtrage de Kalman étendu pour estimer les paramètres de masse et centrage.

Pour les lois feedback, des travaux ont déjà été réalisés dans [Jean, 2001], [Cumer et Mouyon, 2003]. Ils s'inspirent de la démarche utilisée dans cette thèse, et s'appuient sur le schéma présenté en figure 9.1.

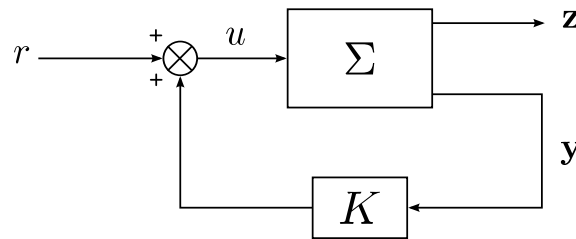


FIG. 9.1: Système à retoucher

Le principe est de déterminer une réponse souhaitée en  $\mathbf{z}$  et de calculer l'erreur résiduelle  $\Delta\mathbf{z}$ . La RI du système en boucle fermée vue de la commande est ensuite identifiée et  $\Delta\mathbf{z}$  est inversée de façon à obtenir un signal de commande optimal  $\Delta\mathbf{u}^*$ . La variation effective  $\Delta\mathbf{y}$  est alors calculée par simulation du système en appliquant  $\Delta\mathbf{u}^*$ . Le gain  $\Delta K$  reproduisant au mieux  $\Delta\mathbf{u}^*$  est alors calculé à partir de  $\mathbf{y} + \Delta\mathbf{y}$ . Nous pouvons voir que cette méthodologie est très proche de la méthode de synthèse par inversion que nous avons présentée au cours la partie II.

## Perspectives applicatives

Lors du traitement des applications, nous avons soulevé quelques points spécifiques à ces dernières que nous n'avons pas eu le temps de traiter.

Tout d'abord, concernant l'estimateur de dérapage, nous pourrions reposer le problème de l'apprentissage hors ligne sous forme implicite. En effet, nous avons extrait les coefficients à modéliser à partir d'une expression explicite du dérapage. Nous avons donc modélisé des rapports de coefficients et non les coefficients. Il est envisageable de poser le problème de façon implicite, en calculant directement les coefficients aérodynamiques satisfaisant :

$$mgNy = -\frac{1}{2}\rho S V_a^2 \left( Cy_\beta \cdot \beta + \Delta Cy_\beta + \frac{\ell}{V_a} (Cy_p \cdot p + Cy_r \cdot r) + \Delta Cy_{\delta p} + \Delta Cy_{\delta r} \right)$$

Nous pourrions également envisager d'améliorer l'estimateur en affinant davantage la modélisation du  $Cy$ , en incluant par exemple les effets dus à  $\dot{\beta}$ . Il serait également souhaitable de prendre en compte les cas de pannes moteurs asymétriques au cours de l'apprentissage.

Enfin, nous pourrions considérer l'utilisation de filtres adaptatifs pour filtrer les données utilisées en temps réel afin d'améliorer la qualité de l'estimation ou bien d'utiliser des filtres

interpolés par des MEX après un processus d'apprentissage portant sur la valeur optimale des paramètres de ces filtres.

Pour ce qui est des lois de compensation en boucle ouverte, il en existe un certain nombre dans les lois de pilotage des avions Airbus auxquelles nous pourrions aisément appliquer la méthodologie proposée :

- compensation des becs/volets : l'objectif est de limiter les variations d'altitude liées à la rentrée/sortie des becs/volets à l'aide d'une loi prenant en entrée la déflexion de ces surfaces ;
- compensation de  $Nx$  : il s'agit de limiter la prise d'assiette longitudinale liée à une variation de poussée moteur à l'aide d'une loi prenant en entrée les paramètres de contrôle moteur ;
- compensation d'effet de sol (loi flare<sup>1</sup>) : il s'agit de compenser la prise d'assiette liée à la variation de portance causée par l'approche du sol lors des phases d'atterrissage. La loi envoie un ordre à la gouverne de profondeur à partir de la dérivée temporelle de la vitesse par rapport au sol ;
- compensation de gradient de vent : il s'agit de compenser la prise d'assiette liée à la modification d'incidence induite par un gradient lors des phases d'approche. La loi envoie un ordre à la gouverne de profondeur à partir de la différence entre la vitesse air corrigée et la vitesse par rapport au sol ;
- réglage de la coordination de virage : il s'agit de réaliser un virage à dérapage nul en agissant uniquement avec le manche. Pour cela, une précommande est envoyée sur la gouverne de direction à partir de l'ordre manche ;
- contrôle de charges voilures [Wildschek *et al.*, 2006], [Ronceray, 2005] : il s'agit de minimiser les charges (i.e. les efforts) en turbulence appliquées sur la voilure, mesurées en essais en vol par des capteurs spécifiques. La loi calcule alors un ordre envoyé aux ailerons à partir de la variation d'incidence  $\alpha_{\text{vent}}$  liée à la turbulence.

Concernant plus particulièrement la compensation des spoilers, il est envisageable d'augmenter légèrement la complexité de la loi afin d'affecter à chaque groupe de surfaces (voire à chaque surface) un correcteur spécifique et non plus un correcteur global. La loi aurait la forme suivante :

$$\delta q_{\text{comp}} = \sum_{i=1}^{N_{\text{spoilers}}} K_i \cdot \frac{1 - \tau_{1,i}s}{1 + \tau_{2,i}s}$$

La compensation pourrait alors gagner en efficacité en affectant une dynamique propre à chaque spoiler.

---

<sup>1</sup>Il s'agit d'une loi spécifique utilisée à la fin de la phase d'approche, au moment de l'arrondi effectué avant de toucher le sol.

# Annexes





# Annexe A

## Rappels mathématiques

### A.1 Algèbre

PRODUIT DE KRONECKER : Soient  $A = (a_{ij})$  une matrice de taille  $N \times P$  et  $B = (b_{ij})$  une matrice de taille  $M \times Q$ . Le produit de Kronecker de  $A$  par  $B$  est la matrice  $NM \times PQ$  telle que

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1p}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{np}B \end{pmatrix}$$

◇

NORME EUCLIDIENNE : La norme euclidienne (ou norme  $\|\cdot\|_2$ ) d'un vecteur  $\mathbf{a}$  de taille  $M$  est la racine carrée de la somme de ses  $M$  composantes au carré :

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{m=1}^M a_m^2} = \mathbf{a}^T \mathbf{a}$$

◇

NORME-2 D'UN OPÉRATEUR : La norme  $\|\cdot\|_2$  d'un opérateur linéaire  $A$  est définie par :

$$\|A\|_2 = \sup_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$$

◇

NORME- $\infty$  : La norme  $\|\cdot\|_\infty$  d'un vecteur  $\mathbf{a}$  de taille  $M$  est le maximum de la valeur absolue de ses  $M$  composantes :

$$\|\mathbf{a}\|_\infty = \max_m \{|a_m| \mid m = 1, \dots, M\}$$

◇

## A.2 Analyse

PRODUITS DE CONVOLUTION : 1. *Cas continu* : soient  $x(t)$  et  $y(t)$  deux signaux continus. Le produit de convolution continu de  $x$  et  $y$ , noté  $x * y(t)$ , s'écrit :

$$x * y(t) = \int_{-\infty}^{\infty} x(t - \tau)y(\tau)d\tau$$

2. *Cas discret* : soient  $x(n)$  et  $y(n)$  deux signaux discrets. Le produit de convolution discret de  $x$  et  $y$ , noté  $x * y(n)$ , s'écrit :

$$x * y(n) = \sum_{k=-\infty}^{\infty} x(n - k)y(k)$$

◇

PRODUIT DE CONVOLUTION CIRCULAIRE : 1. *Cas continu* : soient  $x(t)$  et  $y(t)$  deux signaux continus, modélisés par des fonctions périodiques de période  $T$  et de motif période  $x_T(t)$  et  $y_T(t)$ . Le produit de convolution de  $x$  et  $y$  est alors le signal  $c(t)$  périodique de période de  $T$ , obtenu par convolution de l'un des deux signaux par le motif période de l'autre :

$$\begin{aligned} c(t) &\triangleq x * y_T(t) = \int_0^T x(t - \tau)y_T(\tau)d\tau \\ &\triangleq y * x_T(t) = \int_0^T y(t - \tau)x_T(\tau)d\tau \end{aligned}$$

