

Titre: Identification des coefficients aérodynamiques et commande de vol
Title: non linéaire

Auteur: Anne Girard
Author:

Date: 2007

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Girard, A. (2007). Identification des coefficients aérodynamiques et commande de vol non linéaire [Mémoire de maîtrise, École Polytechnique de Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/8047/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8047/>
PolyPublie URL:

Directeurs de recherche: Jules O'Shea
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

IDENTIFICATION DES COEFFICIENTS AÉRODYNAMIQUES ET
COMMANDÉ DE VOL NON LINÉAIRE

ANNE GIRARD
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(AUTOMATION ET SYSTÈMES)

JUIN 2007



Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 978-0-494-35680-7

Our file *Notre référence*

ISBN: 978-0-494-35680-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

**
Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

IDENTIFICATION DES COEFFICIENTS AÉRODYNAMIQUES ET
COMMANDÉ DE VOL NON LINÉAIRE

présenté par: GIRARD Anne

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. DE SANTIS Romano M., Ph.D., président

M. O'SHEA Jules, D. Ing., membre et directeur de recherche

M. BRAULT Jean-Jules, Ph.D., membre

à ma famille et mes proches

REMERCIEMENTS

Je tiens tout d'abord à remercier M Jules O'SHEA, mon directeur de recherche, pour m'avoir proposé ce sujet "sur mesure", qui m'a permis d'explorer les multiples facettes de la commande et du contrôle dans l'optique d'une application aéronautique.

J'adresse également mes remerciements à mes professeurs, M Jean-Jules BRAULT, qui m'a permis de découvrir l'ensemble des possibilités offertes par les réseaux de neurones, M Lahcen SAYDY pour son exemple de rigueur et de persévérence et M Romano DESANTIS, qui m'a enseigné le goût du "pas à pas" pour accomplir de grands projets.

Je remercie mes proches et mes amis, qui ont tous su me soutenir et me redonner le moral dans les temps difficiles. Sans eux et sans leur conviction profonde de ma réussite, cette expérience n'aurait pas été aussi agréable.

Finalement mes remerciements vont à Supélec (École Supérieure d'Électricité), mon école d'origine, qui m'a permis de découvrir la culture québécoise et de faire aboutir mes projets d'international et d'initiation à un mode d'apprentissage différent à travers la recherche. Je remercie tout particulièrement M Emmanuel GODOY pour son suivi, ses conseils et son soutien tout au long de cette expérience.

RÉSUMÉ

La commande de vol classique consiste à discréteriser l'enveloppe de vol, en y choisissant un ensemble de points de fonctionnement, et à concevoir des contrôleurs linéaires locaux pour chacun de ces points. Toutefois cette méthode demande un effort de conception important.

Un des principaux défis de la commande de vol moderne est de se dégager de ces contraintes grâce à la conception de lois de commande non linéaires et/ou adaptatives basées sur des modèles non linéaires d'avion.

L'objectif de ce mémoire est de développer une commande de vol non linéaire, permettant de contrôler un aéronef en angle d'attaque α , en angle de dérapage β et en angle de roulis ϕ .

La première phase de cette recherche consiste à identifier les coefficients aérodynamiques de l'avion de manière précise. Pour cela, une structure innovante à base de réseaux de neurones est développée. Celle-ci atteint les performances de précision désirée tout en étant de taille restreinte. Elle présente l'avantage de pouvoir être facilement étendue par un apprentissage en ligne, caractéristique qui lui permettra d'être utilisée dans le futur par une commande de vol reconfigurable.

En deuxième phase, une loi de commande par inversion dynamique dont les paramètres sont donnés par le module d'identification précédent est développée. L'identification précise des coefficients aérodynamiques permet de restreindre les erreurs de modélisation et l'utilisation d'une dynamique désirée de type proportionnel + intégral permet alors d'assurer une certaine robustesse au contrôleur.

L'étude se termine sur l'utilisation d'une boucle de commande prédictive, appliquée au système linéarisé par retour de sortie, qui permet d'anticiper les changements de

trajectoires et de limiter les efforts sur la commande. La robustesse de ce contrôle n'est cependant pas traitée et reste matière à recherche future.

Une application au *Fighting Falcon F-16* est faite par simulation sous MatLab et Simulink. Contrairement à la majorité des travaux publiés qui se basent sur des modèles aérodynamiques linéaires, le modèle aérodynamique utilisé est directement celui élaboré par la NASA suite à des essais en soufflerie (Nguyen et al, 1979) et est hautement non linéaire. Le module d'identification, développé sur l'ensemble de l'enveloppe de vol du F-16, comprend un nombre de réseaux de neurones suffisemment restreint pour être implémenté sur un processeur neuronal existant et réalise bien les objectifs de précision. La commande de vol par inversion dynamique est ensuite testée sur une trajectoire donnée, en premier lieu avec un modèle aérodynamique parfait puis avec un modèle aérodynamique "réel" du F-16. Les performances sur cette trajectoire confirment que les objectifs de suivi de trajectoire sont atteints et est un premier point positif pour la démonstration de la robustesse de cette loi de contrôle.

Les objectifs de ce projet ont été atteints : un système de commande d'attitude non linéaire, qui n'a pas recours au découplage des mouvements longitudinaux et latéraux et qui obvie au "gain scheduling", a été développé.

ABSTRACT

Classical flight control is achieved with linear controllers that are scheduled over the entire flight envelope. This approach, although quite straightforward, is tedious and may not be adequate for flying operations at a distance from the linearised region. Thus modern flight control aims at developing non linear and/or adaptive control laws, based on non linear plant models, which are valid over the entire flight envelope.

The objective of this theses is to develop such a control law, in order to command the aircraft's angle of attack α , sideslip angle β and roll angle ϕ .

In the first place, the aircraft's aerodynamic coefficients are identified with precision over the entire flight envelope. To do so, a novel architecture based on neural networks is developed. It achieves the desired performances, and is of a reasonable size with regards to the number of neural networks it contains. This architecture can be easily extended to the on line identification needed for reconfigurable flight control.

In the second place, a dynamic inversion control law, the parameters of which are issued by the neural identification module, is developed. The use of precise aerodynamic coefficient estimates enables limited model errors, and provides an improved robustness of the control law when the desired dynamics is generated by a proportionel+integrator feedback.

The research ends on an extension of this control law by a predictive command. This command, based on the feedback linearised model of the plant, anticipates changes in the reference trajectory and limits the control effort. However the robustness of this control law was not part of this research and remains to be studied.

The theoretical results are applied to the *Fighting Falcon* F-16 through simulation on a MatLab and Simulink platform. The neural identification module contains a limited number of networks, and achieves the desired precision performance. The dynamic inversion command is then tested on both a perfect model and a “reel” model of the plant. In both cases the results confirm that the reference was correctly tracked. This is a positive premise for the robustness of this control law.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xiv
LISTE DES FIGURES	xvi
LISTE DES NOTATIONS ET DES SYMBOLES	xx
LISTE DES ANNEXES	xxiii
INTRODUCTION	1
CHAPITRE 1 PRÉSENTATION DU SYSTÈME	8
1.1 Modèle mathématique de l'avion	8
1.1.1 Repères de référence pour la dynamique du vol	9
1.1.2 Équations newtoniennes	10
1.1.3 Équations du modèle aérodynamique	12
1.1.3.1 Modélisation linéaire	12
1.1.3.2 Modélisation non linéaire	13
1.2 Présentation du <i>Fighting Falcon F-16</i>	14
1.2.1 Un peu d'histoire	14
1.2.2 Le modèle aérodynamique du F-16	15

1.2.3	Conclusion	18
CHAPITRE 2 INTRODUCTION AUX RÉSEAUX DE NEURONES . . .		19
2.1	Le neurone artificiel	19
2.2	Perceptron multicouches	20
2.3	Apprentissage des réseaux de neurones	23
2.3.1	L'algorithme de rétropropagation	24
2.3.2	Améliorations de l'algorithme de rétropropagation	27
2.3.3	L'algorithme de Levenberg-Marquardt	28
2.4	Optimisation de l'apprentissage	29
2.4.1	Données d'entraînement	30
2.4.2	La validation croisée	31
2.4.3	L'arrêt prématuré de l'apprentissage	31
CHAPITRE 3 IDENTIFICATION DU MODÈLE AÉRODYNAMIQUE .		33
3.1	Identification hors ligne et en ligne	33
3.2	Les réseaux de neurones appliqués à l'identification hors ligne . . .	35
3.3	Extension vers un entraînement en ligne	40
3.3.1	Les hypercubes de De Weerdt (2005)	42
3.3.2	Analyse critique	46
3.3.3	Conclusion de l'analyse de la méthode des hypercubes . . .	59
3.4	Une architecture novatrice	60
3.4.1	Partitionnement en régions	62
3.4.2	Architecture d'une région	64
3.4.3	Sélection des régions	67
3.4.4	Changement de région	68
3.4.5	Une approche pour l'apprentissage en ligne	69
3.4.6	Conclusion sur l'architecture développée	71
3.5	Application au F-16	72

3.6 Conclusion	85
CHAPITRE 4 CONCEPTION DU CONTRÔLEUR	86
4.1 Qualités de manœuvrabilité	86
4.1.1 Qualités de manœuvrabilité longitudinales	89
4.1.2 Qualités de manœuvrabilité latérales	91
4.2 Commande de vol par inversion dynamique	95
4.2.1 Introduction à l'inversion dynamique	95
4.2.2 Séparation temporelle de la dynamique	96
4.2.3 Dynamique désirée	98
4.2.4 L'inversion de la commande	101
4.2.5 Un système affine par rapport à la commande	102
4.2.6 La commande en vitesse	104
4.2.7 Etude d'un modèle longitudinal simplifié	105
4.2.8 Application au modèle à 6 degrés de liberté	115
4.2.9 Conclusion sur l'inversion dynamique	124
4.3 Allocation du contrôle	124
4.3.1 Contraintes en temps discret	125
4.3.2 Allocation statique	126
4.3.3 Allocation dynamique	128
4.3.4 Conclusion sur l'allocation du contrôle	130
4.4 Les contraintes sur les états et la commande prédictive	131
4.5 Une approche pour la robustesse	132
4.6 Conclusion sur la commande de vol	133
CHAPITRE 5 RÉSULTATS DE LA SIMULATION NON LINÉAIRE . . .	136
5.1 Simulation en boucle ouverte	136
5.1.1 Manœuvre : vol à ailes horizontales	136
5.1.2 Manœuvre : virage coordonné à $\dot{\psi} = 0.5^\circ.s^{-1}$	140

5.2 Application de la structure du contrôleur au F-16	143
5.2.1 La commande en vitesse	143
5.2.2 L'allocation du contrôle	146
5.2.3 Choix des gains des dynamiques désirées	150
5.3 Simulations en boucle fermée	152
5.3.1 Cas d'un modèle parfait	152
5.3.2 Cas d'un modèle "réel"	155
5.4 Une approche succincte à la commande prédictive	158
5.4.1 Influence des différents paramètres	158
5.4.2 Simulation - suivi de trajectoire	162
5.5 Conclusion sur la simulation non linéaire	165
CHAPITRE 6 VERS DES TRAVAUX FUTURS	168
CONCLUSION	178
RÉFÉRENCES	180
ANNEXES	186

LISTE DES TABLEAUX

Tableau 1.1	Définition de l'enveloppe de vol	17
Tableau 2.1	Algorithme de rétropropagation	26
Tableau 2.2	Algorithme de Levenberg-Marquardt	29
Tableau 3.1	Dépendances des coefficients aérodynamiques	37
Tableau 3.2	Procédure de conception hors ligne d'un réseau optimal . .	39
Tableau 3.3	Valeurs des variables des points statiques	43
Tableau 3.4	Partitions en hypercubes	44
Tableau 3.5	Scénario général pour l'apprentissage en ligne	50
Tableau 3.6	Caractéristiques des instruments de mesure	56
Tableau 3.7	Algorithme pour le choix des régions	63
Tableau 3.8	Espace d'entrée du module d'identification du F-16	74
Tableau 3.9	Compilation par région des coefficients longitudinaux	75
Tableau 3.10	Compilation par région des coefficients latéraux	76
Tableau 3.11	Valeurs extrêmes des coefficients	77
Tableau 3.12	Valeurs expérimentales - Réseaux du modèle longitudinal . .	78
Tableau 3.13	Valeurs expérimentales - Réseaux du modèle latéral	79
Tableau 4.1	Niveaux de manœuvrabilité	88
Tableau 4.2	Classes d'avions	88
Tableau 4.3	Catégorie des phases de vol	89
Tableau 4.4	Coefficient d'amortissement pour le mode phugoïde	90
Tableau 4.5	Coefficient d'amortissement pour le mode rapide	91
Tableau 4.6	CAP pour le mode rapide	92
Tableau 4.7	Constante de temps pour le mode de roulis	94
Tableau 4.8	Qualités de manœuvrabilité du roulis hollandais	94
Tableau 4.9	Temps pour doubler l'amplitude du mode spirale	94
Tableau 5.1	Caractéristiques des réseaux de neurones	144

Tableau II.1	Propriétés géométriques et de masse du F-16	190
Tableau VI.1	Dynamique des actionneurs	216

LISTE DES FIGURES

Figure 1.1	Système d'axes de l'avion et angles d'incidence	9
Figure 2.1	Modèle non linéaire d'un neurone	20
Figure 2.2	Fonctions d'activation usuelles	21
Figure 2.3	Exemple de MLP à une couche cachée	22
Figure 2.4	Arrêt prématuré de l'entraînement	32
Figure 3.1	Representation partielle des points statiques	45
Figure 3.2	Representation partielle des hypercubes	45
Figure 3.3	Performance du module à région unique	48
Figure 3.4	Performance du module à régions multiples	48
Figure 3.5	Phénomène d'oubli	51
Figure 3.6	Performance suite à une augmentation d'angle d'attaque α .	51
Figure 3.7	Scénario 1 - Module d'identification à région unique	53
Figure 3.8	Scénario 1 - Performance lors d'un changement brutal d'angle d'attaque	53
Figure 3.9	Scénario 2 - Module d'identification à région unique	55
Figure 3.10	Scénario 2 - Module d'identification à régions multiples	55
Figure 3.11	Scénario 2 - Performance lors d'un changement brutal de C_x	56
Figure 3.12	Scénario 3 - Module d'identification à région unique	57
Figure 3.13	Scénario 3 - Module d'identification à régions multiples	57
Figure 3.14	Scénario 3 - Comparaison de la performance des deux modules	58
Figure 3.15	Architecture générale	61
Figure 3.16	Architecture d'une région	65
Figure 3.17	Changement de région continu	69
Figure 3.18	Illustration de l'algorithme de choix des régions appliqué au F-16	73
Figure 3.19	Coefficients longitudinaux - Erreur d'entraînement en % . .	80

Figure 3.20	Coefficients latéraux - Erreur d'entraînement en %	81
Figure 3.21	Comportement du module d'identification	82
Figure 3.22	Performance du réseau de la région 1 pour $q = 0$	83
Figure 3.23	Performance du réseau de la région 2 pour $q = 0$	83
Figure 3.24	Performance du réseau de la région 3 pour $q = 0$	84
Figure 3.25	Performance du réseau de la région 4 pour $q = 0$	84
Figure 4.1	Échelle de Cooper-Harper	87
Figure 4.2	Mouvement du mode phugoïde	90
Figure 4.3	Mouvement du roulis hollandais	93
Figure 4.4	Architecture à séparation temporelle de la dynamique	97
Figure 4.5	Architecture alternative	98
Figure 4.6	Schémas bloc comparant trois configurations de commande par inversion dynamique	109
Figure 4.7	Lieu des racines pour la première méthode	110
Figure 4.8	Réponse en tangage suite à un doublet de $1^\circ.s^{-1}$ en commande	111
Figure 4.9	Réponse suite à un doublet de 5° en commande	114
Figure 4.10	Réponse avec saturation suite à un doublet de 5° en commande	114
Figure 4.11	Architecture du contrôleur	134
Figure 5.1	Vol à ailes plates - évolution des variables d'état	137
Figure 5.2	Évolution des variables d'état suite à une perturbation sur δ_e	137
Figure 5.3	Trajectoire de l'avion suite à une perturbation sur δ_e	139
Figure 5.4	Vol à ailes plates - coefficients aérodynamiques	139
Figure 5.5	Virage - évolution des états	141
Figure 5.6	Évolution des états suite à une perturbation sur δ_a et δ_r	141
Figure 5.7	Trajectoire de l'avion suite à une perturbation sur δ_a et δ_r	142
Figure 5.8	Virage - coefficients aérodynamiques	142
Figure 5.9	Décomposition de l'autothrottle	143
Figure 5.10	Commande en P_c - Réponse à un changement de consigne	145

Figure 5.11	Commande en T - Réponse à un changement de consigne	145
Figure 5.12	Influence de W_δ sur l'allocation	149
Figure 5.13	Allocation du contrôle - rejet de perturbation	149
Figure 5.14	Suivi d'une trajectoire - réponse en vitesse V_t	153
Figure 5.15	Suivi d'une trajectoire - réponse en angles α , β et ϕ	153
Figure 5.16	Suivi d'une trajectoire - réponse en vitesse de rotation p , q et r .	154
Figure 5.17	Suivi d'une trajectoire - déflexions des surfaces δ_e , δ_a , δ_r et δ_{lef}	154
Figure 5.18	Suivi d'une trajectoire - coefficients aérodynamiques C_l , C_m et C_n	155
Figure 5.19	Suivi d'une trajectoire - réponse en vitesse V_t	156
Figure 5.20	Suivi d'une trajectoire - réponse en angles α , β et ϕ	157
Figure 5.21	Suivi d'une trajectoire - déflexion des surfaces δ_e , δ_a , δ_r et δ_{lef}	157
Figure 5.22	Influence de la matrice de pondération R	159
Figure 5.23	Influence de la matrice de pondération Q	159
Figure 5.24	Influence de l'horizon de prédiction	160
Figure 5.25	Influence du temps de maintien de la commande	161
Figure 5.26	Rejet de perturbation constante	161
Figure 5.27	Suivi de trajectoire - réponse en vitesse V_t	163
Figure 5.28	Suivi de trajectoire - réponse en α , β et ϕ	163
Figure 5.29	Suivi de trajectoire - réponse en δ_e , δ_a , δ_r et δ_{lef}	164
Figure 5.30	Performance du système linéarisé par retour de sortie sans PI	165
Figure 6.1	Commande en altitude - influence de Q	172
Figure 6.2	Commande en altitude - angle d'attaque généré	172
Figure 6.3	Commande en altitude - réponse en altitude	173
Figure 6.4	Commande en altitude - réponse en vitesse	174
Figure 6.5	Commande en altitude - C.I. $V_t = 150m.s^{-1}$ et $h = 4500m$.	175
Figure 6.6	Commande en altitude - C.I. $V_t = 100m.s^{-1}$ et $h = 5000m$.	175

Figure 6.7	Commande en altitude - C.I. $V_t = 100m.s^{-1}$ et $h = 6000m$.	176
Figure 6.8	Commande en altitude - pente de $10m.s^{-1}$	176
Figure IV.1	Linéarisation entrée-état	198
Figure IV.2	Linéarisation entrée-sortie	201
Figure VI.1	Simulateur	209
Figure VI.2	Comparaison des différents algorithmes d'interpolation en 2D	211
Figure VI.3	Comparaison des différents algorithmes d'interpolation en 1D	212
Figure VI.4	Dynamique des actionneurs	216

LISTE DES NOTATIONS ET DES SYMBOLES

α :	angle d'attaque, en rad (ou $^\circ$, précisé dans ce cas)
b :	envergure des ailes, en m
β :	angle de dérapage, en rad (ou $^\circ$, précisé dans ce cas)
\bar{c} :	corde aérodynamique moyenne, en m
C :	coefficient aérodynamique, sans dimension
CAP :	Control Anticipation Parameter, ou paramètre d'anticipation du contrôle
δ_i :	angle de déflexion de la surface de commande i (a , e ou r), en $^\circ$
δ_{th} :	position de la manette des gaz, en $\%$
g :	constante de gravitation, en $m.s^{-2}$
h :	altitude, en m
h_{moteur} :	moment angulaire du moteur, en $kg.m.s^{-2}$
I_x :	moment d'inertie de roulis, en $kg.m^{-2}$
I_y :	moment d'inertie de tangage, en $kg.m^{-2}$
I_z :	moment d'inertie de lacet, en $kg.m^{-2}$
I_{xz} :	produit d'inertie, en $kg.m^{-2}$
M :	vitesse de l'avion, en nombre de Mach
MLP :	MultiLayer Perceptron, ou perceptron multicouches
m :	masse de l'avion, en kg
p :	vitesse angulaire de roulis dans le repère avion, en $rad.s^{-1}$
q :	vitesse angulaire de tangage dans le repère avion, en $rad.s^{-1}$
\bar{q} :	pression dynamique, en Pa
r :	vitesse angulaire de lacet dans le repère avion, en $rad.s^{-1}$
ρ :	densité de l'air
S :	surface des ailes de l'avion, en m^2

T :	poussée instantanée du moteur, en N
u :	vitesse de l'avion selon l'axe x du repère avion, en $m.s^{-1}$
V_t :	vitesse vraie de l'avion, en $m.s^{-1}$
v :	vitesse de l'avion selon l'axe y du repère avion, en $m.s^{-1}$
W :	poids de l'avion, en $kg.m.s^{-2}$
w :	vitesse de l'avion selon l'axe z du repère avion, en $m.s^{-1}$
x_{cg} :	position du centre de gravité, en % de $overline{e}$
$x_{cg_{ref}}$:	position de référence du centre de gravité, en % de $overline{e}$
x_E :	position de l'avion vers le nord dans le repère terre, en m
y_E :	position de l'avion vers l'est dans le repère terre, en m
ϕ :	assiette de roulis, en rad (ou $^\circ$, précisé dans ce cas)
θ :	assiette de tangage, en rad (ou $^\circ$, précisé dans ce cas)
ψ :	azimut, en rad (ou $^\circ$, précisé dans ce cas)
ζ :	coefficient d'amortissement
ω :	pulsation propre

INDICES

a :	des ailerons
c :	commandé
des :	désiré
e :	de l'élévateur
r :	du gouvernail
l :	du moment de roulis
m :	du moment de tangage
mr :	du mode rapide
mes :	mesuré

<i>n</i> :	du moment de lacet
<i>0</i> :	composante principale
<i>p</i> :	dû au roulis
<i>ph</i> :	du mode phugôide
<i>q</i> :	dû au tangage
<i>r</i> :	dû au lacet
<i>rh</i> :	du roulis hollandais
<i>x</i> :	des forces suivant l'axe x de l'avion
<i>y</i> :	des forces suivant l'axe y de l'avion
<i>z</i> :	des forces suivant l'axe z de l'avion

LISTE DES ANNEXES

ANNEXE I	RAPPELS DE DYNAMIQUE DU VOL	186
I.1	Équations cinématiques et de navigation :	186
I.2	Équations des forces et des moments :	187
I.3	Facteurs de charge :	189
ANNEXE II	DONNÉES AÉRODYNAMIQUES DU F-16	190
ANNEXE III	CODE SOURCE MATLAB	191
ANNEXE IV	COMMANDÉ PAR LINÉARISATION EXACTE	197
IV.1	Linéarisation entrée-état	197
IV.2	Linéarisation entrée-sortie	199
ANNEXE V	COMMANDÉ PRÉdictive LINÉAIRE	202
V.1	Formulation du problème	202
V.2	Le problème de programmation quadratique équivalent	204
ANNEXE VI	SIMULATEUR NON LINÉAIRE DU F-16	208
VI.1	Les variables d'état du F-16	208
VI.2	Les équations du mouvement	208
VI.3	Le modèle aérodynamique	210
VI.3.1	Pour la simulation du comportement “réel” du F-16	210
VI.3.2	Pour la simulation du F-16 identifié	213
VI.4	Le modèle du moteur	213
VI.5	Le modèle de l'atmosphère	214
VI.6	La dynamique des actionneurs	216
VI.7	Les conditions de vol de l'avion	217

INTRODUCTION

Dès le milieu du 20ème siècle, les concepteurs d'aéronefs comprennent les avantages de remplacer le système de commande de vol hydraulique ou mécanique en boucle ouverte par une commande analogique en chaîne fermée, dite "fly-by-wire". Les commandes de vol électriques permettent une nette réduction de masse et augmentent la flexibilité en réponse du système. Elles permettent également une meilleure manœuvrabilité et une augmentation de la sécurité (notamment vis-à-vis du décrochage). En outre les configurations aérodynamiques instables ne sont plus à exclure de l'enveloppe de vol, ensemble des configurations de l'avion admissibles et contrôlées. Ces configurations sont particulièrement intéressantes pour les avions militaires, car elle implique à la fois une meilleure efficacité aérodynamique et une furtivité accrue.

Motivation

La commande de vol classique consiste à discréteriser l'enveloppe de vol, en y choisissant un ensemble de points de fonctionnement, et à concevoir des contrôleurs linéaires locaux pour chacun de ces points. Toutefois cette méthode, dite de programmation des gains ou "gain scheduling", bien que simple et efficace et à laquelle les industriels font confiance du fait de son établissement sur le long terme et de ses résultats sécuritaires dans les conditions nominales, demande un effort de conception important et une identification lourde. De plus si l'avion est mené à se trouver dans des conditions de vol trop éloignées de la région linéarisée, la stabilité du système peut être compromise ; il faut donc discréteriser l'espace d'état avec finesse.

Un des principaux défis de la commande de vol moderne est de se dégager de ces contraintes grâce à la conception de lois de commande non linéaires et/ou adaptatives basées sur des modèles non linéaires d'avion.

Ce mémoire s'intéresse en particulier à l'utilisation des réseaux de neurones dans une telle application.

État de l'art

Dès les années 1950, les chercheurs portent leur intérêt sur le développement de la commande adaptative en vue de concevoir des autopilotes pour des aéronefs à haute performance. En effet, de tels avions fonctionnent sur une large plage de vitesses et d'altitudes. Il est mis en évidence que les lois de contrôle linéaires usuelles, soit à gain constant, sont adaptées à une condition de fonctionnement particulière, mais pas à l'ensemble du régime de vol (Calise & Rysdyk, 1998a). Toutefois le problème de la commande adaptative appliquée à la commande de vol était trop complexe par rapport aux techniques et à la puissance de calcul des processeurs disponibles à l'époque. La technique de programmation des gains, ou "gain scheduling", est alors jugée la méthode la plus adaptée à la commande de vol.

Ce n'est qu'au milieu des années 1980, que des méthodes de contrôle non linéaires sont appliquées, dans la théorie, à la commande de vol. Une des premières méthodes utilisées pour la commande de vol non linéaire est l'inversion dynamique, une technique qui transforme un système initial non linéaire en un système résultant à dynamique linéaire (Meyer, Su & Hunt, 1984). Initialement elle est utilisée pour contrôler certaines sorties spécifiques du système (Lane & Stengel, 1988). Mais la robustesse de ce contrôleur laisse à désirer. Dans les applications suivantes de l'inversion dynamique à la commande de vol, la dynamique de vol est séparée en deux ensembles: celui à dynamique lente et celui à dynamique rapide (Sun &

Clarke, 1994). Une première boucle externe contrôle les états lents (les angles significatifs) en utilisant les états rapides (les taux de rotation significatifs) comme entrée et une boucle interne contrôle les états rapides à partir des entrées réelles du système (élévateur, ailerons et gouvernail). Toutefois la stabilité en boucle fermée n'est pas pour autant assurée, du fait que l'hypothèse d'une telle séparation en régime lent et rapide n'est pas toujours validée. La solution proposée par Escande (1997) pour le problème de benchmark GARTEUR utilise une telle inversion, et ajoute au contrôleur à inversion dynamique non linéaire un contrôleur à régulation linéaire quadratique afin de parvenir aux objectifs de performances. Une autre difficulté rencontrée lors de la conception d'un contrôleur à inversion dynamique est qu'il nécessite un modèle complet et précis de la dynamique de vol, y compris des coefficients aérodynamiques comme le soulignent Lee & Kim (2001). Ainsi l'identification du système fait partie intégrante de la conception d'un contrôleur non linéaire.

Les réseaux de neurones sont particulièrement adaptés pour l'identification et le contrôle de systèmes dynamiques variant dans le temps (Narendra & Parthasarathy, 1990), a fortiori à la commande de vol.

La méthode la plus simple pour construire un système de commande de vol à réseaux de neurones est l'inversion directe du modèle mathématique de l'avion sur l'ensemble de l'enveloppe de vol. Avec des méthodes classiques une telle inversion n'est pas triviale comme il a été souligné précédemment, mais l'utilisation de réseaux de neurones y est adaptée (Huang et al, 1992).

Les réseaux de neurones pouvant approximer de manière précise toute fonction continue inconnue, leur utilisation pour l'identification au sein de la commande prédictive de systèmes hautement non linéaires apparaît également comme un choix naturel, que ce soit pour un modèle linéaire d'aéronef (Nho & Agarwal, 1999) ou pour un processus industriel non linéaire (Zamarreño & Vega, 1999). Classique-

ment la commande prédictive est utilisée dans des processus industriels dont les constantes de temps étaient lentes. Mais l'expérience menée par Haley, Soloway et Gold (1999) a prouvé que celle-ci convenait également à des systèmes à constantes de temps rapides tels que les aéronefs grâce à l'apprentissage en ligne.

À ses débuts, l'utilisation de l'entraînement en ligne des réseaux de neurones est litigieuse pour les aéronefs. Afin de s'assurer que le contrôleur conserve un comportement de base connu, le réseau de neurones développé hors-ligne n'est pas entraîné en ligne, il est donc fixe, mais est augmenté d'un élément adaptatif neuronal qui, lui, apprend en ligne et permet d'annuler les erreurs d'inversion (Kim & Calise, 1993). Une autre alternative est de conserver une technique de contrôle classique, par exemple l'inversion du système, et de l'augmenter de manière analogue à précédemment (Calise & Rysdyk, 1998b). De manière plus générale, un dispositif adaptatif neuronal, ayant pour objectif de compenser les erreurs de modélisation ou celles dues à la non linéarité, peut être ajouté au contrôleur existant, qu'il soit à inversion dynamique (Somakumar & Chandrasekhar, 1998), à retour linéarisant (Lee & Kim, 2001) ou prédictif à modèle de référence (Menhaj & Ray, 2003).

L'utilisation de l'apprentissage en ligne des réseaux de neurones est peu à peu intégrée dans les lois de contrôle, que ce soit de manière indirecte pour l'identification ou directe pour le contrôle. Les lois de commande utilisées par ces contrôleurs neuronaux sont d'ailleurs directement dérivées des lois de contrôle adaptatif classiques comme le contrôle inverse direct (Gili & Battipede, 2001), le contrôle direct adaptatif à modèle de référence (Kamalasadan & Ghandakly, 2005), le contrôle indirect adaptatif à modèle de référence (Suresh et al., 2005) ou encore à modèle interne (Yu, Zhu & Sun, 2005). La programmation des gains d'un contrôleur PID peut même être remplacée par un algorithme adaptatif au sein d'un estimateur à réseau de neurones (Savran, Tasaltin & Becerikli, 2006).

L'aspect reconfigurable de la commande de vol est au cœur des préoccupations actuelles pour la commande de vol. En effet dès 1984, l'US Air Force lance un programme, "Self-repairing flight control system program", ayant pour objectif d'améliorer la fiabilité et la survie d'un aéronef (Eslinger & Chandler, 1988) suite à une panne d'actionneur ou de capteur, ou à un dégât de combat par exemple. On distingue cependant deux approches à la commande de vol reconfigurable (Bodson & Groszkiewicz, 1995).

La première approche, qui découle du programme ci-dessus, est la plus répandue. Les contrôleurs de vols reconfigurables y sont composés d'un module d'identification et d'isolation de panne, puis pour chaque type de faute une structure de contrôleur est préprogrammée. Cependant le nombre de types de panne ou de modification du comportement de l'aéronef est trop important pour être exhaustivement répertorié, ce qui limite l'envergure de leur application.

La deuxième approche consiste à identifier en ligne la dynamique de l'avion et de concevoir automatiquement un contrôleur adapté à celle-ci. Groszkiewicz et Bodson (1997) établissent le lien existant entre ce type de commande de vol reconfigurable et la commande adaptative : la commande adaptative permet non seulement de s'adapter à des conditions différentes de vol mais aussi à une large gamme de pannes sans que celles-ci soient prédéfinies.

Les réseaux de neurones traitent cette seconde approche en incorporant directement le caractère reconfigurable dans le contrôleur sans ajout excessif de complexité. En effet les contrôleurs à réseaux de neurones, qui utilisent de manière extensive l'apprentissage à la fois hors ligne et en ligne, sont capables d'identifier en temps réel de nouvelles configurations, survenant suite soit à un changement de condition de vol soit à une panne ou un dégât qui crée des changements aérodynamiques, sans le recours d'un module de détection de panne (Calise, Lee & Sharma, 2001). Par exemple, un contrôleur reconfigurable par retour d'état à réseau de neurones

permet la stabilisation et l'atterrissement sécuritaire d'un avion suite à une panne d'un élévateur (Pashilkar, Sundararajan & Saratchandran, 2006) ou encore un contrôleur reconfigurable adaptatif direct à réseau de neurones permet de stabiliser un avion dans le cas d'une erreur de modélisation de 30% ou d'une panne d'actionneur (Suresh, Omkar & Mani, 2006). La commande prédictive a également un intérêt particulier vis-à-vis de la reconfiguration d'une commande de vol (Soloway et Haley, 2001 ; Kale et Chipperfield, 2002). En effet le réseau de neurones modélisant le système est entraîné en ligne et s'adapte aux changements pouvant affecter le comportement du système ; les paramètres de la commande étant basés sur ce modèle sont naturellement adaptés suite au changement du modèle.

Cependant jusqu'à maintenant aucun avion de transport civil n'est doté d'une commande autre que par programmation des gains. Les compagnies d'aviation civile sont peu enclines à évoluer vers de nouvelles technologies car la fiabilité de celles-ci ne peut être prouvée à grande échelle directement et la sécurité des voyageurs est primordiale. Il semble donc plus probable que de tels contrôleurs adaptatifs et/ou reconfigurables soient d'abord implantés dans des avions militaires, permettant ainsi la poursuite d'une mission ou une furtivité accrue par rapport aux contrôleurs existants.

Objectifs et contribution

L'objectif de ce projet est de concevoir un contrôleur de vol non linéaire, basé sur l'utilisation de réseaux de neurones et leur grande capacité d'adaptation, dans le cadre d'une application sur un avion de chasse, le *Fighting Falcon F-16*. Il devra être valable sur l'ensemble de l'enveloppe de vol et non autour d'un point de fonctionnement localisé. De plus, la conception veillera à garder une flexibilité suffisante pour une évolution du contrôleur vers un contrôleur reconfigurable.

La démarche qui sera suivie est la suivante :

- Développement d'un module d'identification du modèle aérodynamique du F-16. Celui-ci est nécessaire, que la commande choisie soit directe ou indirecte.
- Conception d'une boucle interne stabilisatrice sur la dynamique rapide du F-16, par inversion dynamique.
- Conception d'une boucle externe d'asservissement. L'inversion dynamique, éventuellement augmentée d'une commande prédictive, sera utilisée pour celle-ci.
- Simulation non linéaire afin de valider les performances du système contrôlé.

Ce projet est uniquement mené en simulation et aucune validation sur prototype n'y est faite. La validation définitive des résultats énoncés ici reste donc à faire avant d'implanter le contrôleur dans le système réel, notamment la capacité et la vitesse de calcul des processeurs doivent être vérifiées.

Organisation du mémoire

Le mémoire est organisé en 5 chapitres.

En première partie (chapitres 1 et 2), le système à l'étude ainsi que des outils mathématiques sont présentés. La deuxième partie (chapitres 3 et 4) résout le problème de l'identification et du contrôle d'un aéronef. Enfin, la troisième partie (chapitre 5) est une application par simulation au F-16 de la solution développée dans la partie précédente.

CHAPITRE 1

PRÉSENTATION DU SYSTÈME

La conception de contrôleurs requiert une certaine connaissance du système à commander. Celle-ci est généralement une description mathématique du système, aussi appelé modèle du système. Il existe deux manières de déterminer un tel modèle : il peut être déduit des lois de la physique, ou bien extrait d'un ensemble de données récoltées au cours d'expériences sur le système. Cette deuxième méthode est appelée “identification du système” par opposition à l'appellation usuelle “modélisation du système” de la première.

Dans ce chapitre, un modèle mathématique général de l'avion est d'abord présenté. Il est ensuite appliqué de manière spécifique au F-16.

1.1 Modèle mathématique de l'avion

L'identification de l'avion est essentiellement concernée par le fait d'obtenir une description mathématique de ses forces et de ses moments aérodynamiques en fonction de grandeurs mesurables telles que les déflexions des surfaces de contrôle, les vitesses angulaires de l'avion, la vitesse de l'avion ou encore son orientation (Klein & Morelli, 2006). L'estimation des paramètres aérodynamiques, à partir de données d'essais en vol ou en soufflerie, nécessite qu'un modèle du mouvement et un modèle aérodynamique de l'avion soient postulés. Ils sont successivement présentés dans cette section après un rappel sur les repères de référence dans lesquels ils peuvent être établis.

1.1.1 Repères de référence pour la dynamique du vol

Le repère géographique : Ce repère de référence, aussi appelé repère NED, est un repère orthogonal direct dont le centre peut être choisi arbitrairement, ici il sera choisi de sorte à coïncider avec la projection du centre de gravité de l'avion sur la surface de la terre. L'axe des x pointe dans la direction du nord, l'axe des y vers l'est et l'axe des z pointe vers le bas de manière parallèle au vecteur gravité local. C'est cette configuration (north, east, down) qui lui a donné son acronyme NED.

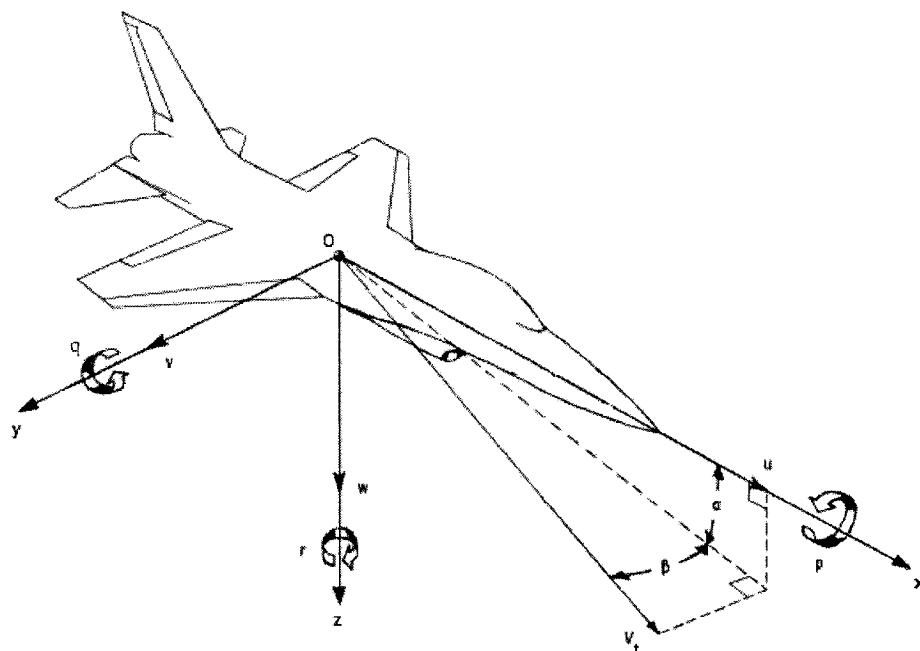


Figure 1.1 Système d'axes de l'avion et angles d'incidence

Le repère avion : Ce repère est un repère orthogonal direct dont l'origine O est le centre de gravité de l'avion. Le plan xOz coïncide avec le plan de symétrie de l'avion, de sorte que l'axe des x ait la même direction que le nez de l'avion et l'axe des z pointe vers le bas lorsque les ailes sont à l'horizontal. L'axe des y est pointé dans une direction parallèle à celle de l'aile droite de l'avion.

Le vecteur vitesse dans le système d'axes de l'avion (voir Figure 1.1) met en évidence deux relations angulaires importantes :

- l'angle d'attaque α est l'angle formé dans le plan $x0z$ entre l'axe des x et la projection du vecteur vitesse parallèlement à l'axe y sur ce plan.
- l'angle de dérapage β est l'angle formé dans le plan $V0y$ entre la projection de l'axe des x parallèlement à l'axe z sur ce plan et le vecteur vitesse V_t .

La figure 1.1 illustre également les relations entre les vitesses selon les axes de l'avion et les trois grandeurs caractéristiques que sont la vitesse réelle V_t , l'angle d'attaque α et l'angle de dérapage β :

$$\begin{aligned} V_t &= \sqrt{u^2 + v^2 + w^2} \\ \alpha &= \tan^{-1} \left(\frac{w}{u} \right) \\ \beta &= \sin^{-1} \left(\frac{v}{V_t} \right) \end{aligned} \quad (1.1)$$

On peut donc reformuler 1.1 sous la forme suivante pour obtenir les composantes du vecteur vitesse :

$$\begin{aligned} u &= V_t \cos \alpha \cos \beta \\ v &= V_t \sin \beta \\ w &= V_t \sin \alpha \cos \beta \end{aligned} \quad (1.2)$$

1.1.2 Équations newtoniennes

Afin de ne pas rendre les équations trop complexes, certaines hypothèses simplificatrices sont faites. Premièrement l'avion est considéré comme un corps rigide, ce qui implique que l'ensemble des modes élastiques ne seront pas traités dans ce projet. De plus il est supposé symétrique par rapport aux axes x - z du repère de

l'avion, de ce fait les produits d'inertie I_{xy} et I_{yz} sont nuls. La masse de l'avion est prise constante. On estime que, sur la fenêtre d'observation, la consommation de carburant est faible et qu'il n'y a pas de déplacement de carburant entraînant une modification de la position du centre de gravité et a fortiori des caractéristiques aérodynamiques de l'avion. Deuxièmement la rotation de la terre, ainsi que l'effet de son rayon de courbure, sont jugés négligeables ; ce qui permet d'assimiler les taux de rotation dans le repère avion à ceux dans un repère inertiel. Finalement seules les forces aérodynamiques (au nombre de 3), de poussée et de gravitation sont prises en compte. De plus la gravité a été estimée constante, et donc indépendante de l'altitude de l'avion.

L'ensemble des équations caractérisant le mouvement de l'avion sont induites des principes de la dynamique newtonienne(voir annexe I). Il peut être représenté par l'équation différentielle suivante qui constitue un modèle d'état de l'avion :

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) \quad (1.3)$$

où \vec{x} est un vecteur des variables d'état $V_t, \alpha, \beta, p, q, r, \phi, \theta, \psi, x_E, y_E, h$; et \vec{u} est un vecteur des variables de contrôle composé de la manette des gaz et des déflexions des surfaces de contrôle. Les variables du vecteur \vec{u} n'apparaissent pas explicitement dans l'ensemble des équations précédentes, mais y figure implicitement car elles influencent la poussée, les forces et moments aérodynamiques agissant sur l'avion.

Les variables mesurées, ou les sorties du modèle mathématique de l'avion, sont $V_t, \alpha, \beta, p, q, r, \phi, \theta, \psi, x_E, y_E, h, a_x, a_y, a_z, \dot{p}, \dot{q}$ et \dot{r} .

L'équation de sortie est alors de la forme suivante :

$$\vec{y} = H(\vec{x}, \dot{\vec{x}}, \vec{u}) \quad (1.4)$$

1.1.3 Équations du modèle aérodynamique

En général, les coefficients aérodynamiques de force et de moment dépendent non linéairement des valeurs présentes et passées de la vitesse de l'avion, des angles d'incidences, des vitesses angulaires de l'avion et des déflections des surfaces de contrôle. Des méthodes analytiques, telles que la dynamique des fluides, permettent de trouver ces dépendances. Cependant, bien que ces méthodes donnent de bons résultats pour de faibles angles d'attaques et de faibles vitesses de rotation, une meilleure estimation est obtenue par des méthodes expérimentales. Parmi celles-ci on compte les essais en soufflerie ou encore les mesures obtenues lors d'essais en vol. En pratique la dépendance par rapport aux valeurs passées est négligée en considérant que l'écoulement est quasi-stationnaire.

1.1.3.1 Modélisation linéaire

Les équations de ce modèle sont obtenues par un développement de Taylor linéaire des forces et des moments aérodynamiques. Celles-ci sont ensuite normalisées afin d'obtenir les coefficients sans dimension du modèle longitudinal C_x , C_z et C_m et ceux du modèle latéral C_y , C_l et C_n :

$$\begin{aligned} C_a &= C_{a_0} + C_{a_V} \frac{\Delta V}{V_0} + C_{a_\alpha} \Delta \alpha + C_{a_q} \frac{q \bar{c}}{2V_0} + C_{a_\delta} \Delta \delta \quad \text{pour } a = x, z \text{ ou } m \\ C_a &= C_{a_0} + C_{a_\beta} \Delta \beta + C_{a_p} \frac{pb}{2V_0} + C_{a_r} \frac{rb}{2V_0} + C_{a_\delta} \Delta \delta \quad \text{pour } a = y, l \text{ ou } n \end{aligned} \quad (1.5)$$

Pour de plus amples détails sur cette modélisation se référer à Etkin (1978).

Ce modèle linéaire est valable localement autour de la condition de vol pour laquelle il a été établi. Cette approche convient bien au “gain scheduling” mais nécessite un processus d'adaptation dans le cas où elle est utilisée pour identifier le modèle aérodynamique sur l'ensemble de l'enveloppe de vol.

1.1.3.2 Modélisation non linéaire

Dans le cas où le modèle mathématique de l'avion recherché est non linéaire, une modélisation non linéaire des coefficients aérodynamiques est préférable.

Généralement les coefficients aérodynamiques sont alors modélisés comme la somme d'un terme statique, qui dépend des angles d'incidence, et de termes dynamiques qui dépendent linéairement des vitesses angulaires et des déflections des surfaces de contrôle. De plus il est usuel de considérer que, pour des régimes de vol subsoniques, les changements de vitesse n'affectent pas la valeur des coefficients aérodynamiques par rapport à celle obtenue à une vitesse V_0 donnée (Klein & Morelli, 2006):

$$\begin{aligned} C_a &= C_{a_0}(\alpha, \beta)_{q=\delta=0} + C_{a_q}(\alpha) \frac{q\bar{c}}{2V_0} + C_{a_\delta}(\alpha)\delta && \text{pour } a = x, z \text{ ou } m \\ C_a &= C_{a_0}(\alpha, \beta)_{p=r=\delta=0} + C_{a_p}(\alpha) \frac{pb}{2V_0} + C_{a_r}(\alpha) \frac{rb}{2V_0} + C_{a_\delta}(\alpha)\delta && \text{pour } a = y, l \text{ ou } n \end{aligned} \quad (1.6)$$

Ce modèle fournit des expressions similaires à celles utilisées pour les essais en soufflerie. Les valeurs des coefficients aérodynamiques peuvent alors être obtenues par interpolation des valeurs mesurées fournies sous forme de tables. On remarque également que, contrairement à la modélisation linéaire, les variables ne sont pas des variations par rapport à une valeur de référence car le modèle est valable sur l'ensemble de l'enveloppe de vol. De ce fait, une telle modélisation est privilégiée dans le cadre de ce mémoire.

Klein et Morelli (2006) soulignent que, dans le cas de manœuvres de grande amplitude ou de vol à angle d'attaque élevé, le comportement aérodynamique peut être considérablement différent de celui observé à angle d'attaque plus faible par exemple. Il est alors préférable d'identifier plusieurs modèles en partitionnant l'espace des variables indépendantes dont dépendent les coefficients aérodynamiques.

1.2 Présentation du *Fighting Falcon F-16*

L'ensemble du projet est appliqué au *Fighting Falcon F-16*. Le choix de cette application est intimement lié au fait que le rapport de Nguyen et al. (1979) a été rendu public et est donc une grande source d'information sur le modèle mathématique de l'avion mais aussi sur les coefficients aérodynamiques, les résultats des essais en soufflerie étant compris dans ce rapport. Après un bref historique du F-16 et de ses spécificités, le modèle aérodynamique de celui-ci est présenté.

1.2.1 Un peu d'histoire

La naissance du F-16, appelé communément *Fighting Falcon*, est issue d'un projet émanant du Département de la Défense des États-Unis. En 1971, le programme visant à concevoir un avion de chasse léger ("Lightweight Fighter") avec des spécifications précises est mis en place. Cet avion de chasse se devait d'être léger, de l'ordre de 9 000 kg soit la moitié du poids du plus léger avion de chasse de l'époque, d'être peu coûteux en conception, de petite taille et l'accent devait être mis sur une manœuvrabilité accrue, notamment en accélération et vitesses de rotation, au détriment de la vitesse. A l'origine les spécifications étaient de mach 1.6 à 12 000 m d'altitude. Le premier prototype de F-16 pris son envol en 1974, pour devenir le chasseur bombardier multi-rôle le plus répandu au monde.

En outre, le F-16 est doté de certaines caractéristiques innovantes. C'est le premier avion à se voir doter d'un système "fly-by-wire" : il n'y a aucun lien mécanique entre les commandes du pilote et les surfaces de contrôles que sont l'élévateur, les ailerons et l'empennage de direction. À la place, les commandes du pilote sont transmises à un système de bord qui se charge de transmettre l'information aux surfaces de commandes après avoir fait les ajustements nécessaires. Ce système de

contrôle entièrement informatisé est d'ailleurs rendu nécessaire par la dynamique du F-16, doté d'une stabilité statique négative. Généralement un avion est conçu avec une stabilité positive lui permettant en absence de contrôle de rester dans une position d'équilibre, mais cela se traduit par une réduction de manœuvrabilité. L'accent étant mis sur la manœuvrabilité accrue, le F-16 a une stabilité statique légèrement négative en vol subsonique, mais bénéficie d'une stabilité positive en vol supersonique.

1.2.2 Le modèle aérodynamique du F-16

Le modèle aérodynamique du F-16 est basé sur le rapport du centre de recherche de Langley de la NASA (Nguyen et al, 1979). Celui-ci rend public les propriétés géométriques et massiques de l'avion, les entrées possibles des actionneurs, les équations de mouvement, le modèle atmosphérique utilisé, les données aérodynamiques et le modèle du moteur.

Les données aérodynamiques proviennent d'essais en soufflerie statiques et dynamiques à faible vitesse sur un modèle réduit à 16% du F-16 et sont fournies sous forme de tables pour un large spectre d'angles d'attaque et d'angles de dérapage: $-20^\circ < \alpha < 45^\circ$ et $-30^\circ < \beta < 30^\circ$.

Le F-16 possède, outre les surfaces de contrôles usuelles, un volet sur le bord d'attaque. Celui-ci n'est pas commandé directement par le pilote, il est ajusté automatiquement en fonction de l'angle d'attaque, de la pression statique et de la pression dynamique, selon l'équation 1.7.

$$\delta_{lef} = 1.38 \frac{2s + 7.25}{s + 7.25} \left(\alpha \frac{180}{\pi} \right) - 9.05 \frac{\bar{q}}{P_s} + 1.45 \quad (1.7)$$

Les équations non linéaires qui régissent la valeur des coefficients aérodynamiques sont analogues à celles du modèle général présenté en 1.1.3.2 à quelques modifications près :

- le volet du bord d'attaque introduit des termes supplémentaires
- la contribution due à l'élévateur n'est pas dissociée du terme principal

Les équations pour le modèle longitudinal sont les suivantes :

$$\begin{aligned}
 C_a &= C_{a_0}(\alpha, \beta, \delta_e) + q \left(\frac{\bar{c}}{2V_t} \right) \left(C_{a_q}(\alpha) + \Delta C_{a_lef_q}(\alpha) \left(1 - \frac{\delta_{lef}}{25} \right) \right) \\
 &\quad + \Delta C_{a_lef}(\alpha, \beta) \left(1 - \frac{\delta_{lef}}{25} \right) \text{ pour } a = x \text{ ou } z \\
 C_m &= C_{m_0}(\alpha, \beta, \delta_e) \eta_{\delta_e} + q \left(\frac{\bar{c}}{2V_t} \right) \left(C_{m_q}(\alpha) + \Delta C_{m_lef_q}(\alpha) \left(1 - \frac{\delta_{lef}}{25} \right) \right) \\
 &\quad + \Delta C_{m_lef}(\alpha, \beta) \left(1 - \frac{\delta_{lef}}{25} \right) + \Delta C_m(\alpha) + (x_{cg_{ref}} - x_{cg}) C_z
 \end{aligned} \tag{1.8}$$

et pour le modèle latéral pour $a = l, y$ ou n :

$$\begin{aligned}
 C_a &= C_{a_0}(\alpha, \beta, \delta_e) + \Delta C_{a_lef}(\alpha, \beta) \left(1 - \frac{\delta_{lef}}{25} \right) + \Delta C_{a, \delta_r=30^\circ}(\alpha, \beta) \left(\frac{\delta_r}{30} \right) \\
 &\quad + \left(\Delta C_{a, \delta_a=20^\circ}(\alpha, \beta) + \Delta C_{a_lef, \delta_a=20^\circ}(\alpha, \beta) \left(1 - \frac{\delta_{lef}}{25} \right) \right) \left(\frac{\delta_a}{20} \right) \\
 &\quad + p \left(\frac{b}{2V_t} \right) \left(C_{a_p}(\alpha) + \Delta C_{a_lef_p}(\alpha) \left(1 - \frac{\delta_{lef}}{25} \right) \right) \\
 &\quad + r \left(\frac{b}{2V_t} \right) \left(C_{a_r}(\alpha) + \Delta C_{a_lef_r}(\alpha) \left(1 - \frac{\delta_{lef}}{25} \right) \right) + \Delta C_{a_{beta}}(\alpha) \beta \\
 &\quad \left(-\frac{\bar{c}}{b} (x_{cg_{ref}} - x_{cg}) C_y \text{ si } a = n \right)
 \end{aligned} \tag{1.9}$$

où pour tout a :

$$\begin{aligned}
 \Delta C_{a,lef}(\alpha, \beta) &= C_{a,lef}(\alpha, \beta) - C_{a_0}(\alpha, \beta, \delta_e = 0) \\
 \Delta C_{a,\delta_r=30^\circ}(\alpha, \beta) &= C_{a,\delta_r=30^\circ}(\alpha, \beta) - C_{a_0}(\alpha, \beta, \delta_e = 0) \\
 \Delta C_{a,\delta_a=20^\circ}(\alpha, \beta) &= C_{a,\delta_a=20^\circ}(\alpha, \beta) - C_{a_0}(\alpha, \beta, \delta_e = 0) \\
 \Delta C_{a,lef,\delta_a=20^\circ}(\alpha, \beta) &= C_{a,lef,\delta_a=20^\circ}(\alpha, \beta) - C_{a,lef}(\alpha, \beta) - \Delta C_{a,\delta_a=20^\circ}(\alpha, \beta)
 \end{aligned} \tag{1.10}$$

Les données fournies sous forme de table dans le rapport de la NASA (Nguyen et al, 1979) et leur illustration permet d'apprécier la non linéarité des différentes contributions aux coefficients aérodynamiques. Soulignons toutefois que ces données étant discrètes, il est usuel d'utiliser des méthodes d'interpolation pour obtenir des valeurs sur l'ensemble de l'enveloppe de vol.

Ce modèle aérodynamique est valable dans l'enveloppe de vol définie par le tableau 1.1, dont les données proviennent du rapport de la Nasa (Nguyen et al, 1979).

Tableau 1.1 Définition de l'enveloppe de vol

Variable	Bornes	Unité
vitesse M	[0.1 ; 1]	nb de Mach
altitude h	[0 ; 15 500]	m
angle d'attaque α	[-10 ; 45]	degré
angle de dérapage β	[-30 ; 30]	degré
vitesse angulaire en roulis p	[-595 ; 595]	$^\circ.s^{-1}$
vitesse angulaire en tangage q	[-205 ; 205]	$^\circ.s^{-1}$
vitesse angulaire en lacet r	[-135 ; 135]	$^\circ.s^{-1}$
déflexion de l'élévateur δ_e	[-25 ; 25]	degré
déflexion des ailerons δ_a	[-21 ; 21]	degré
déflexion du gouvernail δ_r	[-30 ; 30]	degré
déflexion du bord d'attaque δ_{lef}	[0 ; 25]	degré

1.2.3 Conclusion

Dans ce chapitre, le système à l'étude a été présenté. Le modèle mathématique général de l'avion a d'abord été rappelé, et une attention particulière a été apportée au modèle aérodynamique de l'avion. Le choix de cette modélisation détermine le degré de précision du modèle, et c'est également la valeur de ses paramètres qui différencie le comportement des différents avions.

L'ensemble est alors appliqué au F-16 car cet avion de combat sert d'illustration à travers l'ensemble de ce mémoire.

CHAPITRE 2

INTRODUCTION AUX RÉSEAUX DE NEURONES

Les réseaux de neurones tirent leur nom de leur grande similitude avec le cerveau humain. L’agencement de structures individuellement simples, les neurones, permet de créer une structure intelligente capable de résoudre des problèmes complexes: le cerveau humain est capable, par exemple, de reconnaître un visage familier en moins de 200ms. Les réseaux de neurones artificiels ne sont pas aussi performants, mais leur capacité est bien supérieure à celle d’un processeur conventionnel.

Dans cette section, la modélisation mathématique d’un neurone est tout d’abord présentée, puis l’architecture classique des Multi-Layer Perceptron (MLP) est explicitée (Haykin, 1999).

2.1 Le neurone artificiel

L’élément de base d’un réseau de neurones est le neurone artificiel ou “perceptron”.

Le modèle mathématique du perceptron est présenté par la figure 2.1.

Soit un neurone k , on lui présente en entrée les signaux $x_1, x_2 \dots x_p$. Ils sont respectivement multipliés par des coefficients, appelés poids synaptiques, $\omega_{k1}, \omega_{k2} \dots \omega_{kp}$.

On définit alors le champ induit local ν_k comme suit :

$$\nu_k = \sum_{j=1}^p \omega_{kj} x_j \quad (2.1)$$

Ce champ est additionné d’une constante, appelée biais, b_k puis traité par une

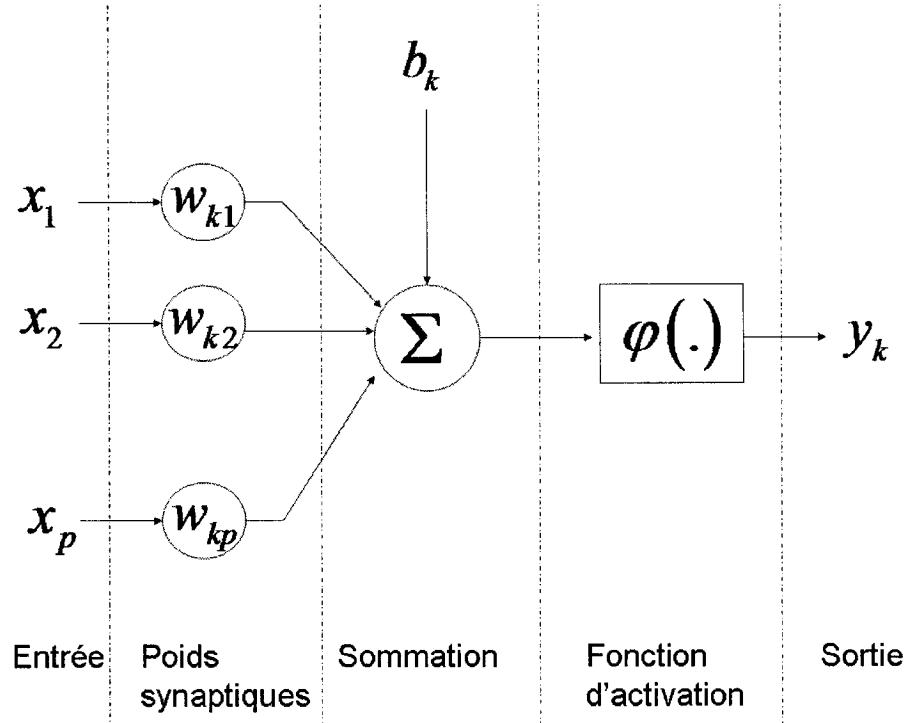


Figure 2.1 Modèle non linéaire d'un neurone

fonction d'activation φ pour donner la sortie du neurone:

$$y_k = \varphi(\nu_k + b_k) \quad (2.2)$$

La fonction d'activation peut être linéaire ou non, la figure 2.2 illustre les types les plus répandus. Le choix des fonctions à utiliser est intimement lié au problème à résoudre.

2.2 Perceptron multicouches

Les perceptrons multicouches (MLP) constituent une classe importante des réseaux de neurones, et sont les plus utilisés dans le cadre de l'identification et du contrôle des systèmes dynamiques non linéaires.

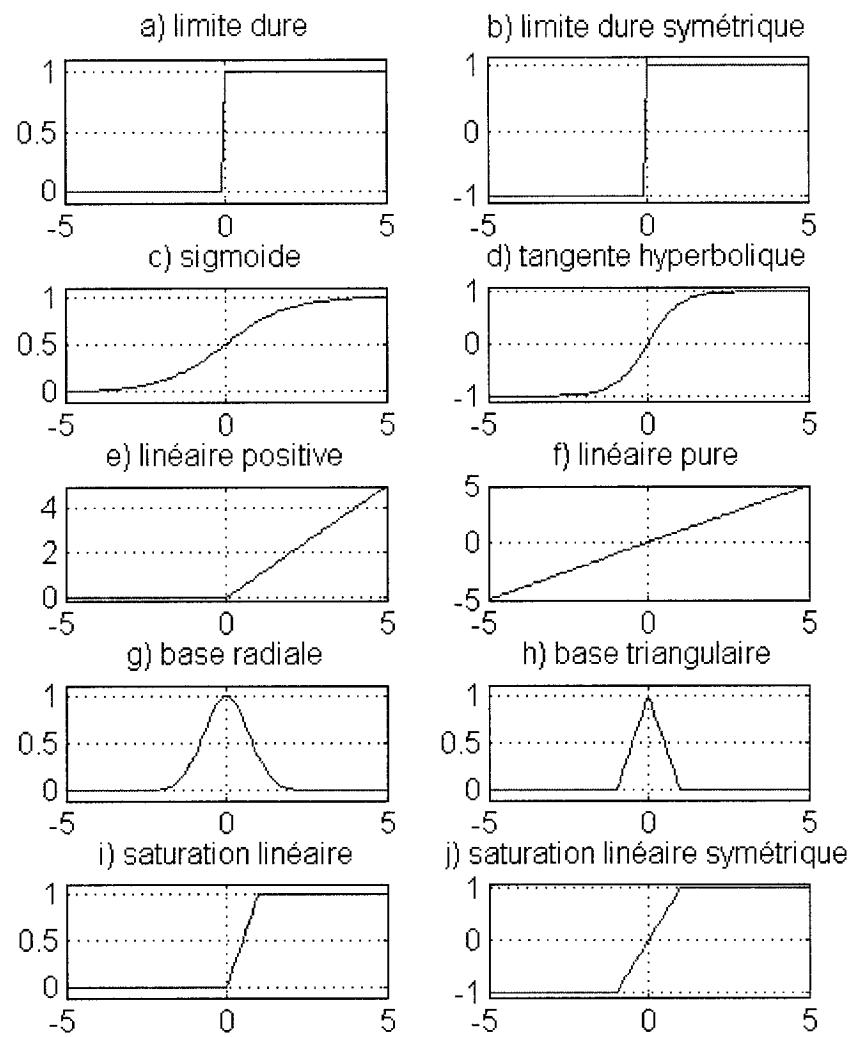


Figure 2.2 Fonctions d'activation usuelles

Un MLP est composé de plusieurs couches, elles-mêmes composés d'un ensemble de neurones. Chaque neurone du réseau est un perceptron tel que décrit dans la section précédente. On distingue la couche d'entrée, la ou les couche(s) cachée(s) et la couche de sortie, comme illustré dans la figure 2.3.