Le motif période  $c_T(t)$  du produit de convolution est dit obtenu par convolution circulaire des motifs périodes  $x_T(t)$  et  $y_T(t)$ .

$$c_T(t) \triangleq x_T(t) \circledast y_T(t)$$

2. *Cas discret* : soient  $x(n)$  et  $y(n)$  deux signaux discrets de même période  $N$  et de motifs périodes  $x_N(n)$  et  $y_N(n)$ . Le produit de convolution de  $x$  et  $y$  est alors le signal  $c(n)$  périodique de période de  $N$ , obtenu par convolution de l'un des deux signaux par le motif période de l'autre :

$$\begin{aligned} c(n) &\triangleq x * y_N(n) = \sum_{k=0}^{N-1} x(n - k)y_N(k) \\ &\triangleq y * x_N(n) = \sum_{k=0}^{N-1} y(n - k)x_N(k) \end{aligned}$$

Le motif période  $c_N(n)$  du produit de convolution est dit obtenu par convolution circulaire des motifs périodes  $x_N(n)$  et  $y_N(n)$ .

$$c_N(n) \triangleq x_N(n) \circledast y_N(n)$$

◇

GRADIENT D'UNE FONCTION DE  $\mathbb{R}^n$  DANS  $\mathbb{R}$  : Soit  $f : \mathbf{x} \mapsto f(\mathbf{x})$ , une fonction scalaire différentiable de  $\mathbb{R}^n$  dans  $\mathbb{R}$ . La dérivée de  $f$  par rapport à  $\mathbf{x}$ , ou matrice Jacobienne, s'écrit :

$$\frac{\partial f}{\partial \mathbf{x}} = \left( \frac{\partial f(\mathbf{x})}{\partial x_1} \quad \dots \quad \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$$

Le gradient de  $f$  en  $\mathbf{x}$  est le vecteur colonne transposé de  $\frac{\partial f}{\partial \mathbf{x}}$  :

$$\nabla_{\mathbf{x}} f = \left( \frac{\partial f}{\partial \mathbf{x}} \right)^{\top}$$

◇

MATRICE JACOBIENNE D'UNE FONCTION DE  $\mathbb{R}^n$  DANS  $\mathbb{R}^m$  : Soit  $\mathbf{f} : \mathbf{x} \mapsto \mathbf{f}(\mathbf{x})$ , une fonction vectorielle différentiable de  $\mathbb{R}^n$  dans  $\mathbb{R}^m$  :

$$\mathbf{f} : \mathbf{x} \mapsto \mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix}$$

La dérivée de  $\mathbf{f}$  par rapport à  $\mathbf{x}$ , ou matrice Jacobienne, est la matrice de taille  $m \times n$  :

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

◇

MATRICE JACOBIENNE D'UNE FONCTION DE  $\mathbb{R}^{n \times q}$  DANS  $\mathbb{R}^{m \times p}$  : Soit  $F : X \mapsto F(X)$ , une fonction matricielle différentiable de  $\mathbb{R}^{n \times q}$  dans  $\mathbb{R}^{m \times p}$  :

$$F : X \mapsto F(X) = \begin{pmatrix} f_{11}(X) & \dots & f_{1p}(X) \\ \vdots & \ddots & \vdots \\ f_{m1}(X) & \dots & f_{mp}(X) \end{pmatrix}$$

La dérivée de  $F$  par rapport à  $X$ , ou matrice Jacobienne, est la matrice de taille  $mp \times nq$  :

$$\frac{\partial F}{\partial X} = \frac{\partial \text{vect } F(X)}{\partial \text{vect } X} = \begin{pmatrix} \frac{\partial f_{11}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{11}(X)}{\partial x_{nq}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{m1}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{m1}(X)}{\partial x_{nq}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{1p}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{1p}(X)}{\partial x_{nq}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{mp}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{mp}(X)}{\partial x_{nq}} \end{pmatrix}$$

◇

FONCTION DE GREEN D'UN OPÉRATEUR DIFFÉRENTIEL LINÉAIRE : Soient  $L$  un opérateur différentiel linéaire par rapport à la variable  $\mathbf{x}$  et une fonction  $g(\mathbf{x}, \boldsymbol{\xi})$  tels que :

1. à  $\boldsymbol{\xi}$  fixé,  $g(\mathbf{x}, \boldsymbol{\xi})$  est une fonction de  $\mathbf{x}$
2. les dérivées de  $g(\mathbf{x}, \boldsymbol{\xi})$  par rapport à  $\mathbf{x}$  sont continues, sauf en  $\mathbf{x} = \boldsymbol{\xi}$ ; le nombre de dérivées est fixé par l'ordre de l'opérateur  $L$
3. à  $\boldsymbol{\xi}$  fixé,  $g(\mathbf{x}, \boldsymbol{\xi})$  satisfait l'équation différentielle suivante

$$Lg(\mathbf{x}, \boldsymbol{\xi}) = 0$$

sauf en  $\mathbf{x} = \boldsymbol{\xi}$ , où il y a une singularité. Donc, au sens des distributions, l'équation différentielle se réécrit :

$$Lg(\mathbf{x}, \boldsymbol{\xi}) = \delta(\mathbf{x} - \boldsymbol{\xi})$$

Pour l'opérateur différentiel linéaire  $L$ , cette fonction joue un rôle similaire à celui de la matrice inverse dans une équation matricielle. Elle permet de déterminer une solution à l'équation différentielle suivante :

$$L\varphi(\mathbf{x}) = h(\mathbf{x})$$

où  $h(\mathbf{x})$  est une fonction continue, ou continue par morceaux, de  $\mathbf{x} \in \mathcal{X}$ . Cette solution s'écrit alors :

$$\varphi(\mathbf{x}) = \int_{\mathcal{X}} g(\mathbf{x}, \boldsymbol{\xi}) h(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

◇

ESPACE  $\mathcal{L}^p$  ET NORME ASSOCIÉE : Soit  $(\mathcal{S}, \mu)$  un espace mesuré. L'ensemble des fonctions mesurables  $f : \mathcal{S} \rightarrow \mathbb{R}$  telles que :

$$\|f\|_p \triangleq \left( \int_{\mathcal{S}} |f|^p \mu \right)^{1/p} < \infty,$$

associée à la norme  $\|\cdot\|_p$ , dont la définition est donnée ci dessus, est un espace vectoriel semi-normé appelé espace  $\mathcal{L}^p(\mathcal{S}, \mu)$

◇

NOYAU (EN STATISTIQUES) : Un noyau est une fonction intégrable, positive ou nulle, à valeurs réelles, vérifiant les propriétés suivantes :

1.  $\int_{-\infty}^{+\infty} \mathcal{K}(u) du = 1$
2.  $\mathcal{K}(-u) = \mathcal{K}(u), \quad \forall u$

## Annexe B

# Perceptrons multi-couches

CETTE partie donne un bref aperçu du travail préliminaire qui a été effectué au début de la thèse, sur la modélisation des coefficients aérodynamiques. Des travaux antérieurs réalisés au sein d'Airbus [Lavergne, 2005] ont fait appel à des perceptrons multi-couches pour une telle modélisation. Nous avons donc profité dans un premier temps des outils déjà à disposition afin de les appliquer à notre problématique. Toutefois, les résultats obtenus n'ont pas été satisfaisants et nous avons dû nous tourner vers une autre approche. Nous y consacrons tout de même ce chapitre car ces premiers travaux ont été l'occasion de mettre en évidence la nécessité d'un apprentissage local. Dans un premier temps, nous présentons les perceptrons multi-couches et l'algorithme de rétro-propagation utilisé pour réaliser leur apprentissages. Enfin, nous proposons une extension de cet algorithme dans le cas d'une modélisation boîte grise.

### B.1 Description générale

Ce type d'estimateur non-linéaire est la première solution que nous avons évaluée pour modéliser les coefficients aérodynamiques. Leur structure s'inspire de celle des neurones biologiques [Haykin, 1999] et peuvent être représentés tels qu'en figure B.1.

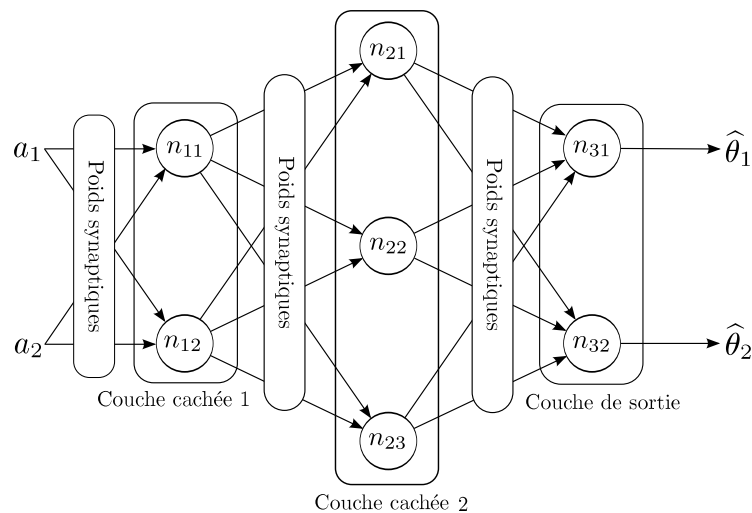


FIG. B.1: Structure d'un perceptron multi-couches

Bien qu'il soit possible d'attribuer à un réseau neuronal plusieurs couches cachées, nous pourrions toujours nous ramener à un perceptron mono-couche en utilisant le théorème d'approximation universelle (cf. théorème 2.2.1 pour le cas des RBF), qui assure qu'un tel réseau peut approcher n'importe quelle application  $\mathcal{C}^\infty$ .

Chaque neurone a une structure qui peut être représentée par la figure B.2.

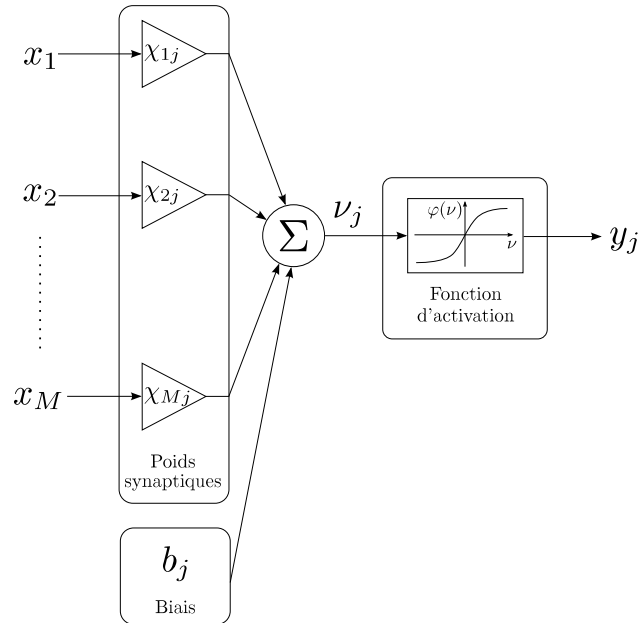


FIG. B.2: Structure d'un neurone

Le réseau est constitué :

- de poids synaptiques donnant un haut degré de connectivité
- d'une fonction d'activation linéaire ou non-linéaire, notée  $\varphi(\nu)$ . Il est important de souligner que la non-linéarité doit être  $\mathcal{C}^\infty$ . Les trois fonctions suivantes sont parmi les plus courantes (cf. figure B.3) :

1. fonction sigmoïde :

$$y_j = \frac{1}{1 + e^{-\nu_j}}$$

ou  $\nu_j$  est le champ local induit<sup>1</sup> du neurone  $j$ , et  $y_j$  est la sortie du neurone

2. une autre fonction non-linéaire de type sigmoïdal, qui peut facilement être implémentée dans un calculateur embarqué :

$$y_j = \frac{\nu_j}{1 + |\nu_j|}$$

La fonction d'activation de la couche de sortie est la fonction identité.

## B.2 Algorithme de rétro-propagation

Pour plus de détails, le lecteur intéressé pourra se référer à [Demuth *et al.*, 2006], [Dreyfus *et al.*, 2004], [Levenberg, 1944] et [Marquardt, 1963]. Cette méthode permet de

<sup>1</sup>C'est la somme pondérée de entrées synaptiques plus le biais

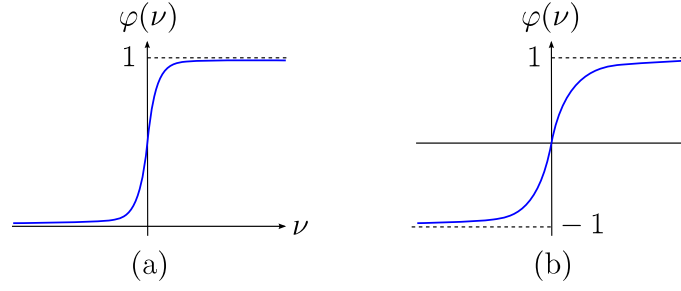


FIG. B.3: (a) Fonction sigmoïde de type 1 - (b) Fonction sigmoïde de type 2

calculer de façon récursive les poids d'un perceptron multi-couches, à partir de l'erreur de sortie courante et selon un critère quadratique sur cette erreur. Considérons un réseau de neurones scalaire à  $R$  couches cachées. Rappelons tout d'abord la relation entrée/sortie du  $j^{\text{ème}}$  neurone de la  $i^{\text{ème}}$  couche cachée :

$$y_j^{(i)} = \varphi^{(i)}(\nu_j^{(i)}), \quad \forall i = 1, \dots, R \text{ et } \forall j = 1, \dots, N^{(i)} \quad (\text{B.1})$$

$$\nu_j^{(i)} = b_j^{(i)} + \sum_{n=1}^{N^{(i-1)}} \chi_{nj}^{(i)} x_n^{(i)} \quad (\text{B.2})$$

$$x_n^{(i)} = y_n^{(i-1)} \quad (\text{B.3})$$

Remarquons que  $(y_1^0 \dots y_M^0)^\top = \mathbf{a}$ , où  $M$  est la taille du vecteur d'entrée.

Considérons le jeu de  $K$  couples entrées/sortie  $(\mathbf{a}_k, \theta_k)$  issus d'une fonction  $f$  que nous souhaitons modéliser à l'aide d'un réseau. Le critère à minimiser est le suivant :

$$\mathcal{J}(\boldsymbol{\chi}) = \sum_{k=1}^K (\theta_k - \hat{f}(\mathbf{a}_k, \boldsymbol{\chi}))^2 = \sum_{k=1}^K \mathcal{J}_k(\boldsymbol{\chi}) \quad (\text{B.4})$$

avec

$$\mathbf{a}_k = (a_{1k} \ a_{2k} \ \dots \ a_{Mk})^\top \quad (\text{B.5})$$

$\boldsymbol{\chi}$  est le vecteur des paramètres du réseau.