La propriété fondamentale des MLP est la suivante : toute fonction f continue sur un ensemble compact peut être approximée par un MLP à une couche cachée pour peu que celle-ci ait un nombre suffisant de neurones.

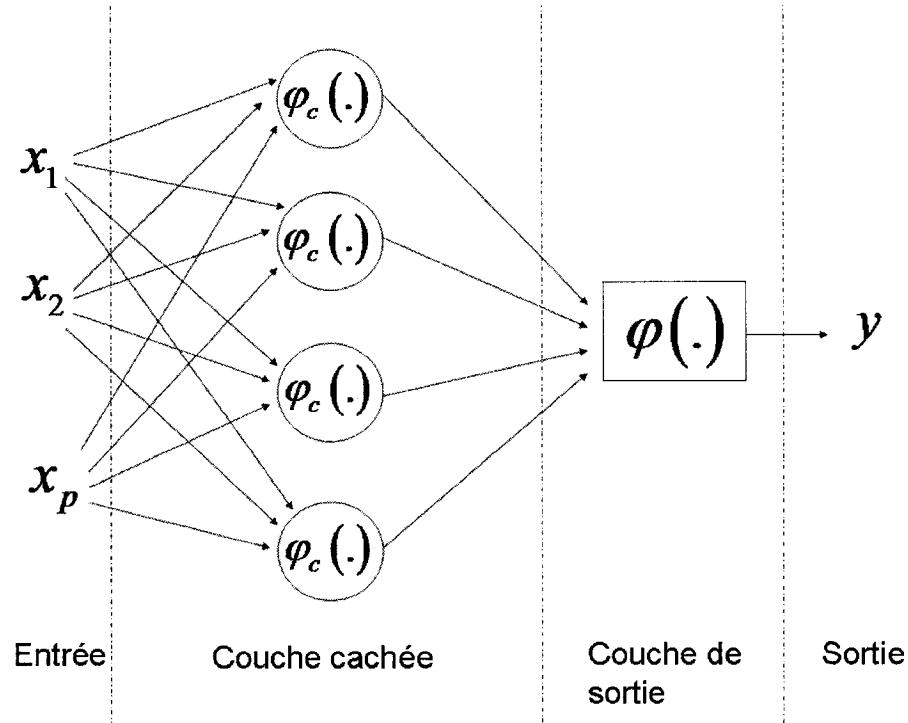


Figure 2.3 Exemple de MLP à une couche cachée

Le champ induit d'un neurone k sur la couche i du réseau est :

$$\nu_k^i = \sum_{j=1}^p \omega_{kj}^i x_j^i \quad (2.3)$$

où les x_j^i sont les entrées de la couche i provenant de la couche j et les ω_{kj}^i les poids associés de cette même couche.

Les sorties de chaque couche sont alors :

$$y_k^i = \varphi(\nu_k^i + b_k) \quad (2.4)$$

De manière générale, un MLP utilise la même fonction d'activation (de préférence non linéaire) pour tous ses neurones. On note cependant que, dans le cas où le MLP sert à une tâche de régression, la couche de sortie sera choisie linéaire.

Les poids et les biais du réseau, dits paramètres libres du réseau, déterminent le comportement du réseau, il convient donc de bien les fixer en fonction du problème à résoudre. Le processus qui a pour finalité de déterminer les paramètres optimaux s'appelle l'apprentissage.

2.3 Apprentissage des réseaux de neurones

Une des caractéristiques les plus significatives des réseaux de neurones est leur capacité d'apprendre de leur environnement et d'améliorer leur performance.

L'apprentissage consiste à modifier la valeur des paramètres libres du réseau de neurones, à savoir les biais et les poids synaptiques de chaque neurone, car ces paramètres déterminent entièrement le comportement du réseau comme il a été énoncé précédemment. Le processus d'apprentissage modifie les paramètres libres selon une loi donnée. Un réseau de neurone est ainsi typiquement un système qui s'adapte à son environnement.

On distingue deux types d'apprentissage : l'apprentissage supervisé et celui non supervisé. Dans le cadre d'un apprentissage supervisé, un ensemble de données d'entrée et de sortie désirée associée est fournie au réseau de neurones et le processus d'apprentissage vise à minimiser l'erreur entre la sortie réelle du réseau et la sortie désirée. En l'absence de sortie désirée, on parle d'apprentissage non supervisé. L'apprentissage supervisé peut être associé à une fonction d'identification

(trouver une loi d'entrée sortie) tandis que l'apprentissage non supervisé est plus volontiers associé à une fonction de contrôle.

Il existe également deux modes d'apprentissage : l'apprentissage “hors ligne” et l'apprentissage “en ligne”. Lorsqu'un ensemble de données disponible pour l'entraînement a été préalablement obtenu à partir de l'environnement du réseau de neurones, cet ensemble est présenté dans sa totalité en une seule fois (“batch learning”) ou un point à la fois (“sequenital learning”) au réseau de neurones. Les phases d'entraînement et de simulation sont donc distinctes, c'est un apprentissage “hors-ligne”. Lorsque cette information n'est pas disponible, le réseau peut être entraîné au cours d'une simulation pendant laquelle il incorpore les nouveaux points. L'entraînement et la simulation étant simultanées, c'est un apprentissage “en-ligne”.

Généralement l'apprentissage d'un réseau de neurones est basé sur la minimisation de l'erreur entre la sortie réelle de celui-ci et une réponse désirée, qu'elle soit donnée dans le cas supervisé ou générée dans le cas non supervisé. Cette minimisation est obtenue grâce à l'application d'un algorithme, le plus répandu étant l'algorithme de rétropropagation.

2.3.1 L'algorithme de rétropropagation

Le principe de l'algorithme est de rétropropager un vecteur d'erreur afin d'adapter les poids du réseau à la fonction d'identification à laquelle il est dédié.

Le signal d'erreur à la sortie du neurone k à l'instant d'itération n est :

$$e_k(n) = d_k(n) - y_k(n) \quad (2.5)$$

où y_k est la sortie à l'instant n du neurone k et d_k la réponse désirée.

On définit alors l'énergie instantanée de l'erreur à l'instant n par :

$$\zeta(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad (2.6)$$

où C est l'ensemble des neurones de la couche de sortie du réseau.

L'énergie d'erreur moyenne peut alors être calculée :

$$\zeta_{moy} = \frac{1}{N} \sum_{n=1}^N \zeta(n) \quad (2.7)$$

Celle-ci peut être interprétée comme une fonction de coût qui traduit la performance de l'apprentissage. En effet l'objectif de l'apprentissage est d'ajuster les paramètres libres du réseau, à savoir poids synaptiques et biais, afin de minimiser ζ_{moy} .

L'algorithme de rétropropagation est basée sur une approximation similaire à celle de l'algorithme des moindres carrés. En effet on estime que la correction à appliquer aux poids synaptiques ω_{kj}^i de la couche i est proportionnelle à la dérivée partielle de l'énergie instantanée par rapport au poids synaptique :

$$\Delta \omega_{kj}^i = -\eta \frac{\partial \zeta(n)}{\partial \omega_{kj}^i} \quad (2.8)$$

η est le taux d'apprentissage de l'algorithme de rétro-propagation.

L'associativité des dérivées partielles permet d'obtenir la loi suivante :

$$\Delta \omega_{kj}^i = \eta \delta_k^i(n) y_k^i(n) \quad (2.9)$$

où δ_k^i est le gradient local de la couche i et est défini par :

$$\delta_k^i(n) = -\frac{\partial \zeta(n)}{\partial \nu_{kj}^i} = e_k^i(n) \varphi_k^i(\nu_k^i(n)) \quad (2.10)$$

Si i est une couche de sortie, l'erreur se calcule directement d'après la relation 2.5.

Sinon on peut montrer (Haykin, 1999) que :

$$e_k^i(n) = \sum_j \delta_j^{i+1}(n) \omega_{jk}^{i+1}(n) \quad (2.11)$$

L'application de l'algorithme de rétropropagation pour l'entraînement d'un réseau de neurones est résumée dans le tableau 2.1. La fonction d'activation choisie pour les couches cachées est généralement la fonction sigmoïde ou tangente hyperbolique (voir figure 2.2). Les fonctions d'activation de la couche de sortie sont choisies linéaires pour une tâche de régression.

Tableau 2.1 Algorithme de rétropropagation

1. Initialisation du réseau

Choisir aléatoirement les poids synaptiques et les biais dans une distribution uniforme de moyenne nulle et de variance telle que les champs induits des neurones se trouvent dans la partie non saturée des fonctions d'activation sigmoïdales.

2. Présenter les entrées $\vec{x}(n)$ au réseau, calculer les sorties correspondantes et le vecteur d'erreur $\vec{e}(n)$ de la couche de sortie.

3. La propagation :

calculer le champ induit local de chaque neurone k de chaque couche i :

$$\nu_k^i(n) = \sum_{j=0}^{m_k} \omega_{kj}^i(n) y_j^{i-1}(n)$$

4. La rétro-propagation :

calculer les gradients locaux de chaque neurone $\delta_k^i(n)$:

$$= e_k^i(n) \varphi_k^i(\nu_k^i(n)) \text{ si } i \text{ est une couche de sortie}$$

$$= \varphi_k^i(\nu_k^i(n)) \sum_{j=0}^{m_{k+1}} \omega_{kj}^{i+1}(n) y_j^{i-1}(n) \text{ sinon}$$

puis mise à jour des poids synaptiques :

$$\omega_{kj}^i(n+1) = \omega_{kj}^i(n) + \eta \delta_k^i(n) y_j^{i-1}(n)$$

où η est le taux d'apprentissage.

5. Itération des étapes 2 à 4 jusqu'à avoir rencontré un critère d'arrêt

(par exemple fonction coût ζ inférieure à ϵ)

2.3.2 Améliorations de l'algorithme de rétropropagation

Les méthodes d'amélioration de l'algorithme de rétropropagation sont basées sur le développement en série de Taylor de la fonction coût :

$$\zeta_{moy}(\omega(n) + \Delta\omega(n)) = \zeta_{moy}(\omega(n)) + g^T(n) \Delta\omega(n) + \frac{1}{2} \Delta\omega^T(n) H(n) \Delta\omega(n) + o(\omega) \quad (2.12)$$

où $g(n)$ est le gradient local et $H(n)$ la matrice Hessienne locale:

$$\begin{aligned} g(n) &= \frac{\partial \zeta_{moy}(\omega)}{\partial \omega} \Big|_{\omega=\omega(n)} \\ H(n) &= \frac{\partial^2 \zeta_{moy}(\omega)}{\partial \omega^2} \Big|_{\omega=\omega(n)} \end{aligned} \quad (2.13)$$

Dans l'algorithme de rétropropagation, la méthode de la plus grande pente est utilisée (voir équation 2.8). Cette méthode a l'avantage d'être simple du point de vue du calcul, cependant son temps de convergence est lent. Afin d'accélérer la convergence d'un MLP on utilise de l'information de plus haut degré que celle fournie par le gradient local. L'approximation de Newton permet de calculer l'optimum de la fonction coût :

$$\Delta\omega^*(n) = -H^{-1}(n) g(n) \quad (2.14)$$

La convergence vers l'optimum se fait ainsi en une seule étape. Cependant elle implique l'existence de l'inverse du Hessien. D'autre part, même s'il existe, le calcul de celui-ci nécessite beaucoup de mémoire. Une alternative consiste à approximer le Hessien plutôt que de le calculer. C'est ce qui est fait dans l'algorithme de Levenberg-Marquardt présenté ci-après.

2.3.3 L'algorithme de Levenberg-Marquardt

L'algorithme classique de rétropropagation ajuste les poids dans la direction de la plus grande pente (gradient négatif). C'est la direction dans laquelle la fonction d'erreur décroît le plus rapidement, cependant cela n'entraîne pas nécessairement la convergence la plus rapide. La méthode de Newton assure une convergence accélérée, mais nécessite le calcul de l'inverse du Hessien, qui est complexe et peut être mal conditionné. Afin de s'affranchir de cette difficulté, l'algorithme de Levenberg-Marquardt propose d'approximer le Hessien plutôt que de le calculer :

$$\begin{aligned} g_n &= J_n^T(\vec{\omega}) \zeta_n(\vec{\omega}) \\ H_n &= J_n^T(\vec{\omega}) J_n(\vec{\omega}) + \mu I \end{aligned} \tag{2.15}$$

où $J_n(\omega)$ est la Jacobienne à l'instant n , ζ_n est la valeur de la fonction coût à l'instant n et μ un scalaire.

On remarque que si μ est grand alors c'est équivalent à la descente du gradient avec un taux d'apprentissage faible et si μ est nul alors c'est la méthode de Newton-Gauss en utilisant une approximation du Hessien. L'idée derrière l'algorithme de Levenberg-Marquardt est que la méthode de Newton est plus rapide et plus précise dans le voisinage d'un minimum, il faut donc tendre vers cette méthode le plus rapidement possible : μ est donc diminué si l'étape a conduit à une baisse de la valeur de la fonction coût, et μ est augmenté si à l'inverse la fonction coût augmente. Ainsi la valeur de la fonction coût diminue à chaque itération. Une itération est appelée une **époque** d'entraînement.

Le tableau 2.2 illustre cet algorithme.

L'apprentissage d'un réseau de neurone par l'algorithme de Levenberg-Marquardt est très sensible à l'initialisation des poids des neurones. Une mauvaise initialisation

du réseau peut conduire à ce que l'optimum trouvé ne soit qu'un optimum local. Dans le cadre d'un entraînement hors ligne, cette situation peut être évitée par l'application de la validation croisée entre différents modèles ayant des paramètres d'initialisation différents.

Tableau 2.2 Algorithme de Levenberg-Marquardt

1. Présenter les entrées $\vec{x}(n)$ au réseau, calculer les sorties correspondantes et le vecteur d'erreur $\vec{e}(n)$. Calculer la fonction de coût:

$$\zeta_n(\vec{\omega}) = \sum_{k=1}^N e_k^{(n)}(\vec{\omega})$$

2. Calculer la matrice jacobienne $J_n(\vec{\omega})$

3. Mettre à jour les poids :

$$\vec{\omega}_{n+1} = \vec{\omega}_n - (J_n^T(\vec{\omega}_n) J_n(\vec{\omega}_n) + \mu_n I) J_n^T(\vec{\omega}_n) \zeta_n(\vec{\omega}_n)$$

4. Calculer $\zeta_n(\vec{\omega}_{n+1}) = \sum_{k=1}^N e_k^{(n)}(\vec{\omega}_{n+1})$

Si $\zeta_n(\vec{\omega}_{n+1}) < \zeta_n(\vec{\omega}_n)$ alors $\mu_{n+1} = \mu_n - \nu$, où ν est une constante, puis retourner à l'étape 1.

Si $\zeta_n(\vec{\omega}_{n+1}) > \zeta_n(\vec{\omega}_n)$ alors $\mu_{n+1} = \mu_n + \nu$, et retourner à l'étape 3 pour recalculer $\vec{\omega}_{n+1}$ avec la nouvelle valeur de μ_n .

5. Itération des étapes 2 à 4 jusqu'à avoir rencontré un critère d'arrêt (par exemple fonction coût ζ inférieure à ϵ)

2.4 Optimisation de l'apprentissage

La performance finale d'un réseau de neurones suite à l'apprentissage n'est pas indépendante des données fournies pour l'entraînement, de son initialisation ni du nombre d'époques sur lequel il est entraîné. Le processus de validation croisée permet de choisir un réseau, pour une architecture donnée, dont les performances sont optimales. Le phénomène de sur-apprentissage sera quant à lui minimisé par un arrêt prématuré de l'apprentissage.

2.4.1 Données d'entraînement

Les réseaux de neurones sont non linéaires. Cela impose une excitation suffisante du signal d'entrée pour une identification correcte sur l'ensemble de l'espace d'entrée. Plus la dimension de la couche d'entrée est grande, plus la taille de l'ensemble des données d'entraînement doit être grande, ceci est connu sous le nom de “curse of dimensionality”. De plus, il est préférable que deux échantillons d'une même variable successivement fournies au réseau varient passablement (Haykin, 1999).

Afin de rendre l'entraînement le plus efficace possible, il est préférable de traiter au préalable les données.

Si des signaux mesurés sont utilisés comme entrée du réseau, il est recommandé (Nørgaard, Ravn, Poulsen & Hansen, 2000) de les filtrer pour éliminer le bruit de mesure, ou le biais introduit par les appareils de mesure. En l'absence de filtrage, l'entraînement du réseau de neurone est délicat. Un entraînement excessif implique que l'erreur d'entraînement est très faible mais alors le réseau suit le bruit et non le signal.

L'ensemble des données doit être représentatif de la fonction à identifier, les données se doivent d'être équitablement réparties sur l'ensemble de l'enveloppe de vol. En effet si les données sont sur-représentatives d'une région de l'espace d'entrée, le réseau apprend de manière fidèle celle-ci, au détriment d'une bonne approximation en dehors de cette région. De plus, éliminer la redondance implique un temps d'entraînement réduit.

Les fonctions d'activation étant généralement symétriques, il est préférable de prétraiiter les données de sorte que la moyenne de chaque variable soit nulle. Si les amplitudes des variables d'entrées sont très disparates, il convient également d'uniformiser leur variance, sinon la variable ayant la plus grande plage de valeurs se voit attribuer une importance artificielle.

2.4.2 La validation croisée

Le principe de la validation croisée est basé sur la cohérence de la comparaison des différents modèles de réseaux de neurones. Afin de pouvoir comparer la performance de réseaux de neurones, une partie des données est prélevée à des fins de test. En effet, en testant l'ensemble des modèles avec des exemples qui n'ont pas servi à leur entraînement, on peut interpréter l'erreur moyenne de chaque modèle comme un indicateur d'optimalité : plus cette erreur sera faible, plus le réseau s'approchera de la fonction de régression cherchée. En pratique il est conseillé dans Haykin (1999) de conserver 20% des données à disposition pour la validation croisée, cet ensemble de données est alors appelé **ensemble de validation**. Ce principe ne s'applique toutefois que dans le cas d'apprentissage hors-ligne.

Soit un ensemble de réseaux de neurones possédant la même architecture mais ayant des paramètres initiaux distincts. Lorsque l'algorithme d'apprentissage est dépendant de l'initialisation des paramètres comme dans le cas de la rétropropagation ou de l'algorithme de Levenberg-Marquardt, la validation croisée permet de choisir l'initialisation la meilleure. Une autre application directe de la validation croisée est le choix de l'architecture du réseau de neurones la mieux adaptée à un problème donné de régression ou de classification.

2.4.3 L'arrêt prématûr de l'apprentissage

Un paramètre à fixer lors de l'apprentissage d'un réseau de neurones est le nombre d'époques d'apprentissage, c'est-à-dire le nombre de fois que l'on va présenter pour entraînement les données à un réseau de neurones.

Plus le nombre d'époques est grand, plus le réseau apprend les exemples fournis.

Donc l'erreur moyenne quadratique calculée sur les exemples d'entraînement tend vers zéro quand le nombre d'époques augmente. Cependant cet apprentissage se fait au détriment de la généralisation, c'est-à-dire que l'erreur du réseau pour des valeurs non utilisées à l'entraînement (dite erreur de généralisation) sera plus importante que prévue. Afin de contrer le sur-apprentissage, l'apprentissage d'un réseau de neurones sera arrêté prématulement. À cette fin une partie des données est prélevée pour servir d'**ensemble de test**, tandis que le réseau de neurones est entraîné sur toutes les données restantes. L'époque d'entraînement optimale correspond à l'époque pour laquelle l'erreur quadratique de l'ensemble de test est la plus faible (voir figure 2.4). L'arrêt prématuclé de l'apprentissage à cette époque permet d'obtenir une meilleure performance du réseau de neurones.

L'ensemble des données fournies pour l'entraînement, privé de celles sélectionnées pour les ensembles de validation et de test, est appelé **ensemble d'entraînement**.

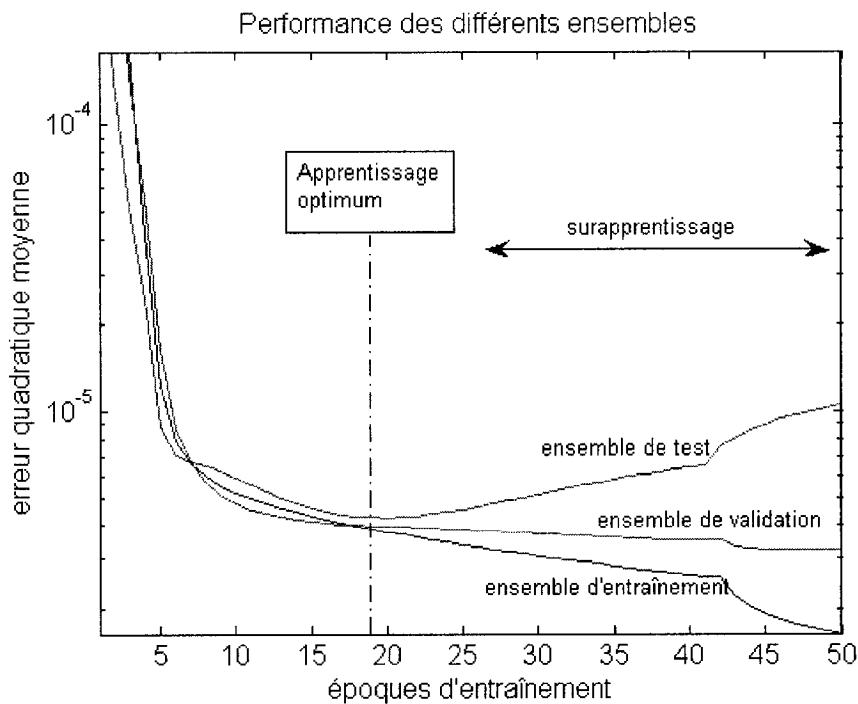


Figure 2.4 Arrêt prématuclé de l'entraînement

CHAPITRE 3

IDENTIFICATION DU MODÈLE AÉRODYNAMIQUE

L'objectif du processus d'identification est d'obtenir un modèle aérodynamique qui représente de manière précise et fidèle les caractéristiques aérodynamiques de l'avion, en conformité aux spécifications requises par une commande de vol non linéaire. La méthode utilisée ici est une identification par réseaux de neurones. Après avoir mis en évidence l'avantage des réseaux de neurones par rapport à des méthodes classiques, la conception d'un module d'identification hors ligne est présentée. L'identification en ligne ayant un inconvénient majeur, une méthode récente pour minimiser celui-ci est étudiée et mène à proposer une architecture novatrice adaptée aux deux types d'apprentissage.

3.1 Identification hors ligne et en ligne

L'identification hors ligne est une analyse post expérience, au sens où les données mesurées au cours d'expériences (par exemple des essais en soufflerie statiques et dynamiques) sont utilisées une fois les expériences achevées. Les mesures sont ap- prises dans leur ensemble par des algorithmes tels que celui des moindres carrés. A l'issue de ce processus, une loi entrée-sortie est déterminée permettant de car- tographier les valeurs des coefficients aérodynamiques sur l'ensemble de l'enveloppe de vol. Cependant avec les méthodes classiques (algorithme des moindres carrés, filtrage de Kalman, ...) le modèle aérodynamique ainsi obtenu est statique, et ne pourra ni être amélioré en vol, ni s'adapter à des changements pouvant survenir dans la dynamique de l'avion.

Une application importante de l'identification en ligne, ou encore temps réel, est de caractériser les changements de la dynamique d'un avion dans le cadre d'une commande de vol adaptative ou reconfigurable. Les variations des coefficients aérodynamiques de l'avion représentent les changements de condition de vol, la consommation de carburant, un changement de configuration de l'avion, ou encore une large gamme de pannes, d'usure ou de dégât survenant sur l'avion (Klein & Morelli, 2006). L'objectif est alors d'estimer en temps réel ces coefficients à partir de données mesurées, de sorte que la logique du contrôleur adaptatif puisse effectuer les changements nécessaires à la loi de commande pour assurer la stabilité de l'aéronef et les objectifs de performance. Classiquement cette identification est faite en considérant un modèle linéaire à paramètres variant dans le temps de l'aérodynamique de l'avion comme présenté en 1.1.3.1.

Les méthodes d'estimation en temps réel telles que l'algorithme des moindres carrés récursif (Ward, Monaco & Bodson, 1998) ou du filtrage de Kalman étendu (Jategaonkar, 2006) sont couramment utilisées. Les avantages principaux de ces méthodes sont leur capacité à estimer en temps réel les paramètres aérodynamiques de l'avion sans avoir à utiliser l'ensemble des données additionné de chaque nouvelle mesure, et leur capacité à suivre des paramètres variant dans le temps. Cependant elles présentent des inconvénients majeurs :

- un modèle linéaire sous-jacent à paramètres variables dans le temps doit être fixé.
- les connaissances a priori, obtenues lors des essais en soufflerie ou des essais en vol, ne sont pas exploitées par ces algorithmes récursifs d'identification.
- dans des conditions de vol normales, les variables de contrôles et d'état sont relativement constantes pendant de longues périodes. Pendant celles-ci l'excitation du système dynamique est proche du niveau de bruit, or dans

de telles circonstances les méthodes précédentes de régression donnent des résultats imprécis.

Ainsi l'identification hors ligne des paramètres aérodynamiques d'un aéronef permet d'obtenir un modèle sur l'ensemble de l'enveloppe de vol, mais celui-ci n'est pas capable d'incorporer de nouvelles données ni de s'adapter si la modélisation effectuée n'est pas suffisamment précise. L'identification en temps réel est capable d'une telle adaptation, mais présente l'inconvénient de n'être valable que pour le point de fonctionnement auquel se trouve l'avion : aucune donnée connue a priori ne peut être exploitée par ces méthodes.

Les réseaux de neurones sont adaptés aux deux types d'identification, comme le souligne Richardson (1999). Grâce à leur capacité d'apprentissage, ils peuvent apprendre hors-ligne la loi entrée-sortie induite par les mesures récoltées au cours d'expérience et de simulations ; l'utilisation de l'apprentissage en ligne leur permet d'apprendre de nouvelles informations acquises au cours du vol.

3.2 Les réseaux de neurones appliqués à l'identification hors ligne

L'objectif de ce mémoire est de concevoir une loi de commande non linéaire valable sur l'ensemble de l'enveloppe de vol. Celle-ci nécessite une modélisation du système sur ce même ensemble, à savoir d'après le chapitre 1 identifier le modèle aérodynamique de l'avion. Il a été souligné que celle-ci pouvait se faire hors ligne par des algorithmes d'interpolation ou de régression à partir de données issues de l'expérience et fournies sous forme de tables de références (Jategaonkar, 2006).

Les avantages des réseaux de neurones par rapport à ces méthodes sont :

- l'espace mémoire est réduit, car toute l'information connue a priori est comprise dans la valeur des poids synaptiques.
- les réseaux de neurones ont la capacité d'interpoler les données sans effort de calcul supplémentaire.
- il n'y a pas à définir préalablement la structure inhérente à la loi entrée-sortie donnant les coefficients aérodynamiques, ce qui permet de conserver intacte l'information contenue dans les mesures notamment au niveau de la non linéarité et du couplage entre les coefficients.

Cependant la précision de l'approximation est fonction de l'architecture du réseau, à savoir le type de réseau utilisé, son nombre de neurones et de couches cachées, la valeur de ses paramètres et le choix de ses entrées et sorties. Tous ces paramètres doivent donc être déterminés. Un objectif de performance usuel (Richardson, 1999) est d'estimer les coefficients aérodynamiques avec une erreur quadratique moyenne d'entraînement inférieure à $2,5 \cdot 10^{-4}$, cette valeur sera prise comme référence pour ce mémoire.

Un réseau de neurones MLP à une couche cachée est capable d'approximer toute fonction non linéaire complexe avec une précision aussi fine que voulue à condition de présenter un nombre suffisant de neurones sur sa couche cachée (Haykin, 1999). De plus, dans le cas où un réseau doit effectuer une régression, il est usuel d'utiliser une fonction d'activation sigmoïdale symétrique sur la couche cachée et une fonction linéaire sur la couche de sortie. Les méthodes d'optimisation de l'entraînement (voir 2.4) permettent alors avec un nombre de neurones fixés d'obtenir un réseau optimal.

Tableau 3.1 Dépendances des coefficients aérodynamiques

Modèle	Coefficients	Variables
Longitudinal	C_x, C_m, C_z	$\alpha, \beta, \delta_e, q, V_t, (\delta_{lef})$
Latéral	C_l, C_y, C_n	$\alpha, \beta, \delta_a, \delta_r, p, r, V_t, (\delta_{lef}, \delta_e)$

En ce qui concerne le choix des entrées, le modèle aérodynamique établi en 1.1.3.2 met en évidence les dépendances entre les coefficients aérodynamiques et certains paramètres. Celles-ci sont récapitulées dans le tableau 3.1, où les paramètres entre parenthèses sont ceux particuliers au F-16. Un choix possible est soit de concevoir un unique réseau identifiant tous les coefficients, soit de concevoir deux réseaux, le premier identifiant le modèle aérodynamique longitudinal et le second le modèle latéral.

Il est fortement recommandé (Haykin, 1999) de ne pas avoir de poids synaptique nul dans un réseau, car une telle configuration implique que le neurone ajoute du bruit dans le réseau. Or les modèles longitudinal et latéral sont découplés par rapport à certaines variables, notamment les vitesses angulaires et les déflections des surfaces de contrôle. On peut donc craindre que de prendre comme sortie les six coefficients aérodynamiques, et donc l'ensemble des variables présentes dans le tableau 3.1 comme entrée, entraîne la nullité de certains poids synaptiques. Cette configuration n'est pas retenue : on adopte donc deux réseaux.

Ainsi le seul paramètre qui reste à déterminer est le nombre de neurones sur la couche cachée de chacun des réseaux. Cependant afin d'être efficacement utilisé par le contrôleur non linéaire, le module d'identification doit produire une valeur estimée de chaque coefficient aérodynamique le plus rapidement possible. Ce délai étant proportionnel au nombre de neurones sur la couche cachée, les objectifs de conception sont de minimiser le nombre de neurones sur celle-ci et d'atteindre la précision d'estimation requise.

Afin de minimiser le nombre de neurones, deux méthodes existent. La première, dite de “pruning”, consiste à créer un réseau avec un nombre de neurones très élevé et par inspection de retirer les neurones dont les poids synaptiques en amont ou en aval sont nuls. Seulement il n'y a pas de garantie que le réseau final obtenu satisfasse encore le critère de performance fixé initialement. La seconde méthode, dite de “growth”, consiste à augmenter progressivement le nombre de neurones d'un réseau jusqu'à obtenir le critère de performance requis. C'est celle adoptée.