Pour calculer le gradient, il suffit de calculer les gradients partiels  $\mathcal{J}_k$  relatifs à la  $k^{\text{ème}}$  observation et de les additionner. La rétro-propagation consiste essentiellement en l'application itérative de la règle des dérivées composées. Nous pouvons écrire pour le  $j^{\text{ème}}$  neurone de la  $i^{\text{ème}}$  couche cachée :

$$\forall n = 1, \dots, N^{(i-1)}, \quad \frac{\partial \mathcal{J}_k}{\partial \chi_{nj}^{(i)}} = \frac{\partial \mathcal{J}_k}{\partial \nu_j^{(i)}} \cdot \frac{\partial \nu_j^{(i)}}{\partial \chi_{nj}^{(i)}} \quad (\text{B.6})$$

$$= \delta_{jk}^{(i)} x_n^{(i)} \quad (\text{B.7})$$

Nous voyons que nous ne devons calculer que les  $\delta_{jk}^{(i)}$ , qui représentent le gradient partiel du  $k^{\text{ème}}$  sous-critère par rapport au  $j^{\text{ème}}$  champ local induit de la  $i^{\text{ème}}$  couche cachée. Nous



pouvons les calculer de façon récursive assez facilement en *allant de la sortie vers les entrées*<sup>2</sup>.

Nous avons :

- pour le neurone de sortie ( $i = R, N^{(i)} = 1, j = 1$ ) :

$$\delta_{jk}^{(R)} = \frac{\partial \mathcal{J}_k}{\partial \nu_j^{(R)}} \quad (\text{B.8})$$

$$= \frac{\partial}{\partial \nu_j^{(R)}} \left[ \left( y_k - \varphi^{(R)} \left( \nu_{jk}^{(R)} \right) \right)^2 \right] \quad (\text{B.9})$$

$$= -2 \varphi^{(R)'} \left( \nu_{jk}^{(R)} \right) \left( y_k - \varphi^{(R)} \left( \nu_{jk}^{(R)} \right) \right) \quad (\text{B.10})$$

Or,  $\varphi^{(R)}$  est la fonction identité. Donc nous pouvons écrire :

$$\delta_{jk}^{(R)} = -2 \left( y_k - \varphi^{(R)} \left( \nu_{jk}^{(R)} \right) \right) \quad (\text{B.11})$$

- pour le  $j^{\text{ème}}$  neurone de la  $i^{\text{ème}}$  couche cachée ( $i < R$ ) :

$$\delta_{jk}^{(i)} = \frac{\partial \mathcal{J}_k}{\partial \nu_j^{(i)}} \quad (\text{B.12})$$

$$= \frac{\partial \mathcal{J}_k}{\partial \nu^{(i+1)}} \times \frac{\partial \nu^{(i+1)}}{\partial \nu_j^{(i)}} \quad (\text{B.13})$$

$$= \left( \frac{\partial \mathcal{J}_k}{\partial \nu_1^{(i+1)}} \quad \cdots \quad \frac{\partial \mathcal{J}_k}{\partial \nu_{N^{(i+1)}}^{(i+1)}} \right) \times \begin{pmatrix} \frac{\partial \nu_1^{(i+1)}}{\partial \nu_j^{(i)}} \\ \vdots \\ \frac{\partial \nu_{N^{(i+1)}}^{(i+1)}}{\partial \nu_j^{(i)}} \end{pmatrix} \quad (\text{B.14})$$

$$= \left( \delta_{1k}^{(i+1)} \quad \cdots \quad \delta_{N^{(i+1)}k}^{(i+1)} \right) \times \begin{pmatrix} \chi_{j1}^{(i+1)} \\ \vdots \\ \chi_{jN^{(i+1)}}^{(i+1)} \end{pmatrix} \varphi^{(i)'} \left( \nu_{jk}^{(i)} \right) \quad (\text{B.15})$$

$$= \varphi^{(i)'} \left( \nu_{jk}^{(i)} \right) \sum_{m=1}^{N^{(i+1)}} \delta_{mk}^{(i+1)} \chi_{jm}^{(i+1)} \quad (\text{B.16})$$

Nous avons donc exprimé les  $\delta_{jk}^{(i)}$  en fonction de ceux de la couche cachée suivante.

Nous pouvons ainsi itérer le calcul jusqu'à la couche d'entrée et exprimer le gradient de  $\mathcal{J}_k$ . Le calcul des gradients par rapport aux biais  $b_j^{(i)}$  s'effectue de la même manière.

Pour résumer, le calcul des poids synaptiques par rétro-propagation est composé de deux étapes :

1. Une phase de *propagation* où l'on calcule la sortie du réseau
2. Une phase de *rétro-propagation* où l'on calcule les  $\delta_i^k$  en allant de la sortie vers les entrées

<sup>2</sup>d'où le terme de *rétro-propagation*

## B.3 Généralisation

Nous proposons d'effectuer l'apprentissage de plusieurs réseaux de neurones en parallèle, avec un critère d'erreur sur une fonction des sorties de ces réseaux. Nous nous plaçons dans un cas de modélisation boîte grise.

### B.3.1 Définitions

Soit  $L$  réseaux à une couche cachée, dont la fonction est identique d'un réseau à l'autre et à couche de sortie linéaire. Nous disposons d'un ensemble d'apprentissage de  $K$  éléments  $(\mathbf{a}_k, y_k)$ .

Soit  $l = 1 \dots L$ . Pour le  $l^{\text{ème}}$  réseau, nous avons :

- $M_l$  entrées  $x_{i_l}$  avec  $i_l = 1 \dots M_l$
- $N_l$  neurones dans la couche cachée, indicés par  $j_l = 1 \dots N_l$
- $P_l = 1$  sortie<sup>3</sup>  $y_l$
- $\chi_{i_l j_l}$  le poids synaptique affectant la connexion de l'entrée  $x_{i_l}$  vers le  $j_l^{\text{ème}}$  neurone de la couche cachée
- $b_{j_l}$  le biais du  $j_l^{\text{ème}}$  neurone de la couche cachée
- $\omega_{j_l}$  le poids synaptique affectant la connexion entre le  $j_l^{\text{ème}}$  neurone et la sortie  $y_l$
- $\delta_l$  le biais du neurone de sortie

Définissons le critère  $\mathcal{J}$  et l'erreur  $\varepsilon$  :

$$\mathcal{J} = \sum_{k=1}^K \left( g(\hat{\theta}_{1k}, \dots, \hat{\theta}_{Lk}) - y_k \right)^2 = \sum_{k=1}^K \mathcal{J}_k \quad (\text{B.17})$$

$$\varepsilon_k = g(\hat{\theta}_{1k}, \dots, \hat{\theta}_{Lk}) - y_k \quad (\text{B.18})$$

où  $f$  est une fonction différentiable.

**Matrice Jacobienne de  $\varepsilon$**  Soit  $\nu_{j_l}$  le champ local induit du  $j_l^{\text{ème}}$  neurone et  $\mu_l$  le champ local induit du neurone de sortie  $y_l$ . Nous modifions les notations par rapport au paragraphe précédent afin de ne pas alourdir le formalisme déjà chargé.

$$\nu_{j_l} = b_{j_l} + \sum_{i=1}^{M_l} w_{i j_l} a_i \quad (\text{B.19})$$

$$\mu_l = \delta_l + \sum_{j=1}^{N_l} \omega_j \varphi(\nu_j) \hat{\theta}_l = \mu_l \quad (\text{B.20})$$

Pour le  $k^{\text{ème}}$  exemple :

1. Dérivée par rapport aux poids de la couche de sortie

$$\left( \frac{\partial \varepsilon_k}{\partial \omega_{j_l}} \right)_k = \left( \frac{\partial g}{\partial \hat{\theta}_l} \right)_k \times \varphi(\nu_{j_l k}) \quad (\text{B.21})$$

2. Dérivée par rapport aux poids de la couche cachée

$$\left( \frac{\partial \varepsilon_k}{\partial \chi_{i_n j_n}} \right)_k = \left( \frac{\partial g}{\partial \hat{\theta}_l} \right)_k \times \omega_{j_l} \varphi'(\nu_{j_l k}) \times a_{i_l k} \quad (\text{B.22})$$

---

<sup>3</sup>Pour des raisons de clarté, nous ne traiterons que le cas mono-sortie et ce, sans perte de généralités

3. Dérivée par rapport au biais de la couche de sortie

$$\left(\frac{\partial \varepsilon_k}{\partial \delta_l}\right)_k = \left(\frac{\partial g}{\partial \widehat{\theta}_l}\right)_k \quad (\text{B.23})$$

4. Dérivée par rapport aux biais de la couche cachée

$$\left(\frac{\partial \varepsilon_k}{\partial b_{ji}}\right)_k = \left(\frac{\partial g}{\partial \widehat{\theta}_l}\right)_k \times \omega_{ji} \varphi'(\nu_{jlk}) \quad (\text{B.24})$$

## B.4 Conclusion

Ces réseaux ont été testés sur l'estimateur de dérapage (chapitre 7, p. 105). Les résultats obtenus n'ont pas été satisfaisants : bien que les réseaux aient été définis à l'aide des modèles donnés au paragraphe §6.3.3, ils n'arrivaient pas à atteindre une performance acceptable, et plus particulièrement le  $\Delta Cy_{\delta p}$ , même en augmentant le nombre de neurones, ce qui est supposé améliorer la précision du réseau. L'utilisation de techniques telle que la méthode du descripteur sonde [Dreyfus *et al.*, 2004], censée identifier les entrées pertinentes, n'a pas apporté d'informations.

De plus, la mise en place d'un algorithme temps-réel a été entravée par la structure même de ces réseaux de neurones et le type d'algorithme d'apprentissage utilisé. En effet, l'algorithme de rétro-propagation est de type global : il ne peut être utilisé que sur des ensembles d'apprentissage pré-déterminés. La difficulté de la résolution est que les réseaux de neurones sont des fonctions non-linéaires en leur paramètres, ce qui complique le calcul d'une solution théorique. Malgré l'utilisation de solvers non-linéaires contraints, la performance finale n'était pas satisfaisante. Cette structure s'avère donc inappropriée à la résolution du problème. Cependant, il existe d'autres outils dont la structure pourrait s'avérer davantage adaptée.

## Annexe C

# Inversion et synthèse simultanées

Après avoir mis en place l'algorithme de Landweber dans le cadre de la compensation des spoilers, nous avons cherché à fusionner les étapes d'inversion et de synthèse en une seule pour la structure de compensation existante, i.e. des gains statiques interpolés.

Dans le cas mono-sortie, qui n'est pas conservatif, nous cherchons à minimiser le critère suivant :

$$\mathcal{J}_\lambda(\mathbf{e}, \mathbf{K}) = \|\mathbf{z} - G\mathbf{e}\|^2 + \lambda^2 \|D\mathbf{e}\|^2 + \|F\mathbf{e} - W\mathbf{K}\|^2 \quad (\text{C.1})$$

$$= \left\| \begin{pmatrix} \mathbf{z} \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} G & 0 \\ -\lambda D & 0 \\ -F & W \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ \mathbf{K} \end{pmatrix} \right\|^2 \quad (\text{C.2})$$

Nous choisissons d'avoir deux pas d'adaptation distincts  $\alpha_e$  et  $\alpha_K$  pour le calcul des solutions  $\tilde{\mathbf{e}}^*$  et  $\tilde{\mathbf{K}}^*$  afin de n'être limité que par le plus faible des deux. Nous trouvons alors un algorithme semblable à celui proposé au paragraphe §3.2.1.2 :

$$\mathbf{e}_0 = 0 \quad (\text{C.3})$$

$$\mathbf{K}_0 = 0 \quad (\text{C.4})$$

$$\mathbf{e}_{n+1} = \mathbf{e}_n + \alpha_e \left( G^\top Z - (G^\top G + \lambda^2 D^\top D) \mathbf{e}_n \right) - F^\top \alpha_K (F\mathbf{e}_n - W\mathbf{K}_n) \quad (\text{C.5})$$

$$\mathbf{K}_{n+1} = \mathbf{K}_n + \alpha_K W^\top (F\mathbf{e}_n - W\mathbf{K}_n) \quad (\text{C.6})$$

Il faut à présent déterminer des conditions sur les pas  $\alpha_e$  et  $\alpha_K$  afin d'assurer la convergence et la stabilité de l'algorithme, ce qui somme toute, revient au même. Pour cela, nous procédons comme au paragraphe §3.2.1.2 où nous étudions l'évolution de l'erreur à l'itération  $n$  par rapport à la solution du problème. Soient  $\mathbf{e}^*$  et  $\mathbf{K}^*$  tels que :

$$(G^\top G + \lambda^2 D^\top D) \mathbf{e}^* = G^\top \mathbf{z} \quad (\text{C.7})$$

$$W\mathbf{K}^* = F\mathbf{e}^* \quad (\text{C.8})$$

Posons  $\Delta\mathbf{e}_n = \mathbf{e}_n - \mathbf{e}^*$  et  $\Delta\mathbf{K}_n = \mathbf{K}_n - \mathbf{K}^*$ . En substituant dans (C.5) et (C.6), nous obtenons :

$$\Delta\mathbf{e}_{n+1} = \Delta\mathbf{e}_n - \alpha_e (G^\top G + \lambda^2 D^\top D) \Delta\mathbf{e}_n - \alpha_K F^\top (F\Delta\mathbf{e}_n - W\Delta\mathbf{K}_n) \quad (\text{C.9})$$

$$\Delta\mathbf{K}_{n+1} = \Delta\mathbf{K}_n + \alpha_K W^\top (F\Delta\mathbf{e}_n - W\Delta\mathbf{K}_n) \quad (\text{C.10})$$

$$\iff \begin{pmatrix} \Delta \mathbf{e}_{n+1} \\ \Delta \mathbf{K}_{n+1} \end{pmatrix} = (I - \alpha_e G_e - \alpha_K G_K) \begin{pmatrix} \Delta \mathbf{e}_n \\ \Delta \mathbf{K}_n \end{pmatrix} \quad (\text{C.11})$$

avec

$$G_e = \begin{pmatrix} G^\top G + \lambda^2 D^\top D & 0 \\ 0 & 0 \end{pmatrix} \geq 0 \quad (\text{C.12})$$

$$G_K = \begin{pmatrix} F^\top F & -F^\top W \\ -W^\top F & W^\top W \end{pmatrix} \geq 0 \quad (\text{C.13})$$

Comme cela a déjà été fait auparavant, il faut déterminer  $\alpha_e$  et  $\alpha_K$  tels que :

$$\|I - \alpha_e G_e - \alpha_K G_K\|_2 < 1 \quad (\text{C.14})$$

Notons  $\Gamma = \alpha_e G_e + \alpha_K G_K$ .  $G_e$  et  $G_K$  étant symétriques, le théorème de Weyl nous donne les inégalités suivantes :

$$\alpha_e \lambda_1(G_e) + \alpha_K \lambda_k(G_K) \leq \lambda_k(\Gamma) \leq \alpha_e \lambda_k(G_e) + \alpha_K \lambda_N(G_K) \quad (\text{C.15})$$

où les  $\lambda_k(\Gamma)$  sont les valeurs propres de  $\Gamma$ , rangées par ordre croissant pour  $k = 1, \dots, N$ .