L'utilisation de l'algorithme de Levenberg-Marquardt (voir 2.3.3) implique une convergence accélérée d'un réseau de neurones, et a fortiori un coût et un temps de conception réduits. Toutefois cet algorithme, comme la majorité des algorithmes par paquets (ou “batch”), est sensible à l'initialisation des poids synaptiques lors de la création du réseau de neurones. À nombre fixé de neurones sur la couche cachée, la validation croisée permet de choisir le réseau présentant les meilleures performances de généralisation.

La combinaison de la méthode de “growth” et de l'utilisation de l'algorithme de Levenberg-Marquardt permet d'obtenir une procédure pour la conception d'un réseau de neurone optimal au sens où le nombre de neurones est minimisé et le critère de performance est atteint. Cette procédure est récapitulée dans le tableau 3.2.

On constate que le nombre de neurones nécessaires pour identifier le modèle aérodynamique sur l'ensemble de l'enveloppe de vol avec le critère de performance usuel est important : Richardson (1999) obtient plus de 130 neurones sur la couche cachée pour l'identification de C_z ! Le partitionnement de l'espace d'entrée (voir 1.1.3.2) permet de considérer des comportements de sortie moins complexes. Le nombre de neurones sur la couche cachée de chacun des réseaux assignés à une partition serait donc plus restreint. L'application d'un processus d'interpolation ou de lissage aux frontières des partitions permet de plus de conserver le caractère continu de la

sortie fournie par le module d'identification.

Tableau 3.2 Procédure de conception hors ligne d'un réseau optimal

1. Définition des paramètres

ϵ : Critère de performance

m_1 : nombre initial de neurones sur la couche cachée

N : nombre de réseaux à comparer

p : nombre d'époques d'entraînement

2. Entraînement des réseaux de neurones

Pour $i = 1..N$

créer et initialiser un nouveau réseau à m_1 neurones sur la couche cachée

entraîner le réseau sur p époques

sauvegarder le réseau et ses performances

fin

Si l'erreur quadratique moyenne de l'ensemble de test n'est pas inférieure à celle obtenue pour $m_1 - 1$ neurones, entraîner le réseau à nouveau.

Réitérer jusqu'à obtenir une erreur quadratique plus faible.

3. Analyse des réseaux obtenus

Déterminer le réseau optimal de ce modèle par validation croisée.

Vérifier si les objectifs de performance ont été atteints, si oui c'est le réseau optimal. Si non, augmenter le nombre de neurones de la couche caché et retourner à l'étape 2.

Bien qu'à première vue partitionner l'espace des variables d'entrée du module d'identification semble être une démarche analogue à celle faite par le "gain scheduling", la démarche intrinsèque du partitionnement est bien différente. En effet lors du "gain scheduling", l'enveloppe de vol est discrétisée en un ensemble de points où pour chacun d'eux un modèle de l'avion est élaboré ainsi qu'un contrôleur linéaire local ; la programmation des gains permet alors de couvrir l'ensemble de l'enveloppe de vol. Le partitionnement quant à lui permet d'optimiser la tâche d'identification.

L'architecture interne du module d'identification du modèle aérodynamique de l'avion est certes un ensemble de réseaux de neurones spécialisés dans des régions différentes de l'espace des variables d'entrée, mais d'un point de vue boîte noire il identifie l'ensemble de l'enveloppe de vol de manière performante et continue.

Ainsi une procédure générale pour la conception d'un module d'identification a été mise en évidence. Elle comporte trois étapes majeures à appliquer à l'identification du modèle longitudinal et latéral : le partitionnement de l'espace d'entrée, la conception pour chacune des partitions d'un réseau optimal selon l'algorithme du tableau 3.2 et le choix d'un processus d'interpolation ou de lissage pour obtenir une sortie globale continue.

3.3 Extension vers un entraînement en ligne

Bien que la commande de vol adaptative conçue dans le cadre de ce projet ne soit pas développée en tant que commande reconfigurable, le lien étroit établi dans l'introduction entre ces deux lois de commande suggère une possible extension à la reconfiguration suite à une panne ou un dégât survenant sur l'avion. Le module d'identification est donc développé de sorte à satisfaire, dans une certaine mesure, les exigences de cette éventuelle évolution. On veillera notamment à ce que le module d'identification puisse s'adapter à des changements rapides des valeurs des coefficients aérodynamiques, que ce soit par exemple suite à un changement brutal de condition de vol ou suite à une variation des coefficients. Cet aspect de l'apprentissage n'est cependant pas traité de manière extensive.

Une fois le module d'identification développé hors ligne, il est implanté dans l'avion et peut être programmé pour apprendre en ligne. Les capteurs fournissent des données mesurées à la fois pour les entrées et les sorties désirées des réseaux de

neurones. Le module d'identification du modèle aérodynamique de l'avion utilise alors ces mesures pour entraîner le réseau en ligne. Cela lui permet d'améliorer les estimations initiales obtenues suite à l'apprentissage hors ligne ou bien de s'adapter à des changements survenants dans les valeurs des coefficients aérodynamiques. Comme tout algorithme d'estimation, les réseaux de neurones nécessitent une excitation suffisante en entrée pour obtenir des résultats globaux. Malheureusement, dans des conditions normales de vol, il existe des longues périodes de temps pendant lesquelles la configuration de l'avion est quasi constante. Cela implique une absence d'excitation en entrée des réseaux de neurones du module d'identification. En s'entraînant en ligne dans de telles conditions les réseaux s'adaptent de manière précise à la condition de vol courante, mais oublient la cartographie des coefficients aérodynamiques dans les autres régions de l'enveloppe de vol.

Richardson (1999) suggère d'utiliser un réseau de neurone, appelé "nominal", pour le modèle aérodynamique de l'avion dans ses conditions normales, et de l'étendre par un deuxième réseau qui corrigerait les erreurs du modèle grâce à l'apprentissage en ligne. À intervalles réguliers le réseau nominal est entraîné avec les données du réseau correcteur. Toutefois le réseau correctif est un réseau qui croît en ligne, au sens où des neurones y sont ajoutés en ligne. Le dimensionnement d'un tel réseau n'est pas aisés, de plus il faut garantir que le nombre de neurones le composant n'explose pas.

Une autre méthode permettant d'éliminer ce phénomène d'oubli a récemment été développée dans le laboratoire d'aéronautique de l'université de Delft sur un modèle basse fidélité d'un avion de combat (DeWeerdt et al., 2005). Le principe est de partitionner l'espace d'entrée et d'exploiter les relations invariantes afin de créer une structure dite en hypercubes.

3.3.1 Les hypercubes de De Weerdt (2005)

La structure globale du module d'identification utilisée par De Weerdt (2005) est légèrement différente de celle exposée en 3.2. Si la loi d'entrée-sortie modélisant le comportement aérodynamique de l'avion est amenée à changer, suite par exemple à une panne ou un dégât, l'architecture du réseau de neurones reste inchangée et le nouveau modèle est appris en ligne. Toutefois la perspective d'une telle modification en ligne suppose l'élargissement de la dimension des entrées, car on ne peut exclure la possibilité que la nouvelle loi donnant par exemple la valeur de C_x ne dépende pas des taux de rotation p ou r , ou des déflections des surfaces de contrôle latérales δ_a et δ_r . De Weerdt a donc introduit ce couplage additionnel entre l'ensemble des variables d'entrée et les six coefficients aérodynamiques.

Le module d'identification de De Weerdt a ainsi les six coefficients aérodynamiques en sortie, et $[\alpha, \beta, \delta_e, \delta_a, \delta_r, V_t, h, p, q, r]$ comme vecteur d'entrée.

De Weerdt (2005) émet une hypothèse qui lui permet de réduire la dimension du vecteur des variables d'entrée, à savoir qu'il suppose que les relations 1.6 donnant les coefficients aérodynamiques totaux restent vraies après une panne structurelle pure (i.e. panne qui entraîne uniquement une modification des coefficients aérodynamiques de l'avion, sans changement de masse de l'avion). C'est-à-dire que suite à une telle panne seule change la valeur numérique des coefficients internes dans l'expression des coefficients aérodynamiques (par exemple C_{x_0} , C_{x_q} , ΔC_{x_lef} et $\Delta C_{x_lef_q}$ pour le coefficient aérodynamique C_x) et que l'expression des coefficients aérodynamiques totaux est linéaire en certaines variables :

- C_x, C_m et C_z varient linéairement avec V_t et q
- C_l, C_y et C_n varient linéairement avec V_t, p et r

Cette hypothèse permet de simplifier la structure du module d'identification du modèle aérodynamique du F-16 en réduisant la dimension du vecteur d'entrée des réseaux de neurones de neuf à cinq : $\alpha, \beta, \delta_e, \delta_a$ et δ_r . De Weerdt introduit alors le concept de point statique permettant de fixer des valeurs pour les cinq autres variables (V_t, p, q et r). Si l'invariance linéaire est effectivement conservée après une panne, alors deux valeurs par paramètre sont suffisantes. Mais seule la validation en ligne peut confirmer la validité de cette hypothèse. La vitesse réelle de l'avion ayant été introduite artificiellement dans les relations afin d'obtenir des coefficients sans dimension, la linéarité par rapport à V_t est assurée. Une telle affirmation ne peut pas être avancée pour les vitesses angulaires de l'avion. Afin de s'assurer d'une bonne performance d'approximation (en condition normale et en cas de panne) il lui semble judicieux d'utiliser 5 valeurs pour chacune des autres variables des points statiques (voir tableau 3.3).

Tableau 3.3 Valeurs des variables des points statiques

Variable	Valeurs	Unités
vitesse V_t	500; 700	$ft.s^{-1}$
vitesse angulaire de roulis p	-350; -175; 0; 175; 350	$^{\circ}.s^{-1}$
vitesse angulaire de tangage q	-100; -50; 0; 50; 100	$^{\circ}.s^{-1}$
vitesse angulaire de lacet r	-50; -25; 0; 25; 50	$^{\circ}.s^{-1}$

Pour chaque point statique, l'espace d'entrée est partitionné. De Weerdt (2005) souligne que certaines lignes directrices sont à respecter : les “hypercubes”, soit les partitions à point statique fixé, doivent être choisis les plus grands possibles afin que leur nombre soit minimal, mais suffisamment petits pour que l'effet d'oubli ne soit pas important en leur sein. Le choix exposé au tableau 3.4 est alors fait par De Weerdt. À chaque hypercube correspond un réseau de neurones ayant les mêmes entrées et sorties que le module d'identification global.

Tableau 3.4 Partitions en hypercubes

Variable	Valeurs	Unités
angle d'attaque α	[-10;1] [-1;11] [9;21] [19;31] [29;45]	degré
angle de dérapage β	[-30;-19] [-21;-9] [-11;1] [-1;11] [9;21] [19;30]	degré
déflexion d'élévateur δ_e	[-25;-10] [-14;2] [-2;14] [10;25]	degré
déflexion d'aileron δ_a	[-21.5;-10] [-14;2] [-2;14] [10;21.5]	degré
déflexion de gouvernail δ_r	[-30;-19] [-21;-9] [-11;1] [-1;11] [9;21] [19;30]	degré

Afin de mieux comprendre la notion de point statique, la figure 3.1 met en évidence le maillage de ceux-ci pour une valeur de la variable V fixée, car la représentation en dimension 4 n'est pas possible. Les points pleins sur la figure représentent la position des points statiques. A chacun d'eux correspond un ensemble d'hypercubes.

La figure 3.2 permet de visualiser, à un point statique (V_0, p_0, q_0, r_0) donné, la structure en hypercubes en limitant le nombre de variables à 3 (α , β et δ_e). Le chevauchement entre les différentes régions n'a pas été illustrée pour des questions de visibilité.

Seuls les hypercubes dont la région associée comprend la condition de vol actuelle $(\alpha, \beta, \delta_e, \delta_a, \delta_r)$ sont activés. D'ailleurs, ils se chevauchent de sorte que la transition d'une région à une autre se fasse de manière continue.

Dans le cas idéal, les régions adjacentes apprennent sans erreur la loi d'entrée-sortie locale. Celle-ci étant identique sur la plage de chevauchement pour les régions adjacentes, elles devraient fournir les mêmes sorties et une fonction de basculement suffirait pour passer d'une région à une autre au milieu de la zone de chevauchement.

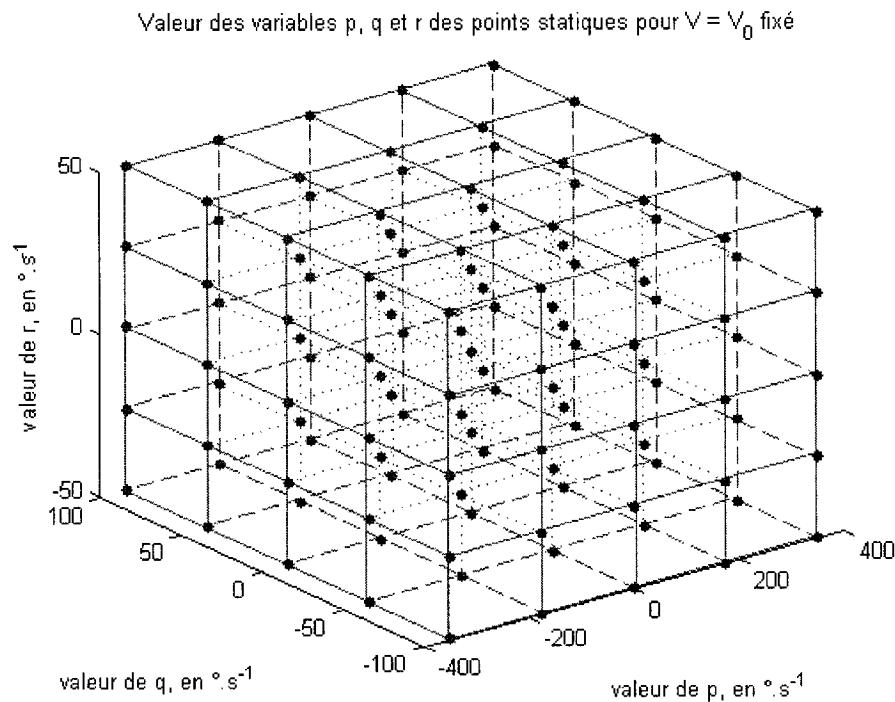


Figure 3.1 Representation partielle des points statiques

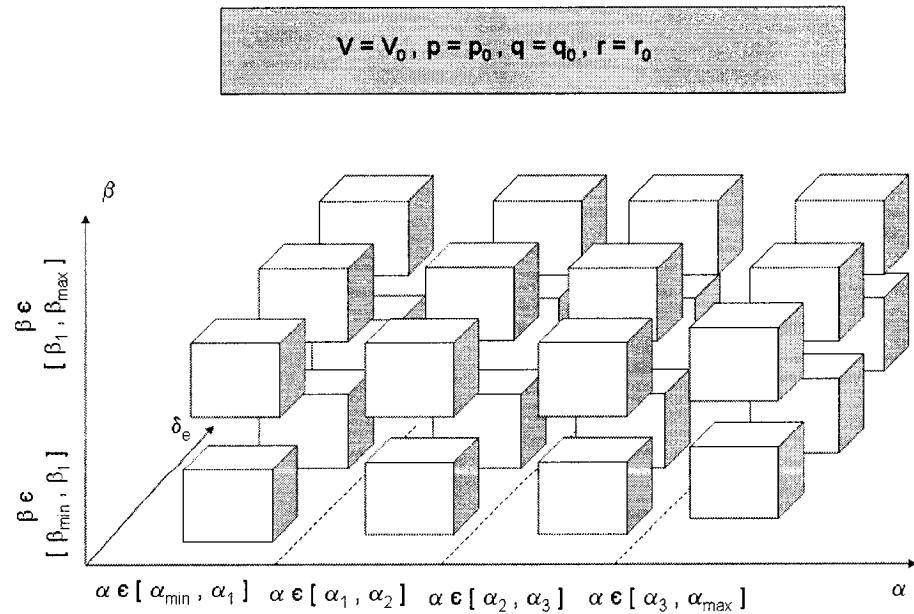


Figure 3.2 Representation partielle des hypercubes

Dans le cas non idéal, la loi d'entrée-sortie est apprise avec un facteur d'erreur, et l'utilisation d'une fonction de basculement provoquerait une discontinuité dans la sortie du module d'identification global. De Weerdt utilise alors une interpolation linéaire dans la région de chevauchement entre les valeurs des coefficients données par chacune des régions activées, soit entre deux et dix régions.

Cette procédure donne alors une valeur des coefficients aérodynamiques en chaque point statique. En général, ces valeurs doivent ensuite être interpolées de sorte à calculer la valeur des coefficients aérodynamiques pour les valeurs de (V_t, p, q, r) de la condition de vol.

En effet, si la condition de vol se trouve être un point statique, les valeurs des coefficients données par celui-ci sont directement utilisées. Par contre, si cela n'est pas le cas, selon le nombre de variables dont la valeur est celle d'un point statique il est nécessaire d'interpoler les sorties de 2 (si trois variables ont des valeurs correspondant à celles d'un point statique) à $2^4 = 16$ (si aucune variable n'a la valeur correspondant à celle d'un point statique) points statiques.

Ainsi la technique des hypercubes entraîne une complexification importante au niveau de la structure du module d'identification. Une analyse des performances globales d'un tel module à hypercubes semble indiquée.

3.3.2 Analyse critique

Le module d'identification développé par De Weerdt a pour finalité d'être utilisé par un contrôleur reconfigurable, et devra donc être programmé pour apprendre en ligne. Or les performances du module n'ont pas été vérifiées dans ce cas là. Certes la validation finale ne peut se faire qu'en ligne, mais la simulation permet d'apprécier dans une certaine mesure le comportement du module d'identification

lorsqu'il est entraîné avec des données caractéristiques d'une condition de vol ou d'une situation.

Afin de ne pas développer toute la structure en hypercubes un cas simplifié est étudié. Seule l'identification du coefficient C_x au point statique $V_t = 150m.s^{-1}$ et $q = 0^\circ.s^{-1}$ est effectuée. De plus un modèle simplifié est considéré, à savoir que le bord d'attaque n'est pas pris en considération et que le couplage supplémentaire introduit par De Weerdt n'a pas été exploité. L'espace d'entrée est donc réduit à l'angle d'attaque α , l'angle de dérapage β et la déflection de l'élévateur δ_e . La partition de cet espace est identique à celle du tableau 3.4.

Cette structure est comparée à un module qui identifie le coefficient C_x sans partitionnement de l'espace d'entrée.

Performance suite à l'apprentissage hors ligne : Les structures décrites précédemment sont développées à angle de dérapage fixé nul et à déflection de l'élévateur fixe. On obtient un module, dit à région unique, qui approxime la loi entrée-sortie sur l'ensemble de la plage d'angle d'attaque, appelé abusivement enveloppe de vol dans cette section, et un module, dit à régions multiples, qui est une application de la technique des hypercubes de De Weerdt.

Pour le module d'identification à régions multiples, les réseaux associés à chaque hypercube (ou chaque partition d'angle d'attaque α) contiennent un faible nombre de neurones (entre 4 et 6) et l'erreur quadratique moyenne d'entraînement, d'une valeur de $2,5.10^{-5}$, y est inférieure à la valeur de référence $2,5.10^{-4}$. Pour obtenir les mêmes performances en termes d'erreur, le module à région unique nécessite une capacité de calcul importante. En effet, sur un Pentium III à 450 *MHz*, un entraînement met plusieurs heures dans ce cas ci, contre une dizaine de minutes lors de l'utilisation des hypercubes. Une architecture à 17 neurones sur la couche cachée permet d'assurer une erreur d'entraînement inférieure à $6,2.10^{-5}$.

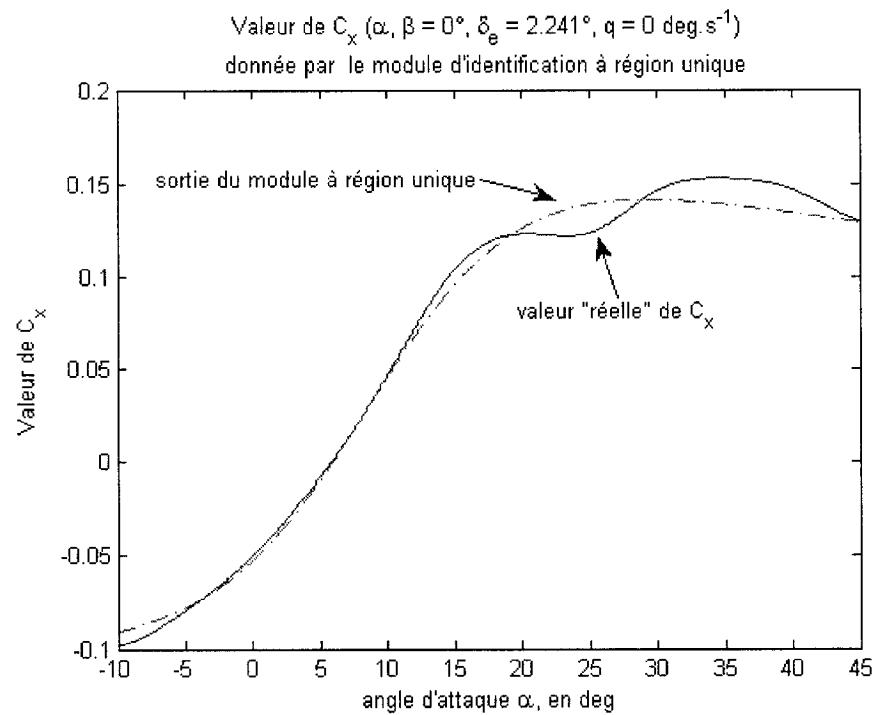


Figure 3.3 Performance du module à région unique

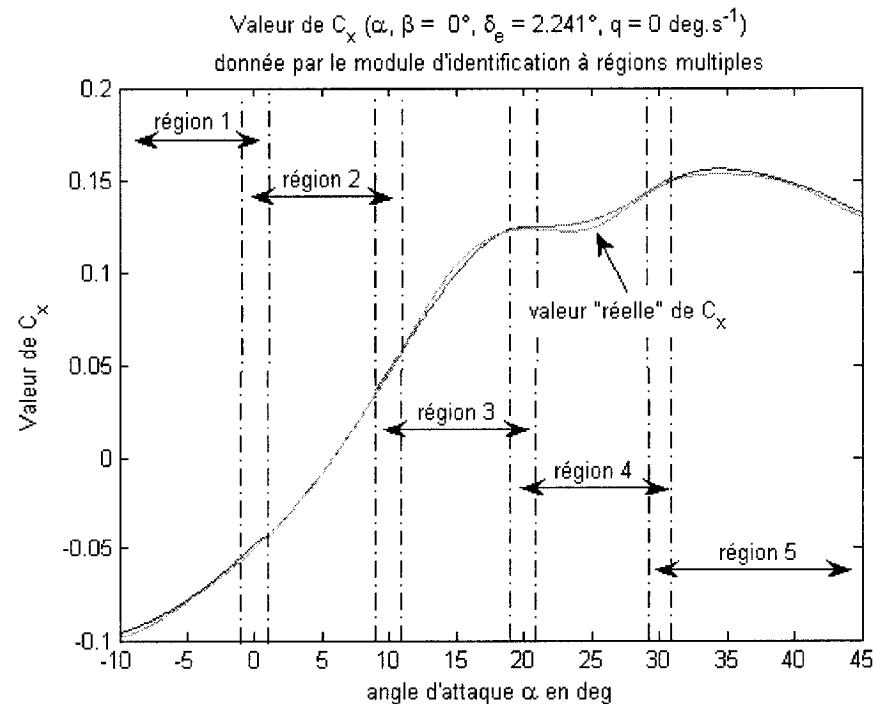


Figure 3.4 Performance du module à régions multiples

Afin de visualiser la différence entre les deux modules, les figures 3.3 et 3.4 comparant l'identification du coefficient C_x , en fonction de l'angle d'attaque α avec l'angle de dérapage et la déflexion de l'élévateur fixes : $\beta = 0^\circ$ et $\delta_e = 2,241^\circ$. La valeur dite "réelle" du coefficient C_x est obtenue par interpolation des données du rapport de la NASA (Nguyen et al, 1979).

Le module d'identification à régions multiples approxime très précisément le coefficient C_x sur toute l'étendue de l'angle d'attaque α . Le module à région unique quant à lui est performant pour des angles d'attaques inférieurs à 15° , au delà l'approximation devient relativement grossière. De plus, le temps d'apprentissage d'un réseau étant proportionnelle à sa taille, ce dernier module est le moins rapide.

Le temps d'apprentissage hors ligne est coûteux. Chaque réseau du module à régions multiples est certes plus rapide et plus simple à concevoir que le module à région unique, toutefois on dénombre 1 440 000 réseaux dans la structure à hypercubes de De Weerdt (2005). Il y a donc un compromis à faire entre finesse du partitionnement et précision de l'approximation.

Performance suite à l'apprentissage en ligne : une étude qualitative est menée afin de dégager certaines hypothèses quant à la performance des modules d'identification suite à l'apprentissage en ligne. Le scénario décrit dans le tableau 3.5 a été exploité dans de nombreuses configurations. Les résultats obtenus pour une modification semblable de la condition de vol, mais à condition initiale différente, sont similaires. Il semblerait qu'ils soient généralisables, toutefois seule une analyse extensive permettrait de valider cette hypothèse.

Afin de simplifier l'interprétation des courbes obtenues, un scénario où seul l'angle d'attaque est amené à changer est considéré. Les autres paramètres sont fixés : l'angle de dérapage β est nul, la déflexion de l'élévateur est égale à $2,241^\circ$, et la vitesse angulaire de tangage est nulle.

Tableau 3.5 Scénario général pour l'apprentissage en ligne

-
- 1. Choisir un module d'identification** du modèle aérodynamique, dont les performances suite à l'entraînement hors ligne sont satisfaisantes.
-
- 2. Générer l'ensemble des couples** $(\alpha, \beta, \delta_e, q)$ caractérisant la trajectoire :
- $0 < t < t_1$: vol horizontal stabilisé à altitude constante ($h = 5715m$) et à vitesse constante ($V = 190m/s$).
- $t_1 < t < t_2$: changement de condition de vol.
- $t_2 < t$: condition de vol constante.
-
- 3. Simulation du module d'identification** : un vecteur d'entrée à la fois est simulé. Le vecteur d'entrée est ajoutée à une FIFO dédiée et la valeur de C_x mesurée à une autre.
- Entraînement : le module est entraîné avec les données se trouvant dans les FIFO dédiées d'entrées et de cibles.
-

Les figures 3.5 et 3.6 sont le résultat d'une augmentation linéaire d'angle d'attaque entre $4,503^\circ$ à $t_1 = 50$ et $6,503^\circ$ à $t_2 = 100$.

Le phénomène d'oubli affecte le module d'identification à région unique :

- avant que l'angle d'attaque ne varie, le module s'est adapté à la condition de vol.
- lorsque l'angle d'attaque commence à augmenter, le module à région unique présente une erreur supérieure à $2,7 \cdot 10^{-4}$ (voir figure 3.6), donc supérieure à l'erreur maximale suite à l'apprentissage hors ligne. Le module a donc perdu de l'information qu'il contenait a priori.
- pour des angles d'attaques supérieurs à 15° (voir figure 3.5), la performance du module s'est nettement dégradée : c'est le phénomène d'oubli.

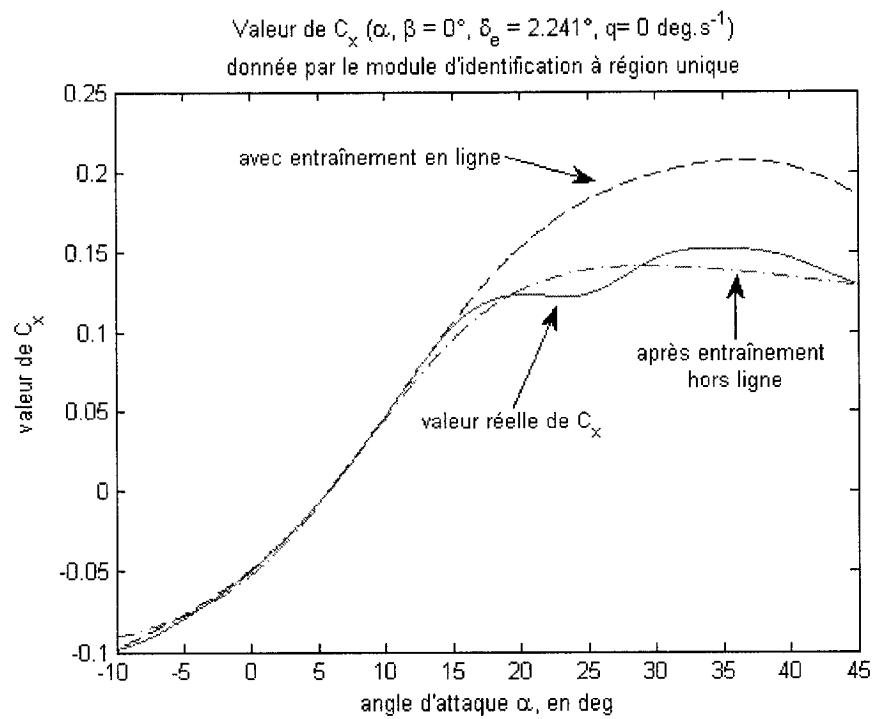
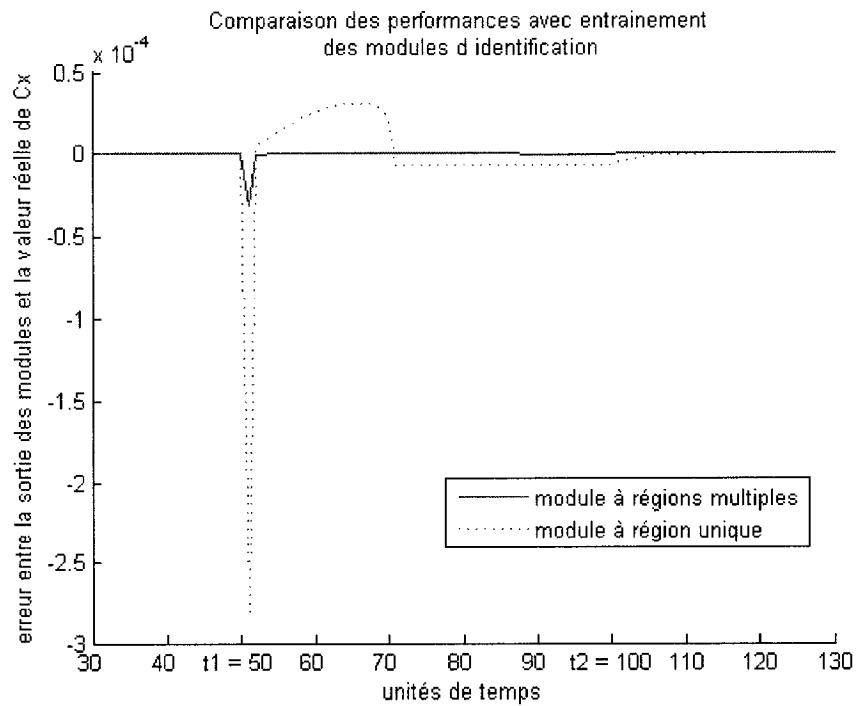


Figure 3.5 Phénomène d'oubli

Figure 3.6 Performance suite à une augmentation d'angle d'attaque α

Le module à régions multiples quant à lui n'est quasiment pas affecté par le changement d'angle d'attaque. La partition a permis de le préserver du phénomène d'oubli, car les régions non traversées par l'angle d'attaque lors du scénario n'ont pas appris en ligne. Ces résultats sont conformes aux attentes et à la raison d'être de la structure en hypercubes.