De plus, nous avons :

$$\|I - \Gamma\|_2 < 1 \iff |\lambda_k(I - \Gamma)| < 1, \quad \forall k = 1, \dots, N \quad (\text{C.16})$$

$$\lambda_k(I - \Gamma) = 1 - \lambda_k(\Gamma), \quad \forall k = 1, \dots, N \quad (\text{C.17})$$

Donc :

$$\|I - \Gamma\|_2 < 1 \iff 0 < \lambda_k(\Gamma) < 2, \quad \forall k = 1, \dots, N \quad (\text{C.18})$$

$$\iff \begin{cases} 0 < \lambda_1(\Gamma) \\ \lambda_N(\Gamma) < 2 \end{cases} \quad (\text{C.19})$$

A l'aide de (C.15), nous pouvons substituer les valeurs de  $\lambda_1(\Gamma)$  et  $\lambda_N(\Gamma)$  dans (C.19) et déduire une condition suffisante de convergence de l'algorithme si  $\alpha_e$  et  $\alpha_K$  vérifient les inégalités suivantes :

$$-\alpha_e \lambda_1(G_e) - \alpha_K \lambda_1(G_K) < 0 \quad (\text{C.20})$$

$$\alpha_e \lambda_N(G_e) + \alpha_K \lambda_N(G_K) < 2 \quad (\text{C.21})$$

$$\alpha_e > 0 \quad (\text{C.22})$$

$$\alpha_K > 0 \quad (\text{C.23})$$

Or, de par la forme de  $G_e$  donnée en (C.12), nous avons  $\lambda_1(G_e) = 0$ . Donc si (C.23) est vérifiée, (C.20) est également vérifiée. Nous pouvons alors proposer des solutions de la forme :

$$\alpha_e = \frac{1 - \epsilon}{\lambda_N(G_e)} \quad \alpha_K = \frac{1 - \epsilon}{\lambda_N(G_K)} \quad (\text{C.24})$$

ou encore

$$\alpha_e = \frac{1/2}{\lambda_1(G_e) + \lambda_N(G_e)} \quad \alpha_K = \frac{1/2}{\lambda_N(G_K) + \lambda_N(G_K)} \quad (\text{C.25})$$

Cependant, la mise en place de la méthode ne s'est pas avérée concluante dans la mesure où les deux sous-problèmes convergeaient avec des constantes de temps très différentes.

## Annexe D

# Vérification de la propriété de la matrice d'observation du filtre en treillis normalisé

Rappelons tout d'abord quelques notations nécessaires à la vérification. Un filtre en treillis normalisé peut s'exprimer sous la forme d'état suivante :

$$\hat{u}(k) = \mathbf{h}(k)^\top \begin{pmatrix} \mathbf{x}(k+1) \\ s(k) \end{pmatrix} \quad (\text{D.1})$$

$$\begin{pmatrix} \mathbf{x}(k+1) \\ s(k) \end{pmatrix} = Q(k) \begin{pmatrix} \mathbf{x}(k) \\ w(k) \end{pmatrix} \quad (\text{D.2})$$

$$Q(k) = Q_1(k) \times \dots \times Q_N(k), \quad (\text{D.3})$$

où  $w(k)$  est l'entrée du filtre,  $\hat{u}(k)$  la sortie du filtre et  $s(k)$  un état factice et avec

$$Q_n(k) = \begin{pmatrix} I_{n-1} & 0 & 0 \\ 0 & -\sin \phi_n(k) & \cos \phi_n(k) & 0 \\ 0 & \cos \phi_n(k) & \sin \phi_n(k) & 0 \\ 0 & 0 & 0 & I_{N-n} \end{pmatrix}, \quad \forall n = 1, \dots, N$$

$$\mathbf{h}(k)^\top = (\nu_0(k) \quad \dots \quad \nu_N(k)).$$

$Q(k)$  étant le produit de  $N$  matrices de rotations de Givens, elle est orthogonale [Golub et Van Loan, 1996]. Remarquons de plus que la matrice  $Q(k)$  est une matrice de *Hessenberg* supérieure, c'est-à-dire une matrice presque triangulaire dont tous les termes en dessous de la première sous-diagonale sont nuls. Quelques calculs directs pour des valeurs de  $N$  raisonnables nous permettent de vérifier que les éléments de cette sous-diagonale sont les  $\cos \phi_n(k)$ , tels que :

$$Q(k) = \begin{pmatrix} \times & \dots & \dots & \dots & \dots & \times \\ \cos \phi_1(k) & \times & & & & \vdots \\ 0 & \cos \phi_2(k) & \ddots & & & \vdots \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \times & \vdots \\ 0 & \dots & \dots & 0 & \cos \phi_N(k) & \times \end{pmatrix} \quad (\text{D.4})$$

Rappelons à présent la propriété à vérifier.

PROPRIÉTÉ : Soit  $Q_o(k + N, k)$  la matrice d'observation du filtre définie comme suit :

$$Q_o(k + N, k) \triangleq \begin{pmatrix} \mathbf{q}_{21}(k)^\top \\ \mathbf{q}_{21}(k+1)^\top Q_{11}(k) \\ \vdots \\ \mathbf{q}_{21}(k+\tau-1)^\top Q_{11}(k+\tau-2) \times \dots \times Q_{11}(k) \end{pmatrix}$$

avec

$$Q(k) = \begin{pmatrix} Q_{11}(k) & \mathbf{q}_{12}(k) \\ \mathbf{q}_{21}(k)^\top & q_{22}(k) \end{pmatrix}$$

. Le calcul direct de  $Q_o(k + N, k)$  pour laquelle les lignes ont été permutées, met en évidence une matrice triangulaire dont les termes diagonaux sont :

$$q_n^o = \prod_{i=1}^n \cos \phi_{N+1-i}(k + n - i), \quad \forall n = 1, \dots, N \quad (\text{D.5})$$

*Vérification.* La forme particulière de la matrice  $Q(k)$  va nous permettre de rendre cette vérification plus simple qu'il n'y paraît. En effet, il nous suffit de connaître les termes de la sous-diagonale de  $Q(k)$ , qui sont les  $\cos \phi_n(k)$ . D'après (D.4), nous pouvons déduire :

$$Q_{11}(k) = \begin{pmatrix} \times & \dots & \dots & \dots & \dots & \times \\ \cos \phi_1(k) & \times & & & & \vdots \\ 0 & \cos \phi_2(k) & \ddots & & & \vdots \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \times & \vdots \\ 0 & \dots & \dots & 0 & \cos \phi_{N-1}(k) & \times \end{pmatrix}$$

$$\mathbf{q}_{21}(k)^\top = (0 \quad \dots \quad 0 \quad \cos \phi_N(k))$$

Le vecteur  $\mathbf{q}_{21}(k)^\top$  nous donne déjà le première ligne de la matrice  $Q_o(k + N, k)$ . Calculons à présent la deuxième ligne :

$$\begin{aligned} & \mathbf{q}_{21}(k+1)^\top Q_{11}(k) \\ &= (0 \quad \dots \quad 0 \quad \cos \phi_N(k+1)) \begin{pmatrix} \times & \dots & \dots & \dots & \dots & \times \\ \cos \phi_1(k) & \times & & & & \vdots \\ 0 & \cos \phi_2(k) & \ddots & & & \vdots \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \times & \vdots \\ 0 & \dots & \dots & 0 & \cos \phi_{N-1}(k) & \times \end{pmatrix} \\ &= (0 \quad \dots \quad 0 \quad \cos \phi_N(k+1) \cos \phi_{N-1}(k) \quad \times) \end{aligned}$$

Nous pouvons d'ores et déjà voir apparaître la récurrence que nous allons confirmer à l'aide

de la troisième ligne :

$$\begin{aligned}
& \mathbf{q}_{21}(k+2)^\top Q_{11}(k+1)Q_{11}(k) \\
&= (0 \quad \dots \quad 0 \quad \cos \phi_N(k+2) \cos \phi_{N-1}(k+1) \quad \times) \\
& \left( \begin{array}{cccccc}
\times & \dots & \dots & \dots & \dots & \times \\
\cos \phi_1(k) & \times & & & & \vdots \\
0 & \cos \phi_2(k) & \ddots & & & \vdots \\
\vdots & 0 & \ddots & \ddots & & \vdots \\
\vdots & & \ddots & \cos \phi_{N-2}(k) & \times & \vdots \\
0 & \dots & \dots & 0 & \cos \phi_{N-1}(k) & \times
\end{array} \right) \\
&= (0 \quad \dots \quad 0 \quad \cos \phi_N(k+2) \cos \phi_{N-1}(k+1) \cos \phi_{N-2}(k) \quad \times \quad \times)
\end{aligned}$$

En répétant ce calcul récurrent, nous pouvons conclure que la matrice  $Q_o(k+N, k)$  est une matrice « anti-triangulaire » inférieure telle que :

$$Q_o(k+N, k) = \begin{pmatrix} 0 & \dots & 0 & q_1^o(k) \\ \vdots & / & / & \times \\ 0 & / & / & \vdots \\ q_N^o(k) & \times & \dots & \times \end{pmatrix}$$

avec

$$q_n^o = \prod_{i=1}^n \cos \phi_{N+1-i}(k+n-i), \quad \forall n = 1, \dots, N$$

La propriété est donc vérifiée par permutation des lignes.  $\square$





# Bibliographie

- [Akçay et Ninness, 1998a] Hüseyin Akçay et Brett Ninness. Orthonormal basis functions for modelling continuous-time systems. *Signal Processing*, 34(9) :1101–1117, 1998.
- [Akçay et Ninness, 1998b] Hüseyin Akçay et Brett Ninness. Rational basis functions for robust identification from frequency and time domain measurements. *Automatica*, 77(1) :261–274, 1998.
- [Aström et Wittenmark, 1995] Karl J. Aström et Björn Wittenmark. *Adaptive Control*. Addison-Wesley, 2<sup>ème</sup> édition, 1995.
- [Atkeson *et al.*, 1997] Chris Atkeson, Andrew Moore, et Stefan Schaal. Locally weighted learning. *AI Review*, 11 :11–73, April 1997.
- [Avanzini *et al.*, 1998] Giulio Avanzini, Guido de Matteis, et Luciano M. de Socio. Analysis of aircraft agility on maximum performance maneuvers. *Journal of Aircraft*, 35(4) :529–535, 1998.
- [Björck, 1996] Åke Björck. *Numerical Methods for Least Squares Problems*. S.I.A.M., 1<sup>ère</sup> édition, 1996.
- [Boiffier, 1998] Jean-Luc Boiffier. *The Dynamics of Flight, The Equations*. Wiley, 1998.
- [Boiffier, 2001] Jean-Luc Boiffier. *Dynamique du vol de l'avion*. Supaéro, 2001.
- [Bottou et Vapnik, 1996] Léon Bottou et Vladimir Vapnik. Local learning algorithms. Note technique, AT&T Bell Laboratories, Septembre 1996.
- [Boyle et Chamitoff, 1999] David P. Boyle et Gregory E. Chamitoff. Autonomous maneuver tracking for self-piloted vehicles. *Journal of Guidance, Control and Dynamics*, 22(1) :58–67, 1999.
- [Broomhead et Lowe, 1988] Dave S. Broomhead et Daniel Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2 :321–355, 1988.
- [Bucharles et Gimonet, 2000] Alain Bucharles et Bernard Gimonet. *Identification des systèmes*. Supaéro, 2000.
- [Cao et Schwartz, 1999] Liyu Cao et Howard M. Schwartz. A directionnal forgetting algorithm based on the decomposition of the information matrix. Dans *7<sup>th</sup> Mediterranean Conference on Control and Automation*, Haifa, Israel, 28–30 Juin 1999.
- [Cao et Schwartz, 2000] Liyu Cao et Howard M. Schwartz. A directionnal forgetting algorithm based on the decomposition of the information matrix. *Automatica*, 36 :1725–1731, 2000.
- [Chai *et al.*, 1996] Joo-Siang Chai, Shaohua Tan, et Chang-Chieh Hang. Gain-scheduling control of nonlinear plant using RBF neural networks. Dans *IEEE Symposium on Intelligent Control*, Dearborn (MI), U.S.A., 15–18 Septembre 1996.

- [Craven et Wahba, 1979] Peter Craven et Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31 :377–403, 1979.
- [Cumer et Mouyon, 2003] Christelle Cumer et Philippe Mouyon. Approche prabi en ajustement de correcteurs. poste 7 - tâche a12.5. Note technique, ONERA, Octobre 2003.
- [De Cock et De Moor, 2003] Katrien De Cock et Bart De Moor. *Subspace identification methods*, chapitre « Control systems robotics and automation », pages 933–979. Eolss Publishers, 2003.
- [de Larminat, 1996] Philippe de Larminat. *Automatique*. Hermès Science, 2<sup>ème</sup> édition, 1996.
- [Demuth *et al.*, 2006] Howard Demuth, Mark Beale, et Martin Hagan. *Neural Network Toolbox, User's guide*. The Mathworks, 2006.
- [Dreyfus *et al.*, 2004] Gérard Dreyfus, Jean-Marc Martinez, Manuel Samuelides, Mirta Gordon, Fouad Badran, Sylvie Thiria, et Laurent Héroult. *Réseaux de neurones, méthodologies et applications*. Eyrolles, 2<sup>ème</sup> édition, 2004.
- [Egger, 2005] Herbert Egger. Accelerated Newton-Landweber iterations for regularizing nonlinear inverse problem. Note technique, Johann Radon Institute for Computational and Applied Mathematics, January 2005.
- [Fleury, 2001] Gilles Fleury. *Analyse Spectrale – Méthodes non-paramétriques et paramétriques*. Ellipses Marketing, 1<sup>ère</sup> édition, 2001.
- [Franklin, 2000] Joel N. Franklin. *Matrix Theory*. Dover Publications, 2000.
- [Ghahramani, 2004] Zoubin Ghahramani. Unsupervised learning, 2004.
- [Gimonet, 2003] Bernard Gimonet. Ajustement de correcteur après essai en vol. Note technique, ONERA, Septembre 2003.
- [Golub et Van Loan, 1996] Gene H. Golub et Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3<sup>ème</sup> édition, 1996.
- [Gray, 2006] Robert M. Gray. Toeplitz and circulant matrices : A review. *Foundations and Trends in Communications and Information Theory*, 2(3) :155–239, 2006.
- [Grünhagen, 1993] Wolfgang Von Grünhagen. Inverse simulation : A tool for the validation of simulation programs – first results. *Zeitschrift für Flugwissenschaften und Weltraumforschung*, 17 :211–219, 1993.
- [Guermeur et Paugam-Moisy, 1999] Yann Guermeur et Hélène Paugam-Moisy. *Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines*, pages 109–138. Hermès, 1999.
- [Hanke, 1991] Martin Hanke. Accelerated Landweber iterations for the solution of ill-posed equations. *Numerische Mathematik*, 60(1) :341–373, 1991.
- [Haykin, 1999] Simon Haykin. *Neural Networks, a comprehensive foundation*. Prentice-Hall, 2<sup>ème</sup> édition, 1999.
- [Hizal, 1999] Nafiz Aydın Hizal. Gain scheduling adaptive model control. *Turkish Journal of Engineering and Environmental Science*, 23(3) :209–218, 1999.
- [Jacobs *et al.*, 1990] Robert A. Jacobs, Michael I. Jordan, et Andrew G. Barto. Task decomposition through competition in a modular connectionist. Note technique, University of Massachusetts, Amherst, MA, USA, 1990.
- [Jacobs *et al.*, 1991] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, et Geoffrey E. Hinton. Adaptive mixture of experts. *Neural Computation*, 3 :79–87, 1991.