Les scénarios suivant permettent d'évaluer qualitativement la performance des modules d'identification dans le cas où une panne ou un dégât survient. La re-configuration du contrôleur n'est pas développée dans le cadre de ce projet, mais il est intéressant de concevoir un module d'identification adapté à une telle commande. En effet cela permettrait de simplifier le travail de conception dans le cas d'une évolution de la commande, au sens où le module d'identification pourrait être conservé.

Dans un premier temps modifions légèrement le scénario précédent. Le changement d'angle d'attaque n'est plus progressif entre deux conditions de vol, mais est brutal (t_1 et t_2 égaux), simulant une panne localisée. A $t = t_1 = 30$, l'angle d'attaque passe de $-1,400^\circ$ à $4,530^\circ$. Les résultats de la simulation de ce scénario, dit scénario 1, sont donnés par les figures 3.7 et 3.8.

Dans le cas du module à région unique le phénomène d'oubli est très marqué, avec une erreur de l'ordre de 4.10^{-3} lors du changement d'angle d'attaque (voir figure 3.8). Toutefois le module s'adapte rapidement à la nouvelle configuration. Ceci s'explique par le fait que le changement brutal d'angle d'attaque entraîne une excitation de l'entrée du réseau importante, et dans ce cas l'apprentissage est plus performant (voir section 2.4). L'erreur quadratique moyenne sur l'ensemble de la plage d'angle d'attaque est cependant deux fois plus importante que sans entraînement en ligne (voir figure 3.7).

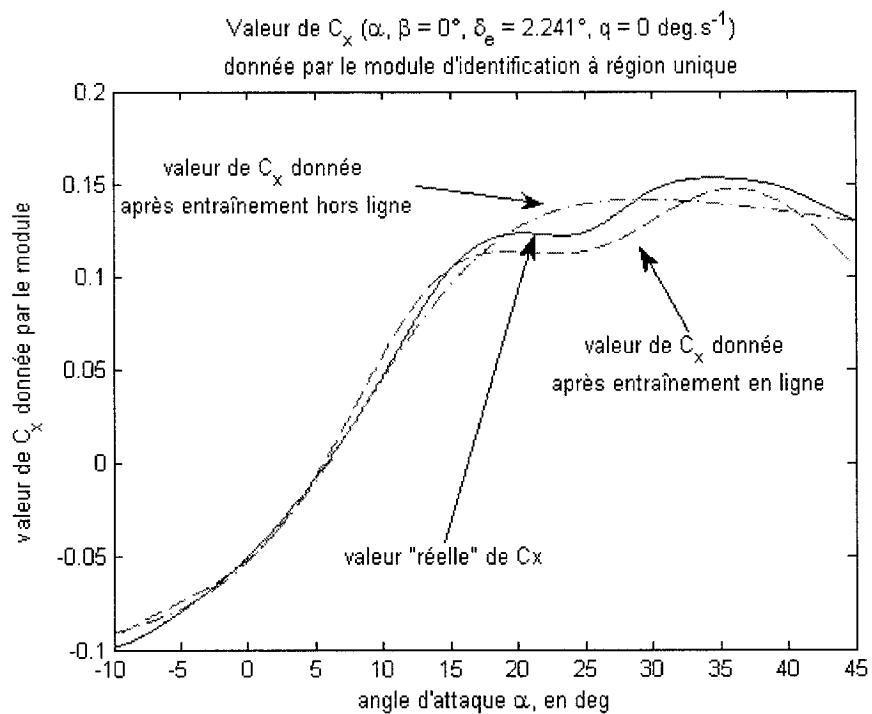


Figure 3.7 Scénario 1 - Module d'identification à région unique

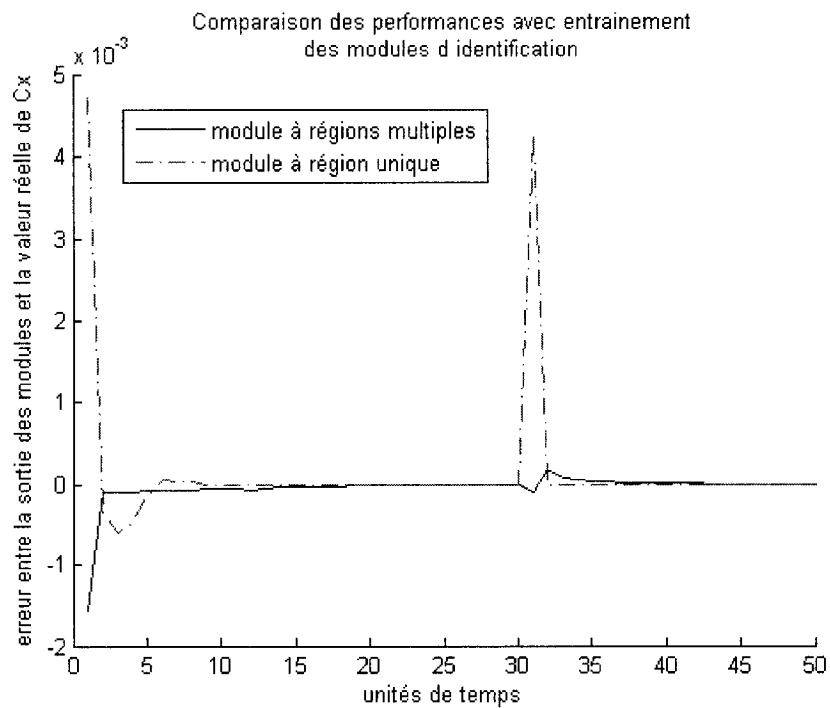


Figure 3.8 Scénario 1 - Performance lors d'un changement brutal d'angle d'attaque

Le module à régions multiples n'est pas autant affecté par le changement brusque de l'angle d'attaque. Il enregistre une erreur plus importante au moment du changement brutal d'angle d'attaque que lors d'un changement progressif, mais ses performances ne sont pas dégradées par rapport à celles sans entraînement.

Le scénario 1 met en évidence que, lors d'un changement brutal des conditions de vol, le module d'identification à régions multiples a une meilleure capacité à estimer la valeur du coefficient aérodynamique sur l'ensemble de la plage d'angle d'attaque.

Dans le cas d'un dégât sur le fuselage ou sur une aile de l'avion, il est vraisemblable que le modèle aérodynamique soit modifié de manière permanente et doive être appris en ligne par le module d'identification. Afin de vérifier le comportement des deux modules suite à une telle panne, considérons non plus un changement brutal d'angle d'attaque, mais celui de la valeur du coefficient C_x (Scénario 2). L'angle d'attaque quant à lui est supposé constant pendant toute la durée du scénario : $\alpha = -1,4^\circ$. Cette configuration est appelée "point de fonctionnement".

En l'absence d'excitation suffisante sur l'ensemble de l'espace d'entrée, aucun des deux modules d'identification n'est à même d'identifier la nouvelle valeur du coefficient aérodynamique (voir figures 3.9 et 3.10) sur l'ensemble de l'enveloppe de vol. Cependant, au point de fonctionnement, soit là où la modification est mesurée, les deux modules identifient correctement ce changement. Le module à régions multiples contient moins de neurones, donc s'adapte plus rapidement aux changements comme le montre la figure 3.11.

Ainsi la structure à régions multiples répond mieux aux exigences de l'identification en ligne que le module d'identification à région unique. En effet, en conditions normales de vol, l'apprentissage en ligne n'entraîne pas de dégradation de performance globale du module d'identification. En cas de conditions dégradées, le module à régions multiples s'adapte plus rapidement aux changements.

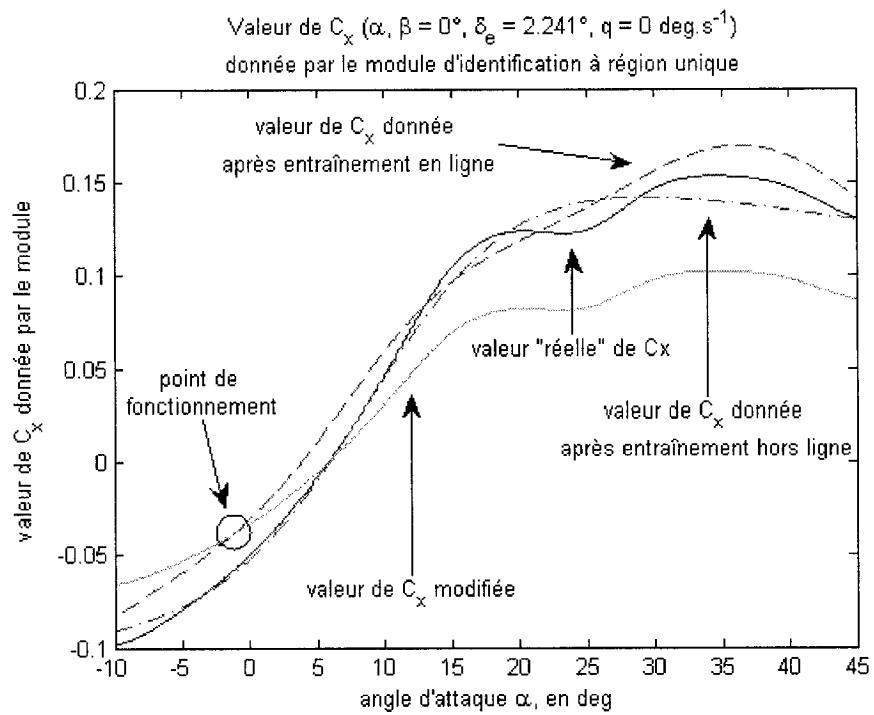


Figure 3.9 Scénario 2 - Module d'identification à région unique

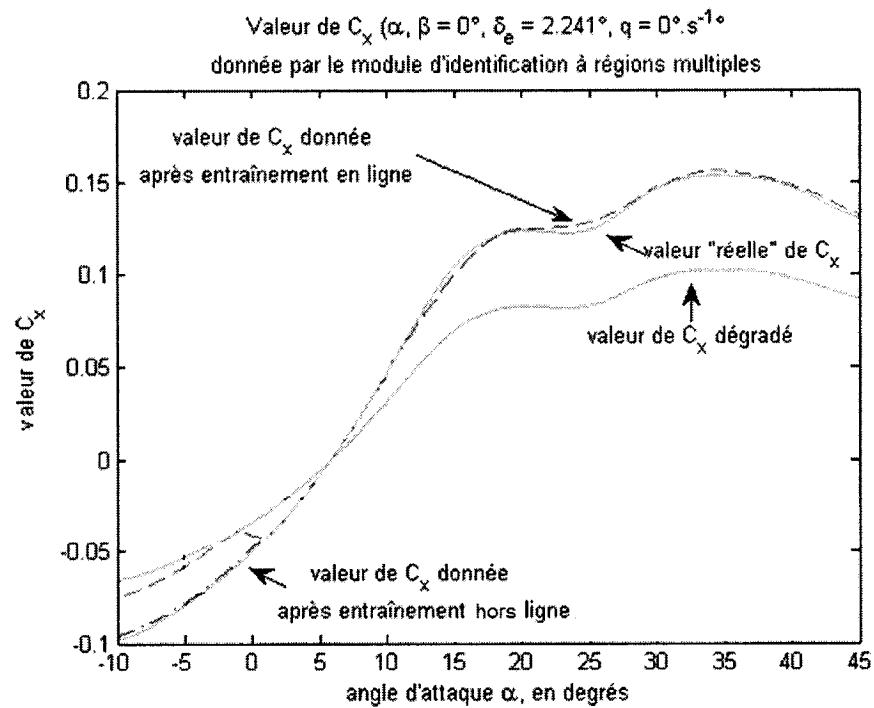


Figure 3.10 Scénario 2 - Module d'identification à régions multiples

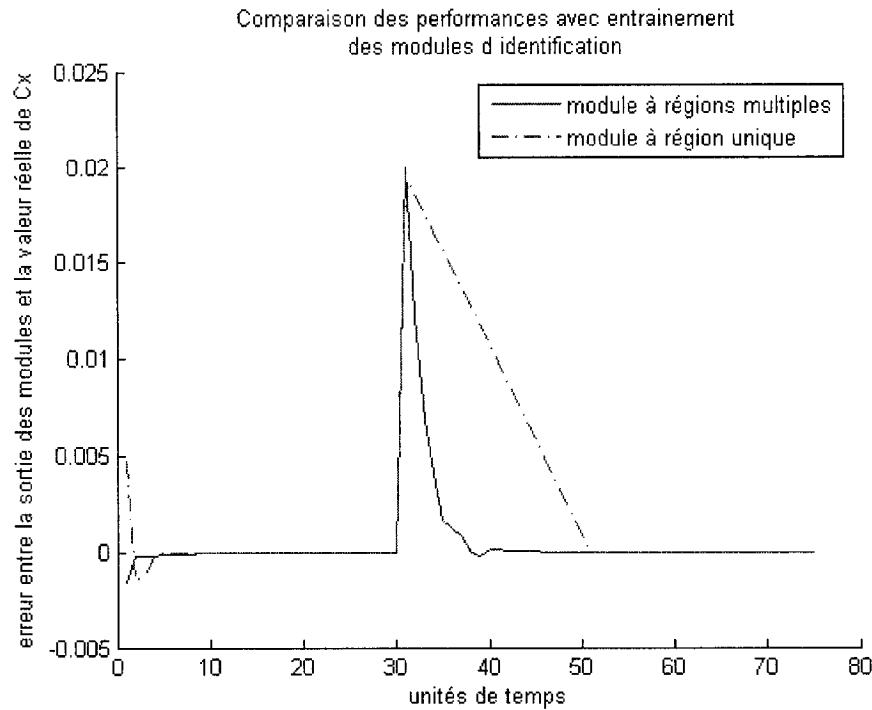


Figure 3.11 Scénario 2 - Performance lors d'un changement brutal de C_x

L'ensemble des simulations précédentes ont été faites en l'absence de bruit de mesure, ou sous l'hypothèse que celui-ci est correctement filtré. Cependant la présence de bruit dans le vecteur d'entrée entraîne généralement une dégradation de performance s'il y a peu d'excitation dans celui-ci et si le réseau est programmé à être entraîné en ligne (voir 2.4).

Tableau 3.6 Caractéristiques des instruments de mesure

Variable	Resolution	Variance du bruit
angle d'attaque α	0,029	0,027
angle de dérapage β	0,018	0,019

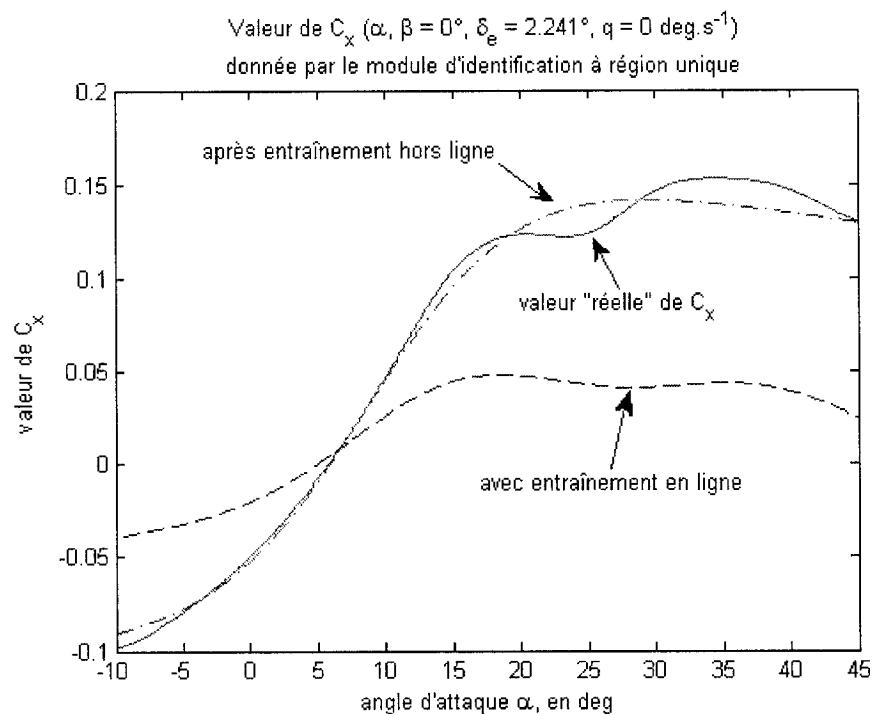


Figure 3.12 Scénario 3 - Module d'identification à région unique

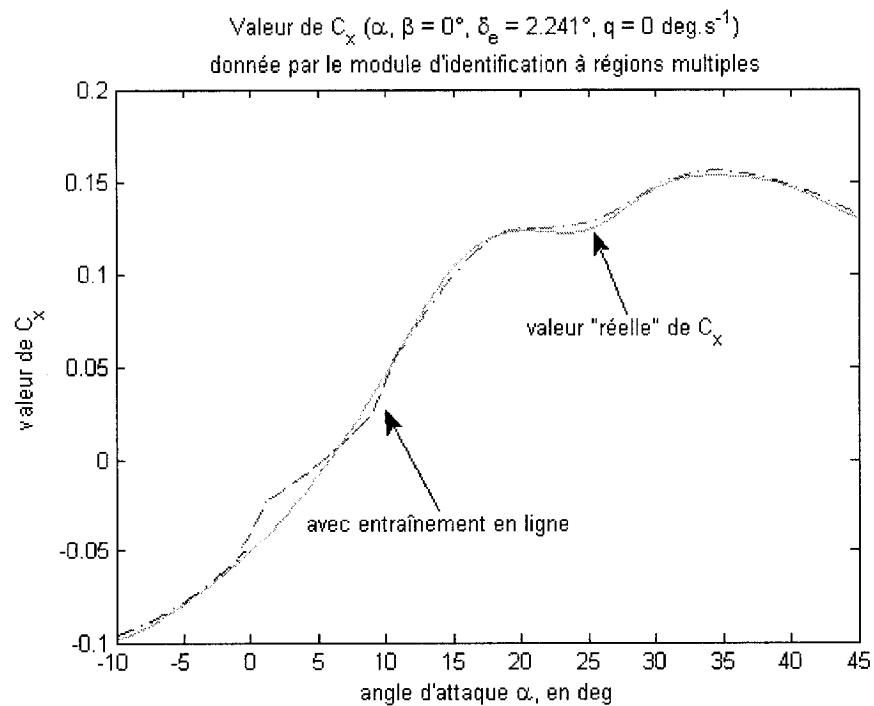


Figure 3.13 Scénario 3 - Module d'identification à régions multiples

Reprendons la simulation d'apprentissage en ligne initiale, soit une augmentation linéaire d'angle d'attaque entre $4,503^\circ$ à $t_1 = 50$ et $6,503^\circ$ à $t_2 = 100$, en y ajoutant du bruit (Scénario 3). Les caractéristiques du bruit introduit sont reportées dans le tableau 3.6, et sont issues des données données par Klein et Morelli (2006).

Comme attendu, la performance globale des modules d'identification s'est dégradée (voir figures 3.12 et 3.13). Le partitionnement de l'espace d'entrée permet d'une part de restreindre l'étendue de la dégradation, et d'autre part de diminuer l'augmentation d'erreur induite par celle-ci pour des configurations autres que celles du point de fonctionnement (voir figure 3.14). En effet, l'erreur est minimale pour $\alpha = 6,503^\circ$, soit l'angle d'attaque de l'avion à la fin du scénario.

Les différents scénarios et leurs résultats permettent de conclure que le module à régions multiples est plus performant que celui à région unique. Toutefois le phénomène d'oubli n'est pas totalement anéanti et reste présent au sein de chaque région.

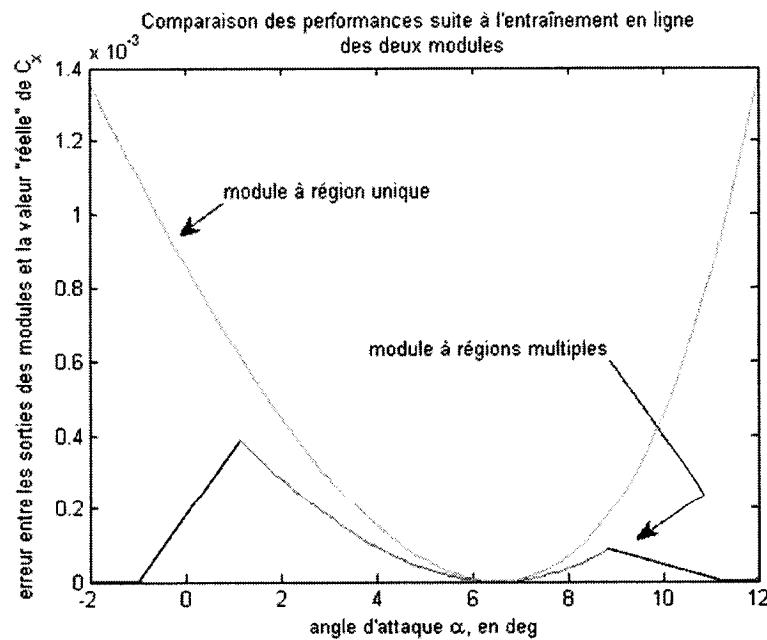


Figure 3.14 Scénario 3 - Comparaison de la performance des deux modules

3.3.3 Conclusion de l'analyse de la méthode des hypercubes

Le partitionnement de l'espace d'entrée présente des avantages notables comme il a été vu précédemment :

- chaque réseau associé à une partition présente un nombre de neurones plus faible qu'un réseau unique utilisé pour identifier les coefficients sur l'ensemble de l'enveloppe de vol.
- le temps d'apprentissage hors ligne est réduit du fait que la complexité de la fonction à identifier est moindre et que le nombre de neurones, donc de poids synaptiques à déterminer, est plus faible.
- le partitionnement restreint l'impact du phénomène d'oubli.
- le module d'identification à régions multiples s'adapte plus rapidement en ligne, que l'avion se trouve en condition de vol normale ou dégradée.

Toutefois certaines limites à la méthode des hypercubes apparaissent :

- le partitionnement effectué et la structure en hypercubes implique la conception de plus d'un million de réseaux pour identifier le modèle aérodynamique complet de l'avion.
- en présence de bruit, chaque hypercube reste soumis au phénomène d'oubli.
- le choix des régions est arbitraire dans la structure de De Weerdt.
- sous des conditions normales de vol, il a été dit en 3.2 qu'un élargissement des dépendances n'est pas souhaitable car cela entraîne la nullité de certains poids synaptiques.

Ainsi l'analyse effectuée sous des conditions restrictives a mis en évidence certains avantages de la méthode des hypercubes, mais aussi ses limites. Cela mène à considérer une architecture novatrice qui minimise les inconvénients de la méthode des hypercubes de De Weerdt (2005).

3.4 Une architecture novatrice

Dans cette section, une nouvelle architecture permettant l'identification du modèle aérodynamique d'un aéronef est proposée. L'architecture générale avec le choix de régions est d'abord présentée, l'architecture interne propre à chaque région est ensuite mise en évidence. Les problématiques de la sélection et du changement de régions sont alors abordées.

La structure en hypercubes de De Weerdt (2005) a été créée spécifiquement pour la reconfiguration. Toutefois, l'utilisation majeure du module d'identification se fera *a priori* en condition de vol dite normale, c'est-à-dire en l'absence de panne et de dégât. Le compromis classique entre fidélité et complexité du modèle est donc à faire, et une architecture moins complexe est envisagée ici.

Un certain nombre des hypothèses faites par De Weerdt sont allégées afin de construire un module d'identification de taille plus petite et donc avec des procédures d'interpolation et de pondération simplifiées. Cela entraîne que l'ensemble des configurations possibles après une panne ou un dégât structurel n'est plus couvert. Dans la mesure où la reconfiguration n'est pas traitée exhaustivement dans ce projet, ces allègements ne sont pas une entrave aux objectifs.

Le partitionnement introduit par De Weerdt implique une explosion du nombre de réseaux au sein du module d'identification. Or l'inspection des coefficients aérodynamiques met en évidence que le facteur de non linéarité le plus impor-

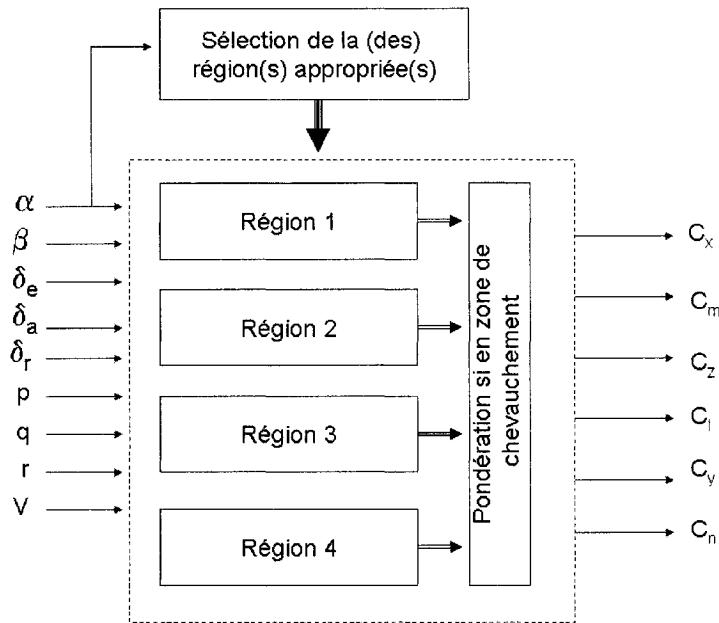


Figure 3.15 Architecture générale

tant est l'angle d'attaque. En effet, le modèle aérodynamique non linéaire montre que la non linéarité est induite par les angles d'attaques et de dérapage. L'influence de ce dernier est cependant moindre, d'après l'observation des données fournies par les essais en soufflerie.

On se propose donc de partitionner l'espace d'entrée uniquement suivant l'angle d'attaque α .

L'architecture générale du module d'identification peut être représentée par la figure 3.15, le nombre de 4 régions d'angle d'attaque y est purement illustratif. La valeur de l'angle d'attaque permet de sélectionner certaines régions. L'ensemble des variables leur est présenté, et elles fournissent une valeur pour chacun des six coefficients aérodynamiques. En fonction du nombre de régions sélectionnées, une pondération peut être faite pour fournir les valeurs finales des coefficients aérodynamiques. Ces différents processus sont expliqués ci-après.

3.4.1 Partitionnement en régions

A priori aucune règle définitive ne peut être érigée pour le choix du partitionnement. Il faut cependant suivre quelques principes : il doit être le plus large possible afin de minimiser le nombre de réseaux du module, mais ne doit pas être trop large sous peine d'être soumis au phénomène d'oubli. La validité des modèles linéaires suggère également que la partition peut être relativement large pour les faibles angles d'attaques ($\alpha < 15^\circ$), mais doit être plus fine au delà car les phénomènes non linéaires sont importants à angle d'attaque élevé et une grande partition entraînerait un nombre de neurones élevé. De plus les partitions doivent présenter une zone de chevauchement, afin d'assurer une transition continue entre les différentes régions comme il va être vu ultérieurement.

L'application de ces principes permet d'obtenir rapidement un partitionnement en régions. Mais seul le développement exhaustif du module d'identification, associé à la connaissance du nombre de neurones obtenus sur la couche cachée des réseaux, permettra de valider cette partition.

Une méthodologie systématique est exposée ici, inspirée de la méthode de dichotomie. Celle-ci n'a pas de fondement mathématique démontré.

La plage des valeurs d'angle d'attaque $[\alpha_{min}, \alpha_{max}]$, pour un aéronef donné, est divisée en deux ensembles. Le premier, $[\alpha_{min}, \alpha_{lin}]$, comprend la plage d'angles d'attaque pour laquelle les modèles aérodynamiques linéaires sont valides ; en général $\alpha_{lin} \approx 15^\circ$. Le second ensemble est constitué de la plage $[\alpha_{lin}, \alpha_{max}]$.

Pour chacun des ensembles, l'algorithme du tableau 3.7 permet alors d'obtenir des régions adaptées aux objectifs. Il est vraisemblable que la plage $[\alpha_{min}, \alpha_{lin}]$ ne nécessite aucun partitionnement supplémentaire, les avions commerciaux auraient ainsi un nombre de régions très limité.

Tableau 3.7 Algorithme pour le choix des régions

1. Fixer les objectifs ϵ : critère de performance m_{max} : nombre maximum de neurones sur la couche cachée des réseaux Δ_α : largeur des zones de chevauchement

2. InitialisationSoit $[\alpha_{min}, \alpha_{max}]$ la plage d'angles d'attaque considérée $n = 1$, rang de l'itérationDévelopper le réseau identifiant le modèle aérodynamique avec le critère de performance ϵ Soit m le nombre de neurones de sa couche cachée

3. Simulation du module d'identification :tant que $m > m_{max}$: $n = n + 1$ Diviser la plage $[\alpha_{min}, \alpha_{max}]$ en n plages égales : $[\alpha_{min}, \alpha_1], \dots, [\alpha_{n-1}, \alpha_n]$

Développer le réseau identifiant le modèle aérodynamique sur la plage

 $[\alpha_{min}, \alpha_1]$ avec le critère de performance ϵ Soit m le nombre de neurones de sa couche cachéefin

4. ItérationReprendre les étapes 2 et 3 avec la plage $[\alpha_1, \alpha_{max}]$ restante

5. Chevauchement des régionsLes partitions $[\alpha_{min}, \alpha_1], [\alpha_1, \alpha_2], \dots, [\alpha_{n-1}, \alpha_{max}]$ sont obtenues

Étendre ces plages de sortes à ce qu'elles se chevauchent :

 $[\alpha_{min}, \alpha_1 + \frac{1}{2}\Delta_\alpha], [\alpha_1 - \frac{1}{2}\Delta_\alpha, \alpha_2 + \frac{1}{2}\Delta_\alpha], \dots, [\alpha_{n-1} - \frac{1}{2}\Delta_\alpha, \alpha_{max}]$

Afin de trouver plus rapidement un partitionnement, l'algorithme peut être appliqué à un seul coefficient aérodynamique plutôt qu'aux six et en un point statique où les valeurs de référence pour les taux de rotation sont nulles (cela sera discuté en 3.4.2). Il convient de choisir un coefficient qui présente un maximum de non linéarité. Cependant les objectifs, en termes de performance d'identification et de nombre maximum de neurones sur la couche cachée des réseaux, doivent être adaptés en conséquence. Une fois le partitionnement trouvé, le module peut être développé avec l'ensemble des six coefficients en sortie.

3.4.2 Architecture d'une région

Chaque région possède la même architecture interne. Elle est illustrée par la figure 3.16, les valeurs numériques qui apparaissent sur cette figure sont celles relatives à l'application de cette architecture au F-16.

L'identification des coefficients aérodynamiques longitudinaux et latéraux se fait de manière découpée. Pour chaque région, les trois coefficients longitudinaux (C_x , C_m et C_z) sont identifiés par un ensemble de réseaux de neurones, tandis que les trois coefficients latéraux (C_l , C_y et C_n) le sont par un autre ensemble. Cette architecture permet de conserver un certain couplage entre chaque groupe de coefficients, et permet d'éviter le risque d'obtenir des poids synaptiques nuls. Cette configuration avait d'ailleurs été préconisée en 3.2.

L'hypothèse d'invariance linéaire est acceptée comme valide que ce soit par rapport à la vitesse V_t ou par rapport aux vitesses angulaires p , q , et r . Rappelons qu'elle est à l'origine des modèles classiquement utilisés, elle semble donc légitime. La structure des points statiques est alors modifiée. La vitesse jouant un rôle artificiel (Klein & Morelli, 2006), une seule valeur de référence est nécessaire, notons la V_0 .