- [Jacobs et Jordan, 1990] Robert A. Jacobs et Michael I. Jordan. A competitive modular connectionist architecture. Dans *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems*, pages 767–773, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.
- [Jean, 2001] Sébastien Jean. Restitution d'une perturbation en vitesse de roulis par inversion semi-aveugle. Thèse de master, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, 2001.
- [Jordan et Jacobs, 1994] Michael I. Jordan et Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6 :181–214, 1994.
- [Kulhavy et Kárny, 1984] R. Kulhavy et M. Kárny. Tracking of slowly varying parameters by directional forgetting. *Preprints of the 9th IFAC World Congress*, 10 :78–83, 1984.
- [Landau, 2002] Ioan Doré Landau. *Commande des systèmes – Conception, identification et mise en œuvre*. Hermès-Lavoisier, 1<sup>ère</sup> édition, 2002.
- [Landweber, 1951] Louis Landweber. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 73 :615–624, 1951.
- [Lavergne, 2005] Fabien Lavergne. *Méthodologies de synthèse de lois de commande non-linéaires et robustes : application au suivi de trajectoire des avions de transport civil*. Thèse de doctorat, Université Toulouse III, 2005.
- [Levenberg, 1944] Kenneth Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quarterly of Applied Mathematics*, 2 :164–168, 1944.
- [Magni, 2002] Jean-Francois Magni. *Robust Modal Control with a Toolbox for use with MATLAB<sup>®</sup>*. Kluwer Academic / Plenum Publishers, 2002.
- [Marquardt, 1963] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11 :431–441, 1963.
- [Michelli, 1986] Charles A. Michelli. Interpolation of scattered data : Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2 :11–22, 1986.
- [Moody et Darken, 1989] John Moody et Christian J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1(2) :281–294, 1989.
- [Mouyon *et al.*, 2002] Philippe Mouyon, Christelle Cumer, et Yannick Losser. Retouche de correcteurs. Dans *Conférence Internationale Francophone d'Automatique*, Nantes, France, 8–10 Juillet 2002.
- [Mouyon et Losser, 2002] Philippe Mouyon et Yannick Losser. An autotuned deconvolution algorithm with applications to aircraft landing tests matching and autopilot retuning. Dans *41<sup>st</sup> IEEE Conference on Decision and Control*, Las Vegas, NV, U.S.A., 10–13 Décembre 2002.
- [Mouyon et Vacher, 2001] Philippe Mouyon et Pierre Vacher. Unknown input recovery for closed loop system with saturations. Dans *European Control Conference*, Porto, Portugal, 4–7 Septembre 2001.
- [Nikolakopoulos *et al.*, 2007] George Nikolakopoulos, Anthony Tzes, et Yannis Koveos. Application of adaptive lattice filters to internal model control. *International Journal of Adaptive Control and Signal Processing*, 21(10) :837–863, Septembre 2007.

- [O’Leary et Simmons, 1981] Dianne P. O’Leary et John A. Simmons. A bidiagonalization-regularization procedure for large scale discretization of ill-posed problems. *SIAM Journal on Scientific and Statistical Computing*, 2 :474–489, 1981.
- [Overschee et Moor, 1996] Peter Van Overschee et Bart De Moor. *Subspace Identification of Linear Systems : Theory, Implementation, Applications*. Kluwer Academic, 1996.
- [Park et Sandberg, 1991] Jooyoung Park et Irwin W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3 :246–257, 1991.
- [Poggio et Girosi, 1989] Tomaso Poggio et Federico Girosi. A theory of networks for approximation and learning. Note technique AIM-1140, Massachusetts Institute of Technology, Artificial Intelligence Laboratory – Whitaker College, Center for Biological Information Processing, 1989.
- [Poggio et Girosi, 1990] Tomaso Poggio et Federico Girosi. Networks for approximation and learning. Dans *Proceedings of the IEEE*, volume 78, pages 1481–1497, 1990.
- [Powell, 1988] Michael J. D. Powell. Radial basis function approximations to polynomials. *Numerical analysis*, pages 223–241, 1988.
- [Regalia, 1992] Philip A. Regalia. Stable and efficient lattice algorithms for adaptive IIR filtering. *IEEE Transactions on Signal Processing*, 40(2), Septembre 1992.
- [Regalia, 1994] Philip A. Regalia. *Adaptive IIR Filtering in Signal Processing and Control*. CRC Press, 1<sup>ère</sup> édition, 1994.
- [Ronceray *et al.*, 2007a] Lilian Ronceray, Matthieu Jeanneau, Daniel Alazard, Philippe Mouyon, et Sihem Tebbani. A local learning approach to real-time parameter estimation - application to an aircraft. Dans *4<sup>th</sup> INSTICC International Conference on Informatics in Control, Automation and Robotics*, Angers, France, 9–12 Mai 2007.
- [Ronceray *et al.*, 2007b] Lilian Ronceray, Matthieu Jeanneau, Daniel Alazard, Philippe Mouyon, et Sihem Tebbani. Real-time automatic tuning of an aircraft sideslip estimator. Dans *17<sup>th</sup> IFAC Symposium on Automatic Control in Aerospace*, Toulouse, France, 25–29 Juin 2007.
- [Ronceray *et al.*, 2008] Lilian Ronceray, Daniel Alazard, Philippe Mouyon, Guilhem Puyou, et Sihem Tebbani. Aircraft spoilers compensation design using iterative inversion. Dans *17<sup>th</sup> IFAC World Congress*, Seoul, Korea, 7–11 Juillet 2008.
- [Ronceray *et al.*, 2009] Lilian Ronceray, Philippe Mouyon, Sihem Tebbani, Guilhem Puyou, et Daniel Alazard. Adaptive retuning of feedforward controller – Application to the airbrake compensation of an aircraft. Dans *Soumis à European Control Conference*, 2009.
- [Ronceray, 2005] Lilian Ronceray. Contrôle adaptatif du mode de flexion voilure. Thèse de master, Ecole Supérieure d’Electricité, 2005.
- [Rugh, 1991] Wilson J. Rugh. Analytical framework for gain-scheduling. *IEEE Control System Magazine*, 11(1) :79–84, 1991.
- [Sentoh et Bryson, 1992] Etsuroh Sentoh et Arthur E. Bryson. Inverse and optimal control for desired outputs. *Journal of Guidance, Control and Dynamics*, 15(3) :687–691, 1992.
- [Seren et Hardier, 2008] Cédric Seren et Georges Hardier. CVF – lot 1 : Estimation dynamique embarquée de paramètres (de type masse/centrage). Note technique RTS 16/11970, ONERA/DCSD, Juillet 2008.

- [Silverman et Anderson, 1968] Leonard M. Silverman et Brian D. O. Anderson. Controllability, observability and stability of linear systems. *SIAM Journal of Control*, 6 :121–130, 1968.
- [Tikhonov et Arsenin, 1977] Andrey N. Tikhonov et Vasilii Y. Arsenin. *Solutions of Ill-posed Problems*. Winston, 1<sup>ère</sup> édition, 1977.
- [Vapnik *et al.*, 1997] Vladimir Vapnik, Steven Golowich, et Alex Smola. Support vector method for function approximation, regression estimation, and signal processing. *Neural Information Processing Systems*, 9 :11–73, 1997.
- [Vapnik et Chervonenkis, 1971] Vladimir Vapnik et Alexei Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2) :264–280, 1971.
- [Vapnik, 1982] Vladimir Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, 1<sup>ère</sup> édition, 1982.
- [Vapnik, 1991] Vladimir Vapnik. Principles of risk minimization for learning theory. Dans *Advances in Neural Information Processing Systems*, volume 4, pages 831–838. Morgan Kaufman, 1991.
- [Widrow et Plett, 1996] Bernard Widrow et Gregory L. Plett. Adaptive inverse control based on linear and nonlinear adaptive filtering. Dans *IEEE Conference on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, Venice, Italy, 21–23 Août 1996.
- [Widrow et Walach, 1995] Bernard Widrow et Eugene Walach. *Adaptive Inverse Control*. Prentice-Hall, 1995.
- [Wildschek *et al.*, 2006] Andreas Wildschek, Rudolf Maier, Falk Hoffmann, Matthieu Jeanneau, et Horst Baier. Active wing load alleviation with an adaptive feed-forward control algorithm. Dans *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone (CO), U.S.A., 21–24 Août 2006.
- [Wira, 2002] Patrice Wira. *Réseaux Neuromimétiques, modularité et statistiques : estimation des mouvements pour l’asservissement visuel de robots*. Thèse de doctorat, Université de Haute-Alsace, 2002.