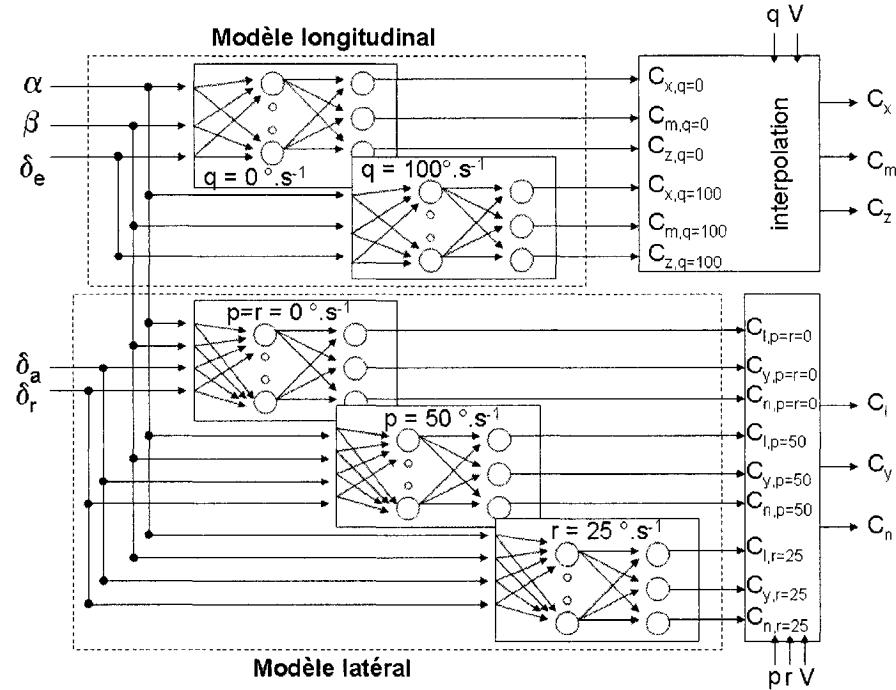


Figure 3.16 Architecture d'une région

Quant aux vitesses angulaires, on peut se limiter à deux valeurs de référence. Afin de simplifier la structure du module d'identification, la première valeur de référence est prise nulle pour chacune des vitesses, la seconde est alors notée respectivement p_0 , q_0 et r_0 . Les nouveaux points statiques sont donc les suivants :

$$\begin{cases}
 \begin{aligned}
 q &= 0^\circ \cdot s^{-1} \\
 q &= q_0^\circ \cdot s^{-1}
 \end{aligned} & \text{pour le modèle longitudinal} \\
 \begin{aligned}
 p = r &= 0^\circ \cdot s^{-1} \\
 p = p_0^\circ \cdot s^{-1}, r &= 0^\circ \cdot s^{-1} \\
 p &= 0^\circ \cdot s^{-1}, r = r_0^\circ \cdot s^{-1}
 \end{aligned} & \text{pour le modèle latéral}
 \end{cases} \quad (3.1)$$

À chaque point statique de chaque région correspond un réseau de neurones. Cha-

cun est obtenu selon la procédure décrite dans le paragraphe 3.2, pour obtenir un réseau dit optimal qui minimise le nombre de neurones sur sa couche cachée.

Les données d'entraînement à utiliser dépendent bien entendu de l'avion considéré et de l'information disponible. Les essais en soufflerie étant une étape incontournable en vue de la certification des aéronefs, les données issues de ces essais devraient être disponibles pour l'analyse post-expérience et sont d'ailleurs particulièrement bien adaptées à l'utilisation qui en est faite ici. En effet, les différentes contributions aux coefficients aérodynamiques sont relevées pendant les essais et sont ensuite fournies sous forme de tables. Il est alors nécessaire de calculer la valeur totale des coefficients aérodynamiques à la condition de vol désirée selon les équations générales suivantes :

$$\begin{aligned}
 C_{a_{total}} &= C_{a_0}(\alpha, \beta, \delta_e) + \frac{q\bar{c}}{2V_t} C_{a_q}(\alpha, \beta, \delta_e) && \text{pour les longitudinaux} \\
 C_{a_{total}} &= C_{a_0}(\alpha, \beta, \delta_a, \delta_r) + \frac{pb}{2V_t} C_{a_p}(\alpha, \beta, \delta_a, \delta_r) \\
 &+ \frac{rb}{2V_t} C_{a_r}(\alpha, \beta, \delta_a, \delta_r) && \text{pour les latéraux}
 \end{aligned} \tag{3.2}$$

Ces équations génériques sont à remplacer par les équations du modèle aérodynamique non linéaire utilisé pour les essais en soufflerie et spécifique à chaque aéronef.

Les données d'entraînement pour chaque point statique de chaque région sont ensuite obtenues en substituant dans les équations précédentes les valeurs de q ou de p et r associées au point statique et en utilisant les valeurs des contributions dont la variable en angle d'attaque appartient à la région considérée.

Ainsi pour chaque région d'angle d'attaque le module d'identification comprend 2 réseaux pour le modèle longitudinal, qui fournissent respectivement $\{C_{x_{q=0}}, C_{m_{q=0}}, C_{z_{q=0}}\}$ et $\{C_{x_{q=q_0}}, C_{m_{q=q_0}}, C_{z_{q=q_0}}\}$, et 3 réseaux pour le modèle latéral, qui

fournissent respectivement $\{ C_{l_{p=0,r=0}}, C_{y_{p=0,r=0}}, C_{n_{p=0,r=0}} \}$, $\{ C_{l_{p=p_0,r=0}}, C_{y_{p=p_0,r=0}}, C_{n_{p=p_0,r=0}} \}$ et $\{ C_{l_{p=0,r=r_0}}, C_{y_{p=0,r=r_0}}, C_{n_{p=0,r=r_0}} \}$.

Les sorties des réseaux sont alors traitées afin de calculer la valeur des coefficients aérodynamiques au point de fonctionnement (V_t, p, q, r) de l'avion.

Les coefficients longitudinaux C_x , C_m et C_z sont obtenus par l'équation:

$$C_a = C_a(\alpha, \beta, \delta_e)_{q=0} + \frac{q\bar{C}}{2V_t} \Delta C_{a_q}(\alpha, \beta, \delta_e) \quad (3.3)$$

où $\Delta C_{a_q} = \frac{2V_0}{q_0\bar{C}} (C_a(\alpha, \beta, \delta_e)_{q=q_0} - C_a(\alpha, \beta, \delta_e)_{q=0})$

et les coefficients latéraux C_l , C_y et C_n par :

$$C_a = C_a(\alpha, \beta, \delta_a, \delta_r)_{p=0,r=0} + \frac{pb}{2V_t} \Delta C_{a_p}(\alpha, \beta, \delta_a, \delta_r) + \frac{rb}{2V_t} \Delta C_{a_r}(\alpha, \beta, \delta_a, \delta_r) \quad (3.4)$$

où $\Delta C_{a_p} = \frac{2V_0}{p_0b} (C_a(\alpha, \beta, \delta_a, \delta_r)_{p=0,r=0} - C_a(\alpha, \beta, \delta_a, \delta_r)_{p=p_0,r=0})$

et $\Delta C_{a_r} = \frac{2V_0}{r_0b} (C_a(\alpha, \beta, \delta_a, \delta_r)_{p=0,r=0} - C_a(\alpha, \beta, \delta_a, \delta_r)_{p=0,r=r_0})$

3.4.3 Sélection des régions

Les régions dont la plage d'angles d'attaque associée ne comprend pas l'angle d'attaque actuel de l'avion sont désactivées, au sens où les entrées ne sont pas fournies aux réseaux de ces régions. Ces réseaux ne fournissent donc aucune sortie dans ce cas. Afin de ne pas avoir une sortie vide pour ces régions, la dernière valeur active est maintenue.

À l'inverse les régions dont la plage comprend l'angle d'attaque actuel de l'avion sont activées : les entrées leur sont présentées et elles fournissent en sortie une

valeur pour chacun des six coefficients aérodynamiques.

Si une seule région est activée, les valeurs des coefficients aérodynamiques sont directement celles fournies par cette région.

Si deux régions sont activées la procédure suivante assure un changement de région conforme aux spécifications de l'identification.

3.4.4 Changement de région

Le changement de région doit être continu, comme il a été énoncé en 3.3.1. Avec l'architecture choisie, le chevauchement est unidimensionnel, et non plus multidimensionnel comme cela pouvait être le cas avec la structure en hypercubes de De Weerdt. En effet, avec l'architecture novatrice, seule change la plage de valeurs de l'angle d'attaque α entre deux régions adjacentes.

Plutôt que d'interpoler linéairement entre les deux valeurs de chaque coefficient aérodynamique aux bornes de la zone de chevauchement, les valeurs données par les deux régions sont pondérées de la manière suivante :

$$C_a = \frac{\alpha - \alpha_1}{\alpha_2 - \alpha_1} C_{a_2} + \frac{\alpha_2 - \alpha}{\alpha_2 - \alpha_1} C_{a_1} \quad (3.5)$$

où C_{a_1} (respectivement C_{a_2}) est la valeur du coefficient donnée par la première région (respectivement la seconde) et α_1 (respectivement α_2) est la valeur d'angle d'attaque à la borne de la zone de chevauchement se trouvant dans la première région (respectivement la seconde).

Cette pondération permet de maintenir une erreur quadratique satisfaisant les performances exigées du module d'identification. En effet, chaque région fournit une

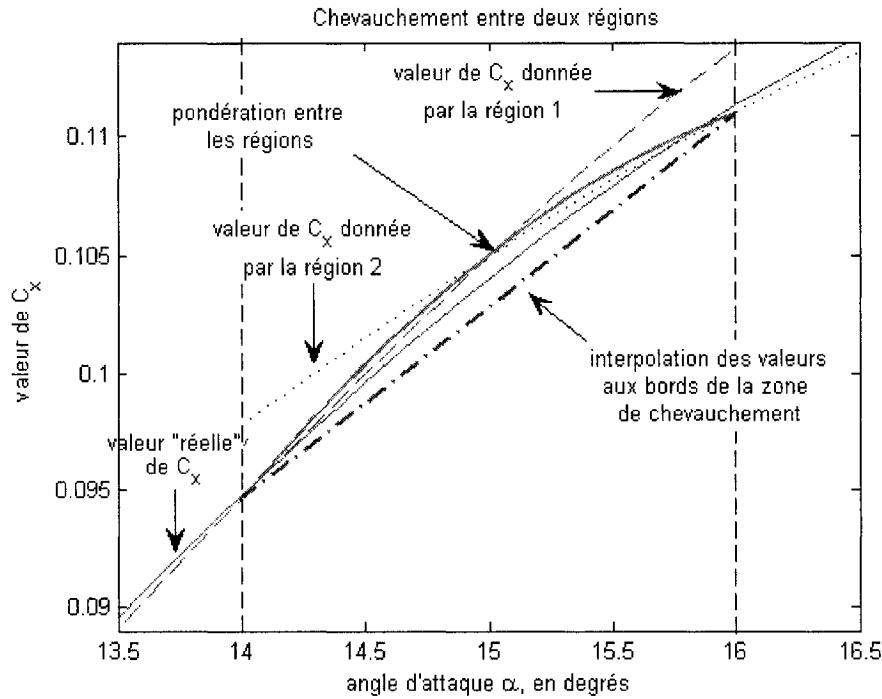


Figure 3.17 Changement de région continu

valeur dont l'erreur quadratique avec la valeur réelle est inférieure à ϵ , et donc

$$(C_a - C_{a_{real}})^2 = \left(\frac{\alpha - \alpha_1}{\alpha_2 - \alpha_1} (C_{a_2} - C_{a_{real}}) + \frac{\alpha_2 - \alpha}{\alpha_2 - \alpha_1} (C_{a_1} - C_{a_{real}}) \right)^2 < \epsilon \quad (3.6)$$

La figure 3.17 illustre la différence entre la pondération des valeurs de chaque région et l'interpolation des valeurs aux bornes de la zone de chevauchement.

3.4.5 Une approche pour l'apprentissage en ligne

Une approche pour l'apprentissage en ligne est proposé par De Weerdt pour son module d'identification. Celle-ci est également applicable à la structure développée ici. L'apprentissage en ligne ne faisant pas partie intégrante de ce projet, seules les grandes lignes (DeWeerdt et al., 2005) sont exposées ici.

Dans un premier temps, les données nécessaires à l'apprentissage en ligne sont calculées à partir des mesures des capteurs selon les équations 3.7 et 3.8. Les moments d'inertie sont supposés connus, et une modélisation du moteur permet d'estimer sa poussée.

$$\begin{aligned} C_x &= \frac{1}{\bar{q}S} (ma_x - T) \\ C_y &= \frac{ma_y}{\bar{q}S} \\ C_z &= \frac{ma_z}{\bar{q}S} \end{aligned} \quad (3.7)$$

$$\begin{aligned} C_l &= \frac{1}{\bar{q}Sb} (I_x \dot{p} - I_{xz} (pq + \dot{r}) + (I_z - I_y) qr) \\ C_m &= \frac{1}{\bar{q}Sc} (I_y \dot{q} + (I_x - I_z) pr + I_{xz} (p^2 - r^2)) \\ C_n &= \frac{1}{\bar{q}Sb} (I_z \dot{r} - I_{xz} (\dot{p} - qr) + (I_y - I_x) pq) \end{aligned} \quad (3.8)$$

Une méthode d'interpolation permet alors l'entraînement en ligne à partir des données stockées dans une FIFO de taille donnée. La taille de cette FIFO est étroitement liée à la dimension de l'espace d'entrée du module d'identification.

Les réseaux de neurones peuvent être implémentés en software ou en hardware. Le software permet une plus grande flexibilité, et le hardware une vitesse accrue. Une possibilité d'implémentation hardware est proposée ci-après.

Un exemple de processeur neuronal est le pRAM-256 VLSI (Ng & Clarkson ,1998). Il possède 6 entrées et 256 neurones. Leur connection étant configurable, il est possible de créer l'architecture désirée. De plus, le nombre d'entrées est compatible avec l'architecture développée ici.

Le nombre de neurones est a priori suffisant pour chaque point statique si le nombre de neurones maximal sur la couche cachée est inférieure à m_{max} donné par l'équation 3.9, où n est le nombre de régions d'angle d'attaque du module d'identification.

$$m_{max} = \frac{256}{n} - 3 \quad (3.9)$$

Lorsque l'ensemble des 256 neurones est utilisé, le délai de traitement des entrées est de $0,154\text{ ms}$ (à 33 MHz). Un cycle d'apprentissage pour ce même ensemble dure quant à lui $0,246\text{ ms}$ (à 33 MHz). Sachant que les accéléromètres fournissent une donnée toutes les $0,5$ à $1,2\text{ ms}$, et les gyroscope toutes les $7,5\text{ ms}$ (Klein & Morelli, 2006), les vitesses de traitement et d'entraînement sont satisfaisantes.

3.4.6 Conclusion sur l'architecture développée

Le module d'identification ainsi créé minimise les inconvénients de la structure en hypercubes de De Weerdt. En effet, il contient $5 \times n$ réseaux de neurones, où n est le nombre de régions d'angle d'attaque. Ce nombre, même avec de nombreuses régions d'angle d'attaque, semble plus raisonnable que le million de réseaux proposé initialement.

Suite à l'entraînement hors ligne, le module d'identification développé a des caractéristiques comparables à celles de la structure en hypercubes de De Weerdt, tout en étant moins complexe que celle-ci :

- le nombre de réseaux du module est réduit par rapport à la structure en hypercubes
- chacun des réseaux peut être construit de sorte à satisfaire le critère de performance requis
- l'obtention de la valeur des coefficients aérodynamiques est simplifiée
- le processus de changement de région est également simplifié
- la valeur des coefficients est continue lors d'un changement de région

3.5 Application au F-16

L'architecture développée précédemment est appliquée à l'identification du modèle aérodynamique du F-16. Quelques ajustements sont toutefois nécessaires. L'espace d'entrée, à la fois du modèle longitudinal et du modèle latéral, doit être élargie pour prendre en compte l'angle du bord d'attaque. L'élévateur doit figurer parmi les entrées "globales" du module, car son angle de déflexion influence l'ensemble des coefficients aérodynamiques, au même titre que les angles d'attaque α et de dérapage β .

La première étape du développement est le choix des régions, conformément à la procedure du paragraphe 3.4.1.

L'inspection des valeurs des coefficients aérodynamiques issues des essais en soufflerie et ainsi que la plage de validité des modèles dit "basse fidélité" du F-16, a mené à choisir $\alpha_{lin} = 15^\circ$. L'algorithme de la section 3.4.1 est appliqué au coefficient C_{x_0} pour $q = 0^\circ \cdot s^{-1}$, avec un critère de performance de $1,4 \cdot 10^{-6}$, égal au dixième du carré de 1% de la variation de ce coefficient, et un nombre maximal de neurones sur la couche cachée de 30.

Il donne que la plage $[-10^\circ, 15^\circ]$ n'a pas à être plus partitionnée. Il donne ensuite qu'il ne suffit pas de partitionner la plage $[15^\circ, 45^\circ]$ en deux, mais pour $n = 3$ les résultats sont satisfaisants. On obtient alors la plage $[15^\circ, 25^\circ]$. La figure 3.18 illustre les résultats obtenus. Les performances sont finalement atteintes en partitionnant la plage restante en deux, soit en $[25^\circ, 35^\circ]$ et $[35^\circ, 45^\circ]$. Les partitions sont ensuite ajustées de sorte à obtenir une zone de chevauchement de 2° entre chacune d'elles; ce choix est motivé par le choix de De Weerdt.

Pour des fins de clarté les régions en angle d'attaque sont numérotées. La région 1 correspond à la plage $[-10^\circ, 16^\circ]$, la région 2 à la plage $[14^\circ, 26^\circ]$, la région 3 à la plage $[24^\circ, 36^\circ]$ et la région 4 à la plage $[34^\circ, 45^\circ]$.

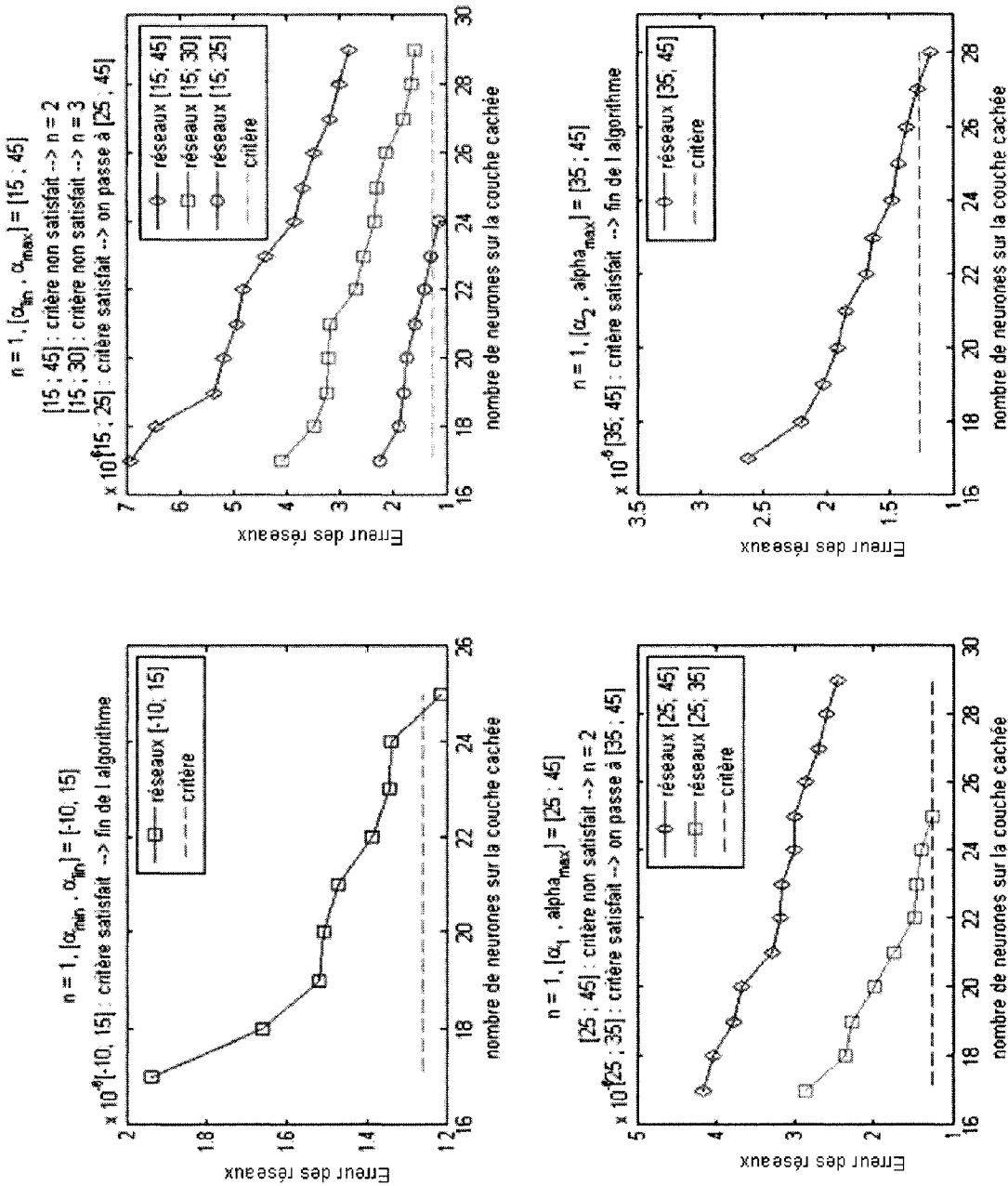


Figure 3.18 Illustration de l'algorithme de choix des régions appliqué au F-16

Tableau 3.8 Espace d'entrée du module d'identification du F-16

Variable	plage	partition ou valeur de référence	unité
Vitesse V_t	[30 ; 300]	150	$m.s^{-1}$
Taux de roulis p	[-595 ; 595]	0 ; 50	$^{\circ}.s^{-1}$
Taux de tangage q	[-205 ; 205]	0 ; 100	$^{\circ}.s^{-1}$
Taux de lacet r	[-135 ; 135]	0 ; 25	$^{\circ}.s^{-1}$
Angle d'attaque α	[-10 ; 45]	[-10 ; 16] [14 ; 26] [24 ; 36] [34 ; 45]	degré
Angle de dérapage β	[-30 ; 30]	[-30 ; 30]	degré
Élévateur δ_e	[-25 ; 25]	[-25 ; 25]	degré
Ailerons δ_a	[-21 ; 21]	[-21 ; 21]	degré
Gouvernail δ_r	[-30 ; 30]	[-30 ; 30]	degré
Bord d'attaque δ_{lef}	[0 ; 25]	[0 ; 25]	degré

Les points statiques doivent être fixés. Pour cela les valeurs de référence de vitesse et des taux de roulis, de tangage et de lacet sont choisies arbitrairement dans l'enveloppe de vol du F-16 établie au chapitre 1, de sorte à ce que $\frac{\bar{e}q_0}{2V_0}$, $\frac{p_{0b}}{2V_0}$ et $\frac{r_{0b}}{2V_0}$ soient proches de l'unité : $V_0 = 150m.s^{-1}$, $p_0 = 50^{\circ}.s^{-1}$, $q_0 = 100^{\circ}.s^{-1}$ et $r_0 = 25^{\circ}.s^{-1}$. Le tableau 3.8 récapitule les variables de l'espace d'entrée.

Une analyse par région des valeurs des coefficients aérodynamiques du F-16 du rapport de la NASA (Nguyen et al, 1979) est nécessaire, car ces données servent à l'entraînement des réseaux et doivent être prétraitées pour optimiser leur apprentissage. Les moyennes ($E(C_a)$) et les écarts types ($\sigma(C_a)$) de chaque coefficient longitudinal (voir tableau 3.9) et latéral (voir tableau 3.10) permettent de normaliser les données d'entraînement selon l'équation suivante :

$$C_{a_{norm}} = \frac{C_a - E(C_a)}{\sigma(C_a)} \quad (3.10)$$

Tableau 3.9 Compilation par région des coefficients longitudinaux

Région		1			
valeur de q		0		100	
		E	σ	E	σ
valeur de C_x	-0,0322	0,0424	-0,0055	0,0533	
valeur de C_m	-0,0220	0,1427	-0,1832	0,1495	
valeur de C_z	-0,2075	0,4961	-0,7796	0,5704	
Région		2			
valeur de q		0		100	
		E	σ	E	σ
valeur de C_x	0,0465	0,0411	0,0804	0,0498	
valeur de C_m	-0,0727	0,1375	-0,2657	0,1366	
valeur de C_z	-1,2496	0,2323	-1,8522	0,2158	
Région		3			
valeur de q		0		100	
		E	σ	E	σ
valeur de C_x	0,0536	0,0494	0,0760	0,0535	
valeur de C_m	-0,1236	0,1271	-0,3277	0,1251	
valeur de C_z	-1,7087	0,2260	-2,3128	0,2369	
Région		4			
valeur de q		0		100	
		E	σ	E	σ
valeur de C_x	0,0615	0,0525	0,0838	0,0557	
valeur de C_m	-0,1612	0,1001	-0,3746	0,0983	
valeur de C_z	-1,9724	0,2054	-2,6987	0,2162	

Tableau 3.10 Compilation par région des coefficients latéraux

Région		1					
valeur de $\{p, r\}$		$\{0, 0\}$		$\{50, 0\}$		$\{0, 25\}$	
		E	σ	E	σ	E	σ
valeur de C_l		$1, 3.10^{-4}$	0, 040	$-1, 0.10^{-2}$	0, 040	$8, 8.10^{-4}$	0, 040
valeur de C_y		$-5, 3.10^{-3}$	0, 362	$-4, 1.10^{-3}$	0, 362	$7, 7.10^{-4}$	0, 362
valeur de C_n		$8, 0.10^{-4}$	0, 077	$9, 5.10^{-4}$	0, 077	$-5, 1.10^{-4}$	0, 077

Région		2					
valeur de $\{p, r\}$		$\{0, 0\}$		$\{50, 0\}$		$\{0, 25\}$	
		E	σ	E	σ	E	σ
valeur de C_l		$3, 2.10^{-4}$	0, 050	$-8, 2.10^{-3}$	0, 050	$5, 4.10^{-3}$	0, 050
valeur de C_y		$-2, 8.10^{-3}$	0, 306	$6, 1.10^{-3}$	0, 306	$9, 2.10^{-3}$	0, 306
valeur de C_n		$7, 8.10^{-4}$	0, 051	$2, 0.10^{-3}$	0, 051	$-7, 5.10^{-3}$	0, 051

Région		3					
valeur de $\{p, r\}$		$\{0, 0\}$		$\{50, 0\}$		$\{0, 25\}$	
		E	σ	E	σ	E	σ
valeur de C_l		$-1, 5.10^{-3}$	0, 044	$-7, 4.10^{-3}$	0, 044	$3, 4.10^{-3}$	0, 044
valeur de C_y		$-1, 2.10^{-2}$	0, 230	$-2, 7.10^{-4}$	0, 230	$-8, 0.10^{-4}$	0, 230
valeur de C_n		$1, 8.10^{-3}$	0, 031	$6, 3.10^{-3}$	0, 032	$-8, 2.10^{-3}$	0, 031

Région		4					
valeur de $\{p, r\}$		$\{0, 0\}$		$\{50, 0\}$		$\{0, 25\}$	
		E	σ	E	σ	E	σ
valeur de C_l		$-1, 9.10^{-3}$	0, 037	$-2, 8.10^{-3}$	0, 037	$-4, 3.10^{-3}$	0, 038
valeur de C_y		$-4, 4.10^{-3}$	0, 182	$-1, 8.10^{-2}$	0, 186	$-7, 9.10^{-3}$	0, 182
valeur de C_n		$3, 2.10^{-4}$	0, 040	$1, 0.10^{-2}$	0, 040	$-1, 1.10^{-2}$	0, 040

Tableau 3.11 Valeurs extrêmes des coefficients

Coefficient	Min.	Max.	Δ	$(\Delta.0,01)^2$
$q = 0^\circ \cdot s^{-1}$	C_x -0,1712	0,1838	0,3550	$1,26.10^{-5}$
	C_m -0,4319	0,3396	0,7715	$5,95.10^{-5}$
	C_z -2,4471	0,9573	3,4044	$1,16.10^{-3}$
$q = 100^\circ \cdot s^{-1}$	C_x -0,1527	0,2175	0,3702	$1,37.10^{-5}$
	C_m -0,6183	0,1915	0,8038	$6,46.10^{-5}$
	C_z -3,2204	0,7574	3,9778	$1,58.10^{-3}$
$p = 0^\circ \cdot s^{-1},$ $r = 0^\circ \cdot s^{-1}$	C_l -0,1155	0,1801	0,2956	$8,74.10^{-6}$
	C_y -0,8096	0,8775	1,6871	$2,84.10^{-4}$
	C_n -0,1966	0,1816	0,3782	$1,43.10^{-5}$
$p = 50^\circ \cdot s^{-1},$ $r = 0^\circ \cdot s^{-1}$	C_l -0,1272	0,1639	0,2911	$8,47.10^{-6}$
	C_y -0,7974	0,8928	1,6902	$2,86.10^{-4}$
	C_n -0,1955	0,1837	0,3792	$1,44.10^{-5}$
$p = 0^\circ \cdot s^{-1},$ $r = 25^\circ \cdot s^{-1}$	C_l -0,1091	0,1736	0,2927	$8,57.10^{-6}$
	C_y -0,7971	0,9593	1,7564	$3,08.10^{-4}$
	C_n -0,2024	0,1759	0,3783	$1,43.10^{-5}$

Les valeurs extrêmes des coefficients (voir tableau 3.11) permettent de déterminer le critère d'arrêt pour l'apprentissage des réseaux. On souhaite identifier les coefficients aérodynamiques à 1% près en moyenne. Pour cela le critère de performance est pris égal à la moyenne des 1% de la variation des trois coefficients aérodynamiques identifiés. Soit pour le modèle longitudinal :

$$\frac{(3,55.10^{-3})^2 + (7,715.10^{-3})^2 + (3,4044.10^{-2})^2}{3} = 4,10.10^{-4}$$

et pour le modèle latéral :

$$\frac{(2,956.10^{-3})^2 + (1,687.10^{-2})^2 + (3,782.10^{-3})^2}{3} = 1,03.10^{-4}$$

Tableau 3.12 Valeurs expérimentales - Réseaux du modèle longitudinal

Région d'angle d'attaque	1	2	3	4
$q = 0^\circ \cdot s^{-1}$				
nb neurones	34	25	41	47
performance C_x	$1,86 \cdot 10^{-6}$	$3,79 \cdot 10^{-6}$	$2,55 \cdot 10^{-6}$	$3,78 \cdot 10^{-6}$
performance C_m	$2,66 \cdot 10^{-5}$	$4,20 \cdot 10^{-5}$	$2,72 \cdot 10^{-5}$	$5,81 \cdot 10^{-5}$
performance C_z	$1,16 \cdot 10^{-4}$	$9,84 \cdot 10^{-5}$	$1,14 \cdot 10^{-4}$	$5,74 \cdot 10^{-4}$
performance globale	$4,83 \cdot 10^{-5}$	$4,81 \cdot 10^{-5}$	$4,79 \cdot 10^{-5}$	$2,12 \cdot 10^{-4}$
$q = 100^\circ \cdot s^{-1}$				
nb neurones	36	26	43	52
performance C_x	$1,72 \cdot 10^{-6}$	$4,24 \cdot 10^{-6}$	$2,18 \cdot 10^{-6}$	$3,26 \cdot 10^{-6}$
performance C_m	$2,65 \cdot 10^{-5}$	$3,90 \cdot 10^{-5}$	$2,38 \cdot 10^{-5}$	$4,64 \cdot 10^{-5}$
performance C_z	$1,16 \cdot 10^{-4}$	$1,01 \cdot 10^{-4}$	$1,21 \cdot 10^{-4}$	$5,28 \cdot 10^{-4}$
performance globale	$4,80 \cdot 10^{-5}$	$4,82 \cdot 10^{-5}$	$4,89 \cdot 10^{-5}$	$1,93 \cdot 10^{-4}$

La moyenne de ces deux valeurs est bien égale au critère usuel, soit une erreur moyenne quadratique d'entraînement inférieure à $2,5 \cdot 10^{-4}$.

Les réseaux sont alors développés selon la procédure mise en évidence en début de chapitre et implémentée sous MatLab (voir annexe III). Leurs caractéristiques, à savoir leur nombre de neurones sur la couche cachée et leur performance de généralisation, sont résumées dans les tableaux 3.12 et 3.13.

Le nombre de neurones de chaque réseau est certes plus élevé que celui des réseaux de la structure en hypercubes de De Weerdt, mais reste raisonnable. En effet, De Weerdt obtient entre 4 et 15 neurones par réseau et donc par région (au nombre de 1400000). L'architecture élaborée ici en compte entre 25 et 61, mais elle ne contient au total que 8 réseaux pour le modèle longitudinal, et 12 pour le modèle latéral.

Tableau 3.13 Valeurs expérimentales - Réseaux du modèle latéral

Région d'angle d'attaque	1	2	3	4
$p = 0^\circ \cdot s^{-1}, r = 0^\circ \cdot s^{-1}$				
nb neurones	40	32	45	59
performance C_l	$1,54 \cdot 10^{-6}$	$3,45 \cdot 10^{-6}$	$4,38 \cdot 10^{-6}$	$4,02 \cdot 10^{-6}$
performance C_y	$5,07 \cdot 10^{-5}$	$5,02 \cdot 10^{-5}$	$3,49 \cdot 10^{-5}$	$6,24 \cdot 10^{-5}$
performance C_n	$4,88 \cdot 10^{-6}$	$4,23 \cdot 10^{-6}$	$3,08 \cdot 10^{-6}$	$6,61 \cdot 10^{-6}$
performance globale	$1,90 \cdot 10^{-5}$	$1,93 \cdot 10^{-5}$	$1,41 \cdot 10^{-5}$	$2,44 \cdot 10^{-5}$
$p = 50^\circ \cdot s^{-1}, r = 0^\circ \cdot s^{-1}$				
nb neurones	40	34	47	61
performance C_l	$1,49 \cdot 10^{-6}$	$3,37 \cdot 10^{-6}$	$4,18 \cdot 10^{-6}$	$3,83 \cdot 10^{-6}$
performance C_y	$4,38 \cdot 10^{-5}$	$3,60 \cdot 10^{-5}$	$3,79 \cdot 10^{-5}$	$5,43 \cdot 10^{-5}$
performance C_n	$4,83 \cdot 10^{-6}$	$3,84 \cdot 10^{-6}$	$3,25 \cdot 10^{-6}$	$6,59 \cdot 10^{-6}$
performance globale	$1,67 \cdot 10^{-5}$	$1,44 \cdot 10^{-5}$	$1,51 \cdot 10^{-5}$	$2,16 \cdot 10^{-5}$
$p = 0^\circ \cdot s^{-1}, r = 25^\circ \cdot s^{-1}$				
nb neurones	41	32	46	60
performance C_l	$1,47 \cdot 10^{-6}$	$3,43 \cdot 10^{-6}$	$4,05 \cdot 10^{-6}$	$4,47 \cdot 10^{-6}$
performance C_y	$3,89 \cdot 10^{-5}$	$4,17 \cdot 10^{-5}$	$4,12 \cdot 10^{-5}$	$5,41 \cdot 10^{-5}$
performance C_n	$4,11 \cdot 10^{-6}$	$4,21 \cdot 10^{-6}$	$3,20 \cdot 10^{-4}$	$6,18 \cdot 10^{-6}$
performance globale	$1,48 \cdot 10^{-5}$	$1,64 \cdot 10^{-5}$	$1,62 \cdot 10^{-5}$	$2,16 \cdot 10^{-5}$

La taille des réseaux obtenus est d'ailleurs compatible avec la proposition d'utiliser le p256Ram. En effet les réseaux pour $q = 0^\circ \cdot s^{-1}$ totalisent 159 neurones, et ceux pour $q = 100^\circ \cdot s^{-1}$ 169. Chaque groupe peut donc être implémenté sur une carte.

L'erreur quadratique moyenne d'entraînement est inférieure à $2,5 \cdot 10^{-4}$, valeur de référence, pour l'ensemble des réseaux. En outre, pour chacun des coefficients l'erreur quadratique moyenne est inférieure au carré d'un pourcent de l'écart entre les valeurs minimale et maximale du coefficient, grandeur donnée dans la dernière colonne du tableau 3.11. Cela confirme que les réseaux obtenus sont satisfaisants.

L'erreur quadratique moyenne d'entraînement a pu être abaissée à 5.10^{-5} pour les réseaux du modèle longitudinal (à l'exception de la région 4) et à 2.10^{-5} pour ceux du modèle latéral, sans pour autant entraîner une explosion du nombre de neurones sur la couche cachée des réseaux. Soit une amélioration de facteur 10 !

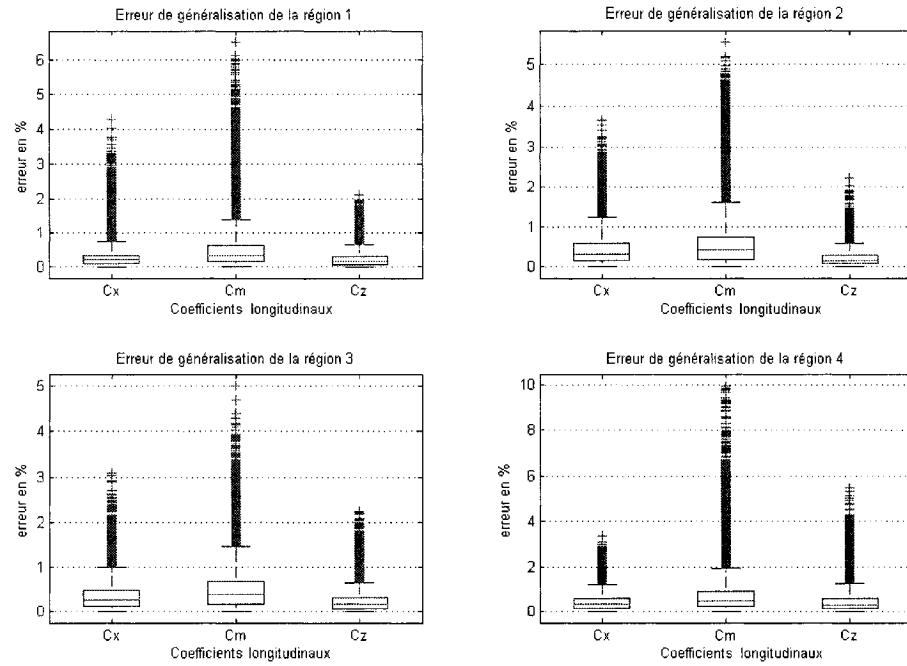


Figure 3.19 Coefficients longitudinaux - Erreur d'entraînement en %

Les boîtes à moustache de Tukey des figures 3.19 et 3.20 permettent d'apprécier la répartition statistique de l'erreur entre les sorties du module d'identification et la valeur réelle des coefficients. Près de 100% des données se trouvent entre les moustaches des boîtes, les bords des rectangles mettent en évidence les premier et troisième quartile, soit 25% et 75% des données. La ligne au milieu des rectangles représente la médiane. Les données non représentatives et exceptionnelles, dites “outlier”, sont ponctuellement marquées par une croix.

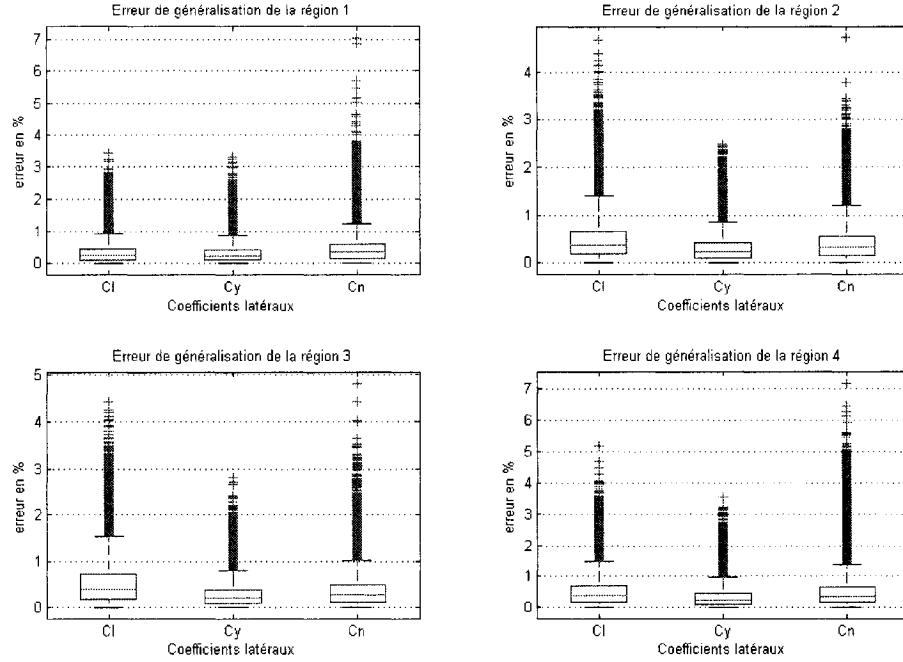


Figure 3.20 Coefficients latéraux - Erreur d'entraînement en %

Les coefficients de force C_x , C_y et C_z sont identifiés à environ 1% près, tandis que les coefficients de moment C_l , C_m et C_n sont identifiés à 1,5% près. Bien que ces performances soient légèrement supérieures aux attentes, elles sont satisfaisantes. Les performances observées ici sont d'ailleurs comparables à la performance des méthodes classiques, qui est de 2,15% pour le filtrage de Kalman étendu par exemple (Jategaonkar, 2006).

Afin de comparer les résultats obtenus avec ceux de la structure en hypercubes, la sortie du module d'identification pour $q = 0^\circ \cdot s^{-1}$, $\beta = 0^\circ$, $\delta_e = 2,2441^\circ$, et $\delta_{lef} = 25^\circ$ est simulée. Cette configuration est identique à celle du paragraphe 3.3.2. La figure 3.21 compare la sortie en C_x avec la valeur “réelle” de ce coefficient. Le résultat obtenu est très satisfaisant compte tenu de la complexité introduite dans l'espace d'entrée par rapport au cas simplifié de l'analyse critique du paragraphe 3.3.2 où seul l'angle d'attaque était amené à varier.

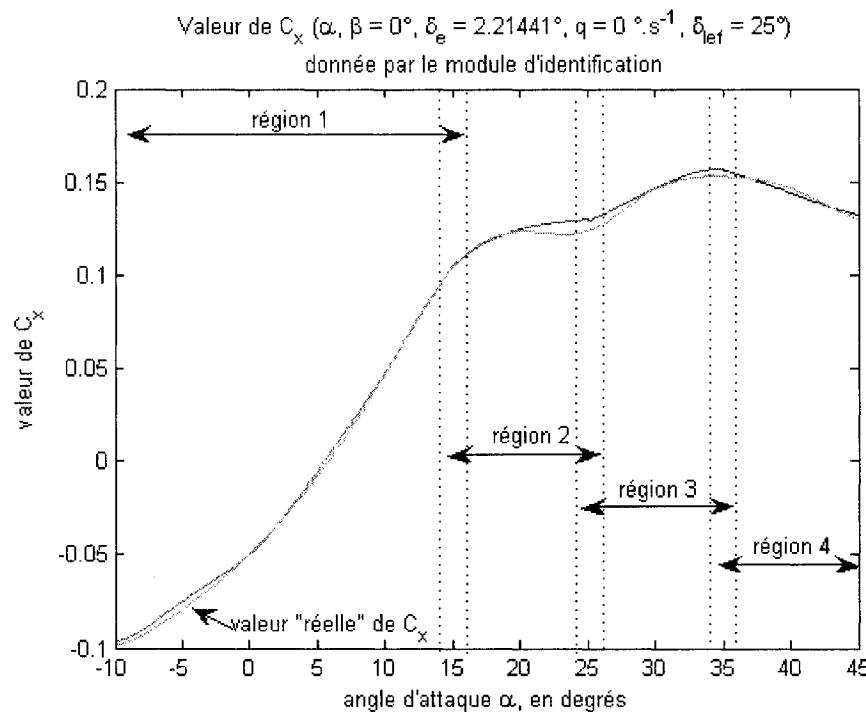


Figure 3.21 Comportement du module d'identification

Les figures 3.22 à 3.25 permettent de visualiser le comportement des réseaux du module d'identification longitudinal pour l'entrée précédente. Les régions de validité des réseaux y ressortent clairement. Dans la majorité des cas, les réseaux conservent leur capacité à extrapoler en dehors de leur données d'entraînement, bien qu'aucune généralité ne puisse être énoncée quant à l'étendue de cette plage.

Les réseaux du module d'identification du modèle aérodynamique du F-16 ont ainsi été développés conformément à la procédure élaborée précédemment et avec des performances conformes aux objectifs fixés dans cette étude.

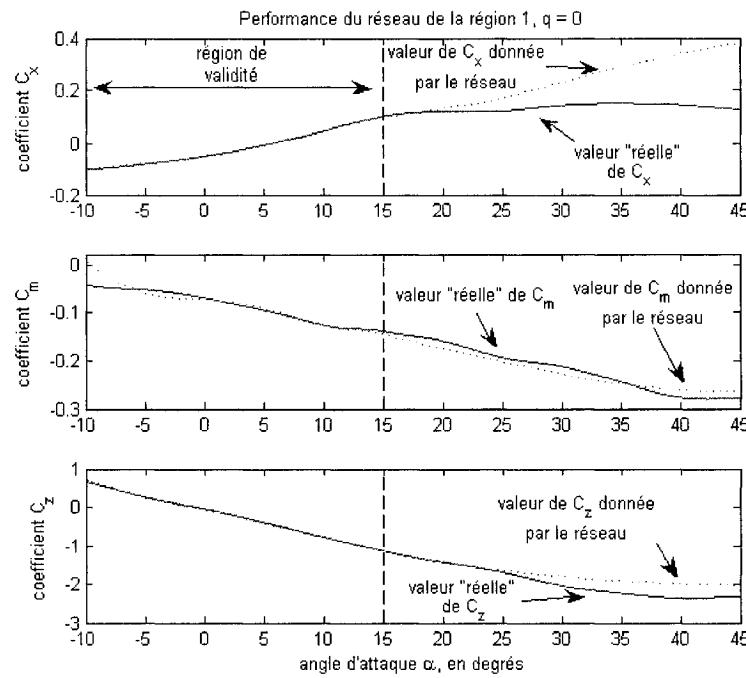


Figure 3.22 Performance du réseau de la région 1 pour $q = 0$

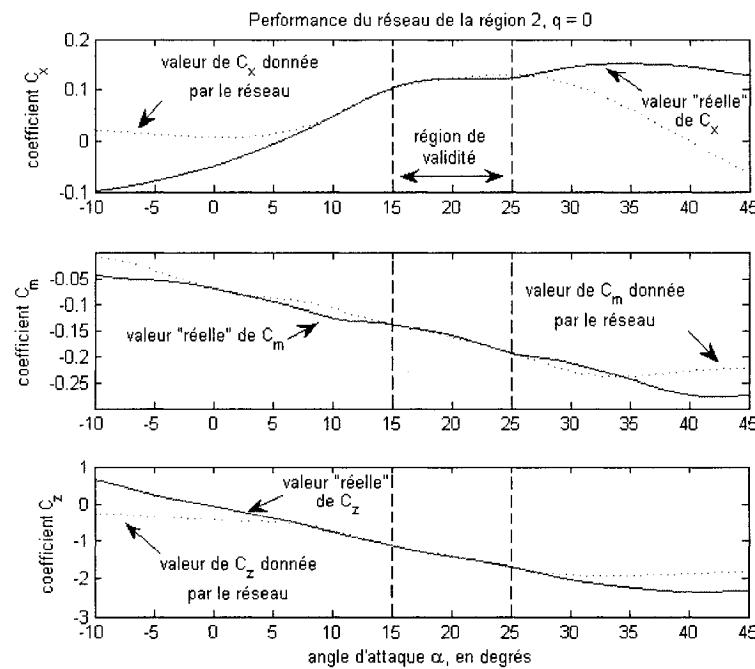


Figure 3.23 Performance du réseau de la région 2 pour $q = 0$

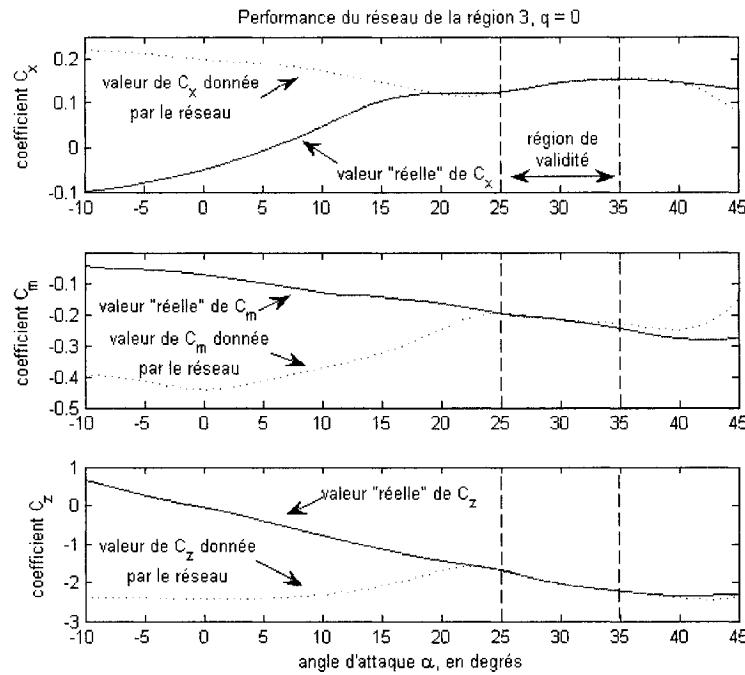


Figure 3.24 Performance du réseau de la région 3 pour $q = 0$

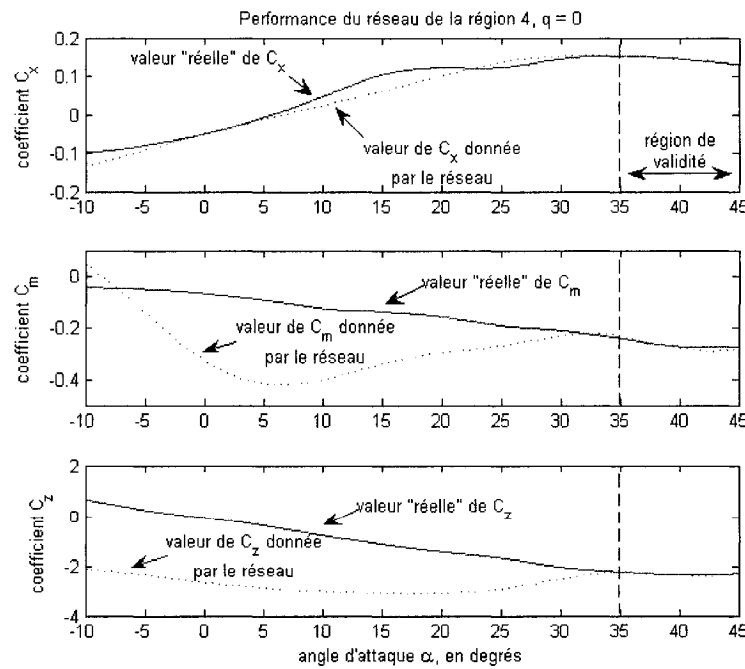


Figure 3.25 Performance du réseau de la région 4 pour $q = 0$

3.6 Conclusion

Le processus d'identification par réseaux de neurones a été mis en évidence dans ce chapitre.

La propriété d'approximateur universel des réseaux de neurones a d'abord été exploitée. Cependant, afin d'être compatible avec une évolution vers des contrôleurs de vol reconfigurables, le module d'identification développé doit permettre une évolution vers l'apprentissage en ligne. Or l'apprentissage en ligne présente un inconvénient majeur, caractéristique de tous les algorithmes d'apprentissage en temps réel : le phénomène d'oubli. L'utilisation des hypercubes permet de s'affranchir dans une certaine mesure de cet inconvénient.

Une architecture novatrice a été conçue, basée sur les conclusions de l'analyse critique de la structure proposée par De Weerdt (DeWeerdt et al., 2005). En particulier, la capacité extensive d'adaptation à tout type de panne structurelle de la structure en hypercubes de De Weerdt a été diminuée au profit d'un nombre plus restreint de réseaux.

La nouvelle architecture a été appliquée au module d'identification du modèle aérodynamique du F-16. Cette application a permis de valider la performance de l'architecture ainsi que la taille restreinte du module. En effet, les coefficients aérodynamiques de force ont été appris à 1% près et ceux des moments à 2% près. Le module compte 4 régions, chacune composées de 5 réseaux dont le nombre de neurones sur la couche cachée varie entre 25 et 61.

L'apprentissage en ligne n'a pas été traité dans le cadre de ce projet, et reste matière à recherche future.

CHAPITRE 4

CONCEPTION DU CONTRÔLEUR

Après avoir modélisé et identifié l'aéronef, celui-ci peut désormais être contrôlé. Les exigences du contrôle, aussi appelées qualités de manœuvrabilité, sont d'abord mises en évidence. La structure globale du contrôleur est ensuite explicitée. Après avoir rapidement exposé la théorie du contrôle utilisé, les différentes composantes du contrôleur sont traitées en détail.

4.1 Qualités de manœuvrabilité

Les spécifications de qualités de manœuvrabilité, ou “flying qualities”, essaient de quantifier la facilité ou bien la difficulté avec laquelle est piloté un avion dans une catégorie donnée de vol. Outre le comportement dynamique de l'avion, l'opinion du pilote peut être influencé par de nombreux facteurs externes tels que la résistance du manche de contrôle, les conditions climatiques ou sa condition physique par exemple. Il est alors courant d'effectuer les tests d'évaluation plusieurs fois avec un grand nombre de pilotes et de prendre une moyenne de tous ces résultats. L'échelle de Cooper-Harper (voir figure 4.1) est un outil systématique pour leur quantification. Elle distingue trois niveaux de manœuvrabilité, répertoriés dans le tableau 4.1. Le premier niveau correspond aux niveaux 1 à 3 de l'échelle de Cooper-Harper, le second aux niveaux 4 à 6 et le troisième aux niveaux 7 à 9.

À chaque niveau correspond des spécifications sur les modes longitudinaux et latéraux de vol. Celles-ci dépendent de plus de la classe de l'avion (voir tableau 4.2) et de la catégorie de la phase de vol (voir tableau 4.3), telles que définies par la

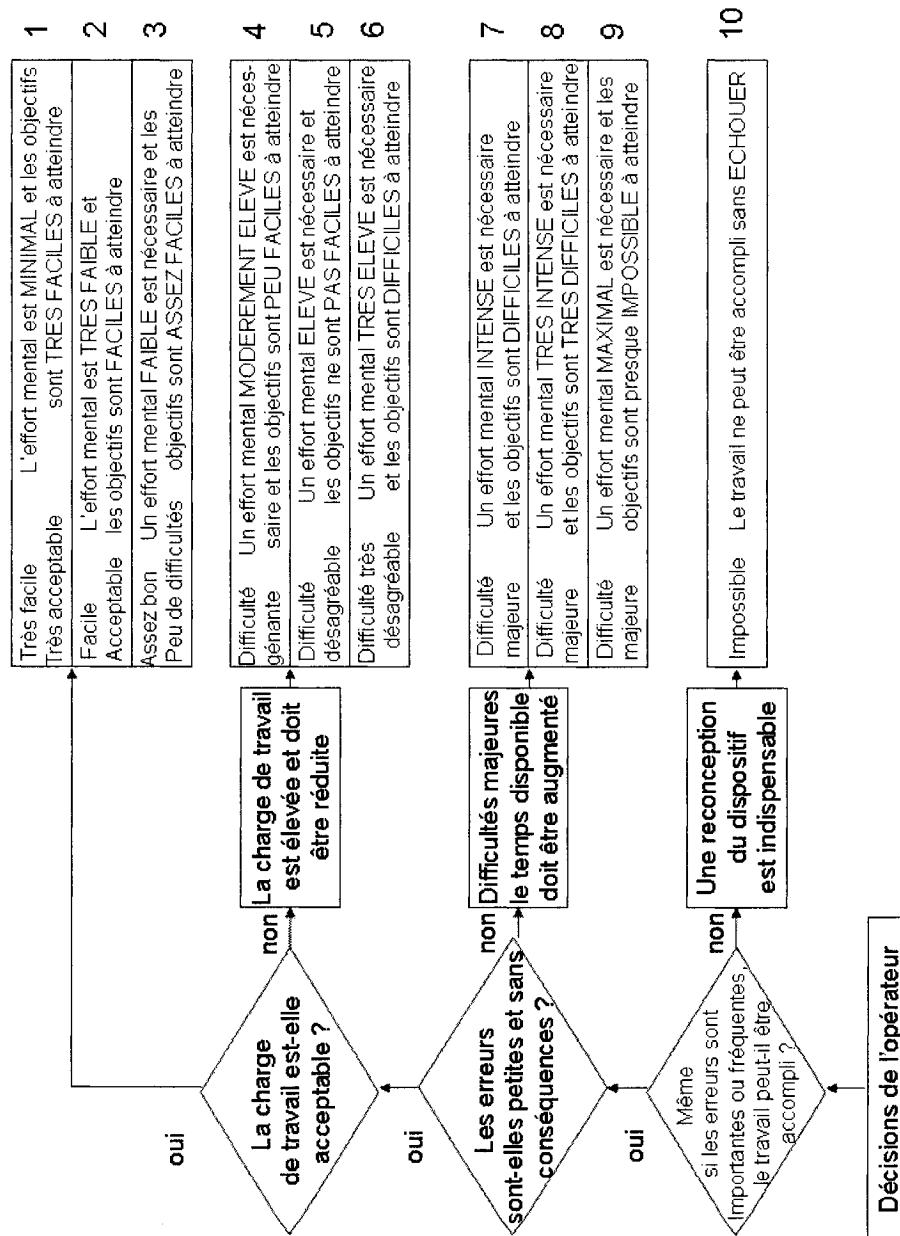


Figure 4.1 Échelle de Cooper-Harper

Tableau 4.1 Niveaux de manœuvrabilité

Niveau	Description
I	Qualités de manœuvrabilité totalement adéquates pour la phase de la mission considérée.
II	Qualités de manœuvrabilité adéquates pour accomplir la phase de la mission considérée lorsque l'effort du pilote est quelque peu augmenté ou que l'efficacité de la mission se dégrade, ou les deux.
III	Qualités de manœuvrabilité telles que la mission peut être contrôlée de manière sécuritaire, mais l'effort du pilote est excessif ou l'efficacité de la mission est inadéquate, ou les deux.

Tableau 4.2 Classes d'avions

Classe	Description
I	Petits avions légers, tels que les aéronefs d'observation légers, utilitaires légers etc.
II	Avions de poids moyen à manœuvrabilité faible à modérée, tels que les aéronefs de sauvetage, de reconnaissance, les avions cargo ou de transport légers et moyens, les bombardiers tactiques, etc.
III	Gros avions lourds à manœuvrabilité faible à modérée, tels que les avions cargo ou de transport lourds, les bombardiers lourds, etc.
IV	Avions hautement manœuvrants, tels que les avions de chasse, de reconnaissance tactique ou d'observation, etc.

Tableau 4.3 Catégorie des phases de vol

Catégorie	Description
A	Phase de vol non terminale qui requiert des manœuvres rapides, une poursuite précise ou le suivi précis d'une trajectoire. Entre autres, sont inclus dans cette catégorie le combat air-à-air, l'attaque au sol, le largage d'armes, la reconnaissance, le ravitaillement en carburant et le vol en formation serrée.
B	Phase de vol non terminale qui requiert des manœuvres graduelles sans besoin de poursuite précise, bien que le suivi précis d'une trajectoire peut être nécessaire. Entre autres, sont inclus dans cette catégorie la montée, le vol en croisière, le ravitaillement en vol (citerne), la descente et les décélérations d'urgence.
C	Phase de vol terminale qui a sinon les mêmes caractéristiques que celles de catégorie B. Entre autres, sont inclus dans cette catégorie le décollage, l'approche, et l'atterrissement.

norme MIL-HDBK-1797A (U.S. DoD, 1997). En condition de vol dite normale, la norme précédente exige que l'aéronef rencontre les spécifications de niveau I. Après une panne mineure, les spécifications sont allégées à celles de niveau II.

4.1.1 Qualités de manœuvrabilité longitudinales

Les qualités de manœuvrabilité longitudinales s'expriment en fonction des coefficients d'amortissement des modes phugoïde et rapide, ainsi que de la pulsation propre de ce dernier.

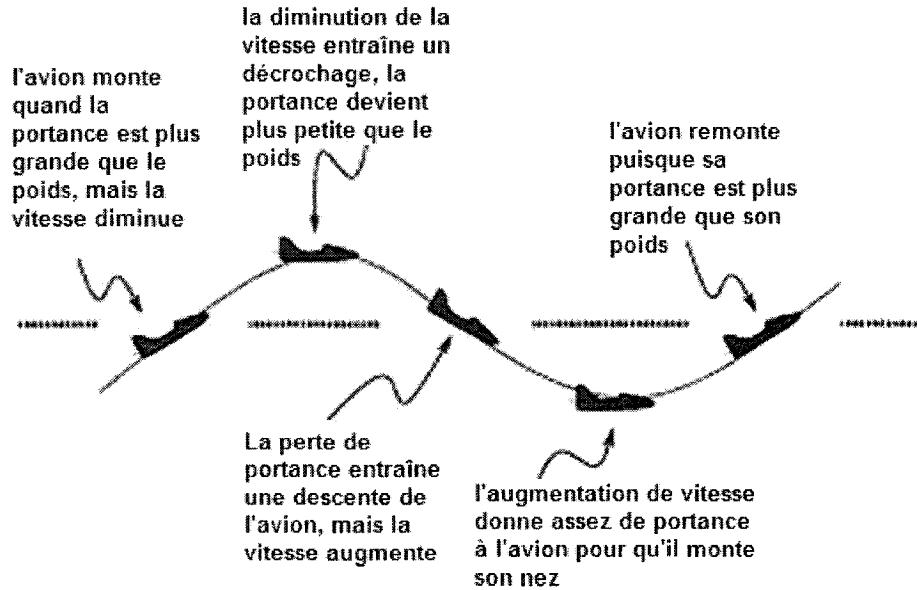


Figure 4.2 Mouvement du mode phugoïde

Tableau 4.4 Coefficient d'amortissement pour le mode phugoïde

Niveau I	$\zeta_{ph} \geq 0,04$
Niveau II	$\zeta_{ph} \geq 0,00$
Niveau III	$T_2 \geq 55s$

Le mode phugoïde est un mode oscillant et basse fréquence. Il est caractérisé par une oscillation lente de la vitesse et de l'altitude, ainsi que de l'assiette de tangage θ et de la pente $\gamma = \theta - \alpha$. La figure 4.2 illustre le mouvement de ce mode.

Les qualités de manœuvrabilité requièrent un coefficient d'amortissement ζ_{ph} conforme aux spécifications du tableau 4.4. En niveau III, le mode phugoïde est supposé instable, soit $\zeta_{ph} < 0$; les spécifications portent alors sur le temps pour doubler d'amplitude T_2 , défini par l'équation 4.1 où ω_{ph} est la pulsation propre du mode phugoïde.

$$T_2 = -\frac{\ln(2)}{\zeta_{ph}\omega_{ph}} \quad (4.1)$$

Le mode rapide, ou “short period”, est un mode rapide fortement amorti. En l’absence de contrôle cette oscillation dure quelques secondes, habituellement moins de 10s, pendant lesquelles l’angle d’attaque α , le taux de tangage q et le facteur de charge n_z varient rapidement tandis que la vitesse longitudinale u reste quasiment constante.

Les spécifications requièrent un coefficient d’amortissement ζ_{mr} qui répond aux caractéristiques récapitulées dans le tableau 4.5 en fonction du niveau de manœuvrabilité et de la classe d’avion. Plutôt que d’imposer des limites sur la pulsation propre du mode rapide ω_{mr} , elles le sont sur un critère additionnel nommé “Control Anticipation Parameter” (CAP). Le CAP est défini comme le ratio suivant:

$$CAP = \frac{\omega_{mr}^2}{n_z/\alpha} \quad (4.2)$$

Il doit être compris entre les limites du tableau 4.6.

Tableau 4.5 Coefficient d’amortissement pour le mode rapide

Catégorie	A et C	B
Niveau I	$0,35 < \zeta_{mr} < 1,30$	$0,30 < \zeta_{mr} < 2,00$
Niveau II	$0,25 < \zeta_{mr} < 2,00$	$0,20 < \zeta_{mr} < 2,00$
Niveau III	$0,15 < \zeta_{mr}$	$0,15 < \zeta_{mr}$

4.1.2 Qualités de manœuvrabilité latérales

Le mode de roulis amorti est un mouvement rapide très amorti durant lequel l’avion a un mouvement de roulis autour de l’axe des x . Il affecte principalement le taux de roulis p , tandis que l’angle de dérapage β , l’azimut ψ et la vitesse angulaire de lacet r restent quasiment constants.

Tableau 4.6 CAP pour le mode rapide

cat.	A	B	C
Niv. I	$0,28 < CAP < 3,6$ $\omega_r \geq 1,0$	$0,085 < CAP < 3,6$	$0,16 < CAP < 3,6$ $\omega_r \geq 0,7$
Niv. II	$0,16 < CAP < 10,0$ $\omega_r \geq 0,6$	$0,038 < CAP < 10,0$	$0,096 < CAP < 10,0$ $\omega_r \geq 0,4$
Niv. III	$0,16 < CAP$	$0,038 < CAP$	$0,096 < CAP$

Le mode spiral est un mode lent qui peut s'avérer légèrement instable. Il donne alors au roulis une amplitude croissante.

Le roulis hollandais est une oscillation latérale et directionnelle, qui affecte principalement l'angle de dérapage β et le taux de lacet r , ce mouvement ressemble à celui des patineurs hollandais d'où son nom. Cependant il est possible qu'il y ait un certain mouvement de roulis dû au couplage roulis-lacet des coefficients aérodynamiques. La figure 4.3 illustre ce mouvement.

Les qualités de manœuvrabilité latérales s'expriment en terme de constante de temps maximale du mode de roulis (voir tableau 4.7), de pulsation propre ω_{rh} et de coefficient d'amortissement ζ_{rh} minimum du mode de roulis hollandais (voir tableau 4.8) et de temps minimal pour doubler l'amplitude en mode spiral (voir tableau 4.9).

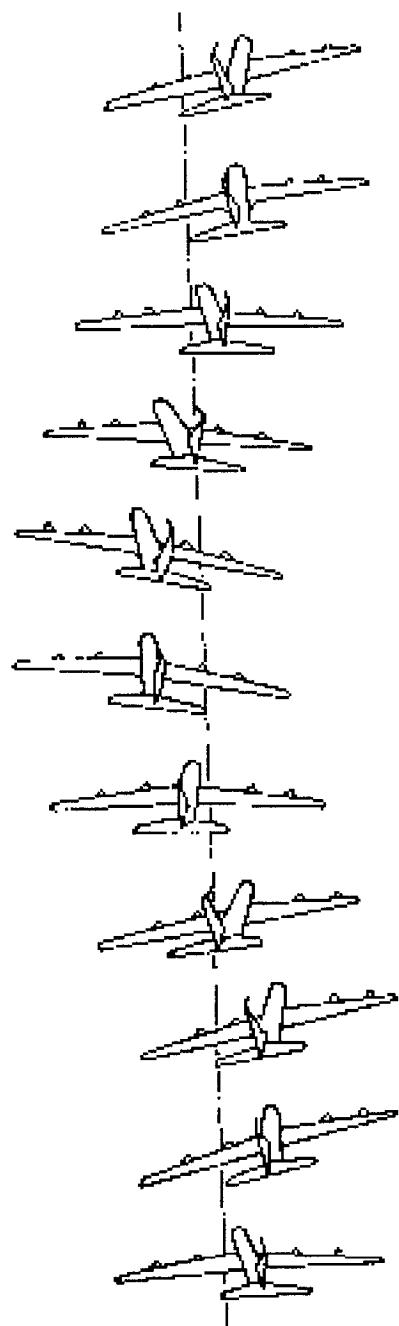


Figure 4.3 Mouvement du roulis hollandais

Tableau 4.7 Constante de temps pour le mode de roulis

classe	cat.	niv. I	niv. II	niv. III
I et IV	A	1,0	1,4	10,0
II et III	A	1,4	3,0	10,0
toutes	B	1,4	3,0	10,0
I et IV	C	1,0	1,4	10,0
II et III	C	1,4	3,0	10,0

Tableau 4.8 Qualités de manœuvrabilité du roulis hollandais

niveau	cat.	classe	min ζ_{rh}	min $\zeta_{rh}\omega_{rh}$	min ω_{rh}
I	A	I et IV	0,19	0,35	1,0
I	A	II et III	0,19	0,35	0,4
I	B	toutes	0,08	0,15	1,0
I	C	I et IV	0,08	0,15	1,0
I	C	II et III	0,08	0,15	0,4
II	toutes	toutes	0,02	0,05	0,4
III	toutes	toutes	0,02	-	0,4

Tableau 4.9 Temps pour doubler l'amplitude du mode spirale

classe	cat.	niv. I	niv. II	niv. III
I et IV	A	12s	12s	4s
I et IV	B et C	20s	12s	4s
II et III	toutes	20s	12s	4s

4.2 Commande de vol par inversion dynamique

L'inversion dynamique est une application particulière de la linéarisation exacte (voir annexe IV). Cette méthode est largement répandue, dans la littérature, mais non dans la pratique. Usuellement elle est appliquée à un modèle non linéaire de l'avion, mais la modélisation des coefficients aérodynamiques y est linéaire. L'objectif de ce chapitre est donc d'appliquer cette méthode au modèle aérodynamique non linéaire choisi dans le cadre de ce mémoire.

4.2.1 Introduction à l'inversion dynamique

Soit le système non linéaire suivant :

$$\dot{x} = f(x) + g(x)u \quad (4.3)$$

où f et g sont des fonctions non linéaires, x est le vecteur d'état du système et u le vecteur de la commande. De plus $\dim(x) = \dim(u)$.

Le choix du contrôle u permet de commander x suivant une dynamique désirée x_{des} . En supposant que g est inversible pour toutes les valeurs de x , ce choix est effectué en “inversant” l'équation 4.3 :

$$u = g^{-1}(x)[\dot{x}_{des} - f(x)] \quad (4.4)$$

Bien que le principe de base de l'inversion dynamique soit simple, certains points méritent d'être discutés.

D'une part g est supposée inversible, or cela n'est pas nécessairement le cas, notamment s'il y a plus de variables d'état que de commande. Cette contrainte est levée

grâce à la séparation temporelle des variables d'état, qui sera discutée en 4.2.2

De plus, supposant g inversible, si $g(x)$ est petit alors l'entrée u devient grande et il y a un risque de saturation des actionneurs. Cette saturation, ainsi que l'incertitude des coefficients aérodynamiques, implique que l'inversion n'est pas parfaite. A des fins de robustesse, l'inversion dynamique est usuellement utilisée dans une boucle interne de contrôle en combinaison avec une loi de commande robuste en boucle externe.

D'autre part le système est supposé affine en la commande, or le modèle général de la dynamique d'un aéronef est :

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \tag{4.5}$$

où f est une fonction non linéaire.

La principale non linéarité provient des coefficients aérodynamiques. Afin d'obtenir un système affine en la commande, deux solutions se présentent. La première est de linéariser les coefficients aérodynamiques. Par exemple pour les coefficients longitudinaux :

$$C_a \approx C_{a_0}(\alpha, \beta) + C_{a_{\delta_e}}(\alpha, \beta)\delta_e$$

La deuxième solution consiste à inclure la dynamique des actionneurs dans les équations d'état. Ces solutions seront discutées en 4.2.5.

4.2.2 Séparation temporelle de la dynamique

Afin de simplifier la structure du contrôleur, certaines hypothèses sur les constantes de temps du système sont faites ((Sun & Clarke, 1994), (Escande, 1997)):

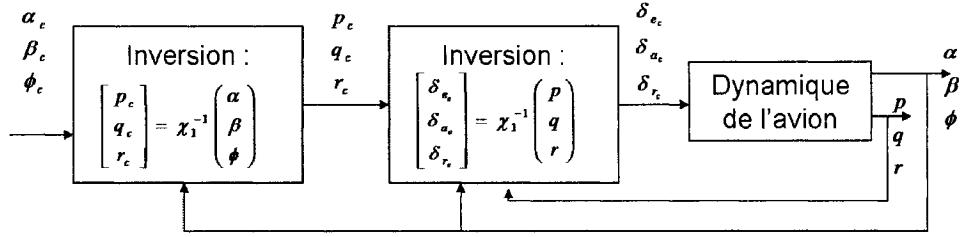


Figure 4.4 Architecture à séparation temporelle de la dynamique

- la dynamique des taux de rotation p, q et r est plus rapide que celle des angles d'incidence, α et β et que celle de l'assiette de roulis ϕ .
- la dynamique des angles (α, β, ϕ) est plus rapide que celle de la vitesse V_t , de l'assiette de tangage θ et de l'azimut ψ

Ces hypothèses permettent de séparer la conception du contrôleur en une cascade de contrôleurs, où les entrées d'un niveau sont les sorties du précédent.

La figure 4.4 met en évidence cette architecture.

La navigation ne faisant pas partie de ce projet, l'azimut ψ n'est pas pris en compte. Il est alors usuel (Escande, 1997) de traiter la vitesse à part, dans une structure appelée “auto-throttle”. Celle-ci est explicitée au paragraphe 4.2.6.

Ainsi, à l'exception de la vitesse, les variables de la boucle la plus externe ne sont pas contrôlées dans le cadre de cette étude.

La dynamique de l'avion peut donc s'écrire sous la forme :

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g(x_1)x_2 \\ \dot{x}_2 &= f_2(x_2) + g(x_2)u \end{aligned} \tag{4.6}$$

où $x_1 = [\alpha, \beta, \phi]^T$ rassemble les variables d'état lentes, $x_2 = [p, q, r]^T$ les rapides et $u = [\delta_e, \delta_a, \delta_r]^T$ celles de la commande.

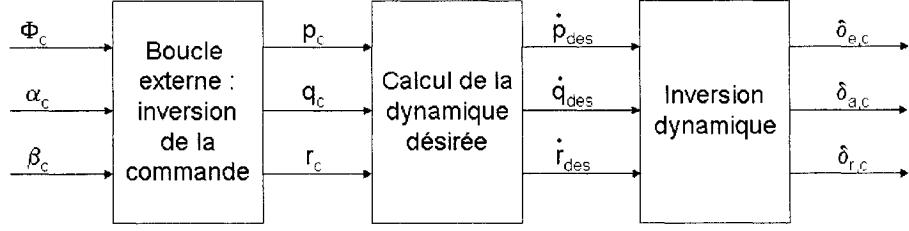


Figure 4.5 Architecture alternative

Une alternative est de calculer directement la commande de la boucle interne $[p_c, q_c, r_c]^T$ à partir de la commande de la boucle externe $[\alpha_c, \beta_c, \phi_c]^T$.

Cette méthode présente l'avantage de réduire le nombre de gains à déterminer. Cependant seule la dynamique de la boucle interne est commandée. De plus, comme il sera vu en 4.2.4, elle requiert la spécification d'une commande additionnelle : celle de l'angle de montée γ_c ou de l'assiette de tangage θ_c .

Soulignons que l'architecture présentée ici est très générale et doit être adaptée au contrôle voulu. Il n'est pas inhabituel par exemple que le pilote fournisse des valeurs de commande en angle d'attaque α_c , angle de dérapage β_c et en taux de roulis p_c . Dans ce cas, la dernière commande est directement utilisée par la boucle d'inversion rapide.

4.2.3 Dynamique désirée

Le calcul de la dynamique désirée des variables contrôlées est une caractéristique du contrôle par inversion dynamique. La commande par retour d'état utilise généralement l'erreur entre la valeur désirée des variables et leur valeur mesurée, ce principe utilisé avec plusieurs formes de dynamique désirée a été mis en évidence dans un rapport de la NASA (Ito et al, 2002).

Le cas proportionnel

Le moyen le plus simple d'obtenir une dynamique désirée est par un retour d'état proportionnel et découplé pour chaque variable d'état x_i :

$$\dot{x}_{ides} = K_{i_p}(x_{i_c} - x_i) \quad (4.7)$$

où K_{i_p} est un gain et x_{i_c} la valeur commandée.

Le gain K_p fixe la bande passante de la réponse. Celle-ci doit être choisie conformément à l'hypothèse de séparation temporelle de la dynamique des états, et ne doit pas exciter des modes non modélisés de la dynamique de l'aéronef.

L'inversion assure $\dot{x}_i = \dot{x}_{ides}$. En boucle fermée, la fonction de transfert de chaque variable d'état x_i est alors celle de l'équation 4.8, et présente un pôle en $s = -K_{i_p}$.

$$\frac{x_i}{x_{i_c}} = \frac{K_{i_p}}{s + K_{i_p}} \quad (4.8)$$

Le cas proportionnel + intégral

Lorsqu'un premier ordre ne permet pas d'obtenir les performances requises, notamment en terme de qualité de manœuvrabilité, alors un système d'ordre plus élevé doit être utilisé. Parmi ces solutions, un retour d'état découpé proportionnel intégral est largement répandu :

$$\dot{x}_{ides} = K_{i_p}(x_{i_c} - x_i) + K_{i_i} \int (x_{i_c} - x) dt \quad (4.9)$$

Le choix des gains K_{i_p} et K_{i_i} permet de fixer l'amortissement ζ_i et la pulsation propre $\omega_{n,i}$ de la variable d'état x_i du système.

Ito et al. suggèrent d'utiliser une forme légèrement différente :

$$\dot{x}_{ides} = K_{i_B} \left(\frac{1}{2}x_{i_c} - x_i \right) + \frac{K_{i_B}^2}{4} \int (x_{i_c} - x_i) dt \quad (4.10)$$

Ainsi il est nécessaire de fixer un seul gain et non plus deux.

La fonction de transfert en boucle fermée de chaque variable détat x_i est alors donnée par l'équation 4.11, et présente un pôle en $s = -\frac{1}{2}K_{i_B}$.

$$\frac{x_i}{x_{i_c}} = \frac{\frac{1}{2}K_{i_B} (s + \frac{1}{2}K_{i_B})}{(s + \frac{1}{2}K_{i_B})^2} = \frac{\frac{1}{2}K_{i_B}}{s + \frac{1}{2}K_{i_B}} \quad (4.11)$$

Le gain peut ainsi être fixé comme pour le gain proportionnel, mais cette forme de dynamique compense à la fois l'erreur en position et en dérivée. La pulsation propre associée à cette commande est $\omega_{i_B} = \frac{1}{2}K_{i_B}$ et l'amortissement est $\zeta_{i_B} = 1$.

L'étude menée par Ito et al. a conclu que l'utilisation d'un gain proportionnel, ou bien du gain proportionnel intégral ci-dessus, est la meilleure solution en terme de coût de contrôle mais aussi du confort des passagers. Cette méthode de choix de la dynamique désirée est donc utilisée.

Le réglage des gains

Les gains de la dynamique désirée doivent être choisis tels que l'aéronef ait un comportement dynamique conforme aux qualités de manœuvrabilité exposées en 4.1.

Dans l'architecture d'inversion dynamique à l'étude, les variables contrôlées sont p , q et r pour l'inversion rapide, α , β et ϕ pour l'inversion lente et V_t pour l'auto-throttle. Or chacune de ces variables peut être associée à un mode.

En effet, dans le modèle longitudinal, le mode rapide agit principalement sur le

taux de tangage q et l'angle d'attaque α , tandis que le mode phugoïde agit sur la vitesse V_t . Le mode de roulis amorti influe sur le taux de roulis p et le mode spiral sur l'assiette de roulis ϕ . Quant au roulis hollandais, il affecte à la fois l'angle de dérapage β et le taux de roulis r .

Les gains doivent donc être choisis tels que les spécifications de ces modes soient satisfaites.

4.2.4 L'inversion de la commande

Le module d'inversion de la commande permet de calculer directement les commandes en taux de rotation p_c , q_c et r_c à partir des valeurs commandées d'angle d'attaque α_c , d'angle de dérapage β_c et d'assiette de roulis ϕ_c .

Rappelons la relation entre les taux de rotation et les angles d'Euler :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4.12)$$

On peut alors remplacer p , q et r par les valeurs de commande recherchées et $\dot{\phi}$, $\dot{\theta}$ et $\dot{\psi}$ par leurs valeurs commandées :

$$\dot{\phi}_c = \frac{d}{dt} \phi_c, \quad \dot{\theta}_c = \frac{d}{dt} \theta_c, \quad \dot{\psi}_c = \frac{d}{dt} \psi_c$$

Comme la navigation n'est pas traitée, on pose $\psi_c = \dot{\psi}_c = 0$. La relation 4.12 devient alors :

$$\begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos \phi \\ 0 & -\sin \phi \end{bmatrix} \begin{bmatrix} \dot{\phi}_c \\ \dot{\theta}_c \end{bmatrix} \quad (4.13)$$

4.2.5 Un système affine par rapport à la commande

L'inversion dynamique requiert un système affine par rapport à la commande. En commande de vol, ceci est généralement obtenu en choisissant des modèles aérodynamiques linéaires. Cependant ce choix n'a pas été fait dans cette étude, bien au contraire.

L'identification des coefficients aérodynamiques par les réseaux de neurones permet de calculer les dérivées partielles de ceux-ci. En effet, un réseau à une couche cachée ayant pour entrée le vecteur $[\alpha, \beta, \delta_e]$ et pour sortie le vecteur $[C_x, C_m, C_z]$, et qui compte m_1 neurones sur sa couche cachée, peut être représenté par la fonction suivante :

$$\begin{bmatrix} C_x \\ C_m \\ C_z \end{bmatrix} = \begin{bmatrix} b_{out} & W_{out} \end{bmatrix} \begin{bmatrix} 1 \\ \varphi(\begin{bmatrix} b_{in} & W_{in} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \beta \\ \delta_e \end{bmatrix}) \end{bmatrix} \quad (4.14)$$

où les matrices W_{out} (dimension $3 \times m_1$) et W_{in} (dimension $m_1 \times 3$) compilent les poids synaptiques de la couche de sortie et de la couche cachée, b_{out} (dimension 3×1) et b_{in} (dimension $m_1 \times 1$) correspondent aux biais des neurones de la couche associée et φ est la fonction sigmoïdale tangente hyperbolique.

De manière analogue un réseau dont le vecteur d'entrée est $[\alpha, \beta, \delta_a, \delta_r]$ et celui de sortie $[C_l, C_y, C_n]$ est représenté par une fonction de même type où W_{in} est de dimension $m_1 \times 4$ et b_{in} de dimension $m_1 \times 4$.

Rappelons que la fonction tangente hyperbolique est définie comme suit :

$$\varphi(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (4.15)$$

et sa dérivée est donc :

$$\frac{\partial \varphi(x)}{\partial x} = 4 \frac{e^{-2x}}{(1 + e^{-2x})^2} = 1 - \varphi^2(x) \quad (4.16)$$

Notons $[y_1, y_2, \dots, y_n]$ les sorties et $[x_1, x_2, \dots, x_m]$ les entrées. La dérivée de la j -ième sortie par rapport à la i -ième entrée est alors :

$$\frac{\partial y_j}{\partial x_i} = w_{out_j} \left[w_{in_i} \cdot * \frac{\partial \varphi}{\partial x} \left(\begin{bmatrix} b_{in} & W_{in} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \right) \right] \quad (4.17)$$

où w_{out_j} est la j -ième ligne de W_{out} , w_{in_i} est la i -ième colonne de W_{in} et $\cdot *$ représente la multiplication matricielle terme-à-terme aussi appelée “dot-product”.

Deux solutions se présentent alors pour rendre le système affine en la commande. La première consiste à linéariser les coefficients aérodynamiques autour d'une position d'équilibre ; la seconde à incorporer la dynamique des actionneurs dans l'état du système.

Si la linéarisation des coefficients est choisie, alors un coefficient $C_a(\alpha, \beta, \delta_e)$ est approximé par son développement de Taylor au premier ordre autour d'un point d'équilibre $[\alpha_0, \beta_0, \delta_{e0}]$:

$$C_a \approx C_a(\alpha_0, \beta_0, \delta_{e0}) + \frac{\partial C_a}{\partial \alpha} \Delta \alpha + \frac{\partial C_a}{\partial \beta} \Delta \beta + \frac{\partial C_a}{\partial \delta_e} \Delta \delta_e \quad (4.18)$$

où $\Delta \alpha = \alpha - \alpha_0$, $\Delta \beta = \beta - \beta_0$ et $\Delta \delta_e = \delta_e - \delta_{e0}$.

Cette approximation implique que la conception du contrôleur est faite autour du point d'équilibre en question, et du “scheduling” sera alors nécessaire. Cette

méthode n'est donc pas retenue. Toutefois, une linéarisation instantanée au point de fonctionnement peut servir dans le cadre d'une commande adaptative.

Si la dynamique des actionneurs est prise en compte, alors la connaissance de la dérivée par rapport au temps des coefficients est nécessaire :

$$\frac{dC_a}{dt} = \frac{\partial C_a}{\partial \alpha} \frac{\partial \alpha}{\partial t} + \frac{\partial C_a}{\partial \beta} \frac{\partial \beta}{\partial t} + \frac{\partial C_a}{\partial \delta_e} \frac{\partial \delta_e}{\partial t} \quad (4.19)$$

Un inconvénient de cette méthode est que le nombre de variables d'état du système augmente. Cependant elle présente l'avantage de ne pas introduire d'erreur supplémentaire dans les expressions des coefficients aérodynamiques.

Une solution innovante est de considérer les coefficients aérodynamiques en moment C_l , C_m et C_n comme les entrées du système. Celui-ci est alors déjà sous la forme affine en la commande, il n'y a donc ni approximation à faire, ni calcul fastidieux des dérivées aérodynamiques. Les réseaux de neurones fournissent une formulation analytique des coefficients aérodynamiques qui est alors utilisée par un algorithme de programmation linéaire pour calculer la valeur des déflexions des surfaces de contrôle donnant les valeurs désirées des coefficients aérodynamiques en moment. Cette solution présente l'avantage d'être facilement adaptable au nombre de surface de contrôle de l'aéronef contrôlé.

4.2.6 La commande en vitesse

La commande de la vitesse totale de l'avion V_t est traitée, séparément des autres commandes, par une application spécifique communément appelée "autothrottle" (Escande, 1997). Cette séparation permet de simplifier la structure du contrôleur et est justifiée par le fait que le temps de réponse est plus élevé que celui des autres commandes.

La poussée T nécessaire est calculée par inversion de l'équation I.6 de l'annexe I et en remplaçant la dérivée de la vitesse \dot{V}_t par celle désirée $\dot{V}_{t_{des}}$:

$$T = \frac{m}{\cos \alpha \cos \beta} \left\{ \dot{V}_{t_{des}} + \frac{\bar{q}S}{m} (-C_x \cos \alpha \cos \beta - C_y \sin \beta - C_z \sin \alpha \cos \beta) - g (\cos \phi \cos \theta \sin \alpha \cos \beta + \sin \phi \cos \theta \sin \beta - \sin \theta \cos \alpha \cos \beta) \right\} \quad (4.20)$$

En choisissant la pulsation propre ω_V et l'amortissement ζ_V conformes aux spécifications du mode phugoïde, la dynamique suivante permet d'assurer la qualité de manœuvrabilité requise :

$$\frac{V_t}{V_{t_c}} = \frac{\omega_V^2}{s^2 + 2\zeta_V\omega_V s + \omega_V^2} \quad (4.21)$$

La valeur du coefficient aérodynamique C_x est nécessairement donnée par le module d'identification développé au chapitre 3. Celle des coefficients C_y et C_z peut être estimée par celle donnée par le module d'identification, ou bien calculée à partir des mesures des accéléromètres a_y et a_z .

Le modèle de la poussée du moteur doit être inversé afin de fournir la commande en manette des gaz δ_{th} . Généralement celui-ci est simple et le calcul se fait directement. Cependant, dans le cas du F-16 par exemple, le modèle du moteur est plus complexe. Dans ce type de cas, il est proposé d'effectuer l'inversion grâce à un réseau de neurones.

4.2.7 Etude d'un modèle longitudinal simplifié

Afin de se familiariser avec le concept de l'inversion dynamique, il est appliqué à un modèle longitudinal simplifié : celui-ci est linéaire et invariant dans le temps.

Soit le modèle longitudinal suivant :

$$\begin{aligned}\dot{\alpha} &= Z_\alpha \alpha + q \\ \dot{q} &= M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e\end{aligned}\tag{4.22}$$

où $x = [\alpha, q]^T$ est le vecteur d'état du système et $u = \delta_e$ la commande.

Application à la boucle interne : l'inversion rapide

Le système est déjà affine en la commande, il n'y a donc pas lieu d'étudier la linéarisation du coefficient aérodynamique en moment de tangage C_m .

On suppose que M_{δ_e} est non nul. On a alors :

$$\delta_e = \frac{1}{M_{\delta_e}} [\dot{q} - M_\alpha \alpha - M_q q] \tag{4.23}$$

On remplace ensuite \dot{q} par la dynamique désirée dans l'équation 4.23 :

$$\delta_{e_c} = \frac{1}{M_{\delta_e}} [\dot{q}_{des} - M_\alpha \alpha - M_q q] \tag{4.24}$$

La dynamique désirée est choisie de forme proportionnelle :

$$\dot{q}_{des} = K_q (q_c - q) \tag{4.25}$$

La dynamique de q doit répondre aux spécifications de qualité de manœuvreabilité liées au mode rapide. Or celles-ci sont du second ordre. Afin de fixer le gain K_q , la dynamique de l'actionneur est incorporée a posteriori. L'actionneur est supposé régi par un premier ordre de constante de temps T_e :

$$\dot{\delta}_e = \frac{1}{T_e} (\delta_{e_{com}} - \delta_e) \tag{4.26}$$

En compilant l'ensemble des équations ci-dessus on obtient alors le système multi-

variable suivant :

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha & 1 & 0 \\ M_\alpha & M_q & M_{\delta_e} \\ -\frac{M_\alpha}{T_e M_{\delta_e}} & -\frac{K_q + M_q}{T_e M_{\delta_e}} & -\frac{1}{T_e} \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_q}{T_e M_{\delta_e}} \end{bmatrix} q_c \quad (4.27)$$

La fonction de transfert $\frac{q(s)}{q_c(s)}$ est obtenu par la transformation de Laplace usuelle de $C(sI - A)^{-1}B$:

$$\frac{K_q(s - Z_\alpha)}{T_e s^3 + (1 - M_q T_e - Z_\alpha T_e) s^2 + (K_q - Z_\alpha(1 - M_q T_e) - M_\alpha T_e) s - Z_\alpha(K_q + M_q)} \quad (4.28)$$

Le lieu des racines permet alors de fixer le gain K_q tel que les spécifications en amortissement et en pulsation propre soient satisfaites au mieux.

Lorsque les spécifications ne sont pas satisfaites, une solution consiste à utiliser une dynamique désirée d'ordre plus élevé. Cependant celle-ci peut engendrer d'autres inconvénients tels que des dépassemens importants qui entraînent la saturation des actionneurs.

Quand la dynamique désirée est de type proportionnel + intégral, la formulation de l'équation 4.10 peut être utilisée lorsque les spécifications correspondent à un deuxième ordre. C'est le cas ici pour le taux de tangage. Ainsi :

$$\ddot{q}_{des} = 2\zeta_q \omega_q (\dot{q}_c - \dot{q}) + \omega_q^2 (q_c - q) \quad (4.29)$$

où ω_q est la pulsation propre du mode rapide et ζ_q son amortissement, fixés dans les plages autorisées par les qualités de manœuvre.

Lorsque les spécifications sont du premier ordre, la formulation de l'équation 4.9 est utilisée en prenant un gain qui donne la bande passante spécifiée.

Dans le cas où la dynamique de l'actionneur n'est pas négligée lors du design, le système a la forme suivante :

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha & 1 & 0 \\ M_\alpha & M_q & M_{\delta_e} \\ 0 & 0 & -\frac{1}{T_e} \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_e} \end{bmatrix} \delta_{e_c} \quad (4.30)$$

La sortie $y = q$ est dérivée jusqu'à ce que δ_{e_c} apparaisse dans l'équation :

$$\ddot{q} = M_\alpha Z_\alpha \alpha + M_\alpha q + M_q \dot{q} - \frac{M_{\delta_e}}{T_e} \delta_e + \frac{M_{\delta_e}}{T_e} \delta_{e_c} \quad (4.31)$$

L'inversion de la commande et le remplacement de \ddot{q} par la dynamique désirée donne:

$$\delta_{e_c} = \frac{T_e}{M_{\delta_e}} (\ddot{q}_{des} - M_\alpha Z_\alpha \alpha - M_\alpha q - M_q \dot{q}) + \delta_e \quad (4.32)$$

En choisissant \ddot{q}_{des} comme dans l'équation 4.29 les spécifications sont satisfaites.

Ces différentes méthodologies sont appliquées à la commande du taux de tangage du modèle longitudinal du F-16 "trimmé" autour de la position d'équilibre suivante: $V_t = 150m.s^{-1}$ et $h = 5000m$. La figure 4.6 présente le schéma bloc de la simulation. On a $M_\alpha = -5,1047$, $M_q = -3,1226$ et $M_{\delta_e} = -0,38014$. De plus à l'équilibre $\alpha = 0,0147rad$ et $n_z = -1$.

L'amortissement du mode rapide désiré est fixé à $\zeta_{mr_{des}} = 0,7$ (valeur choisie arbitrairement dans la plage $[0, 35; 1, 3]$) et le CAP désiré est 1,5 (choisie de même dans la plage $[0, 28 ; 3, 6]$). La connaissance de l'angle d'attaque à l'équilibre et du facteur de charge vertical permet de déduire la pulsation propre désirée :

$$\omega_{mr_{des}} = \sqrt{CAP \frac{n_z}{\alpha}} = 10rad.s^{-1}$$

Le gain K_q de la première méthode est d'abord déterminé par placement de pôles.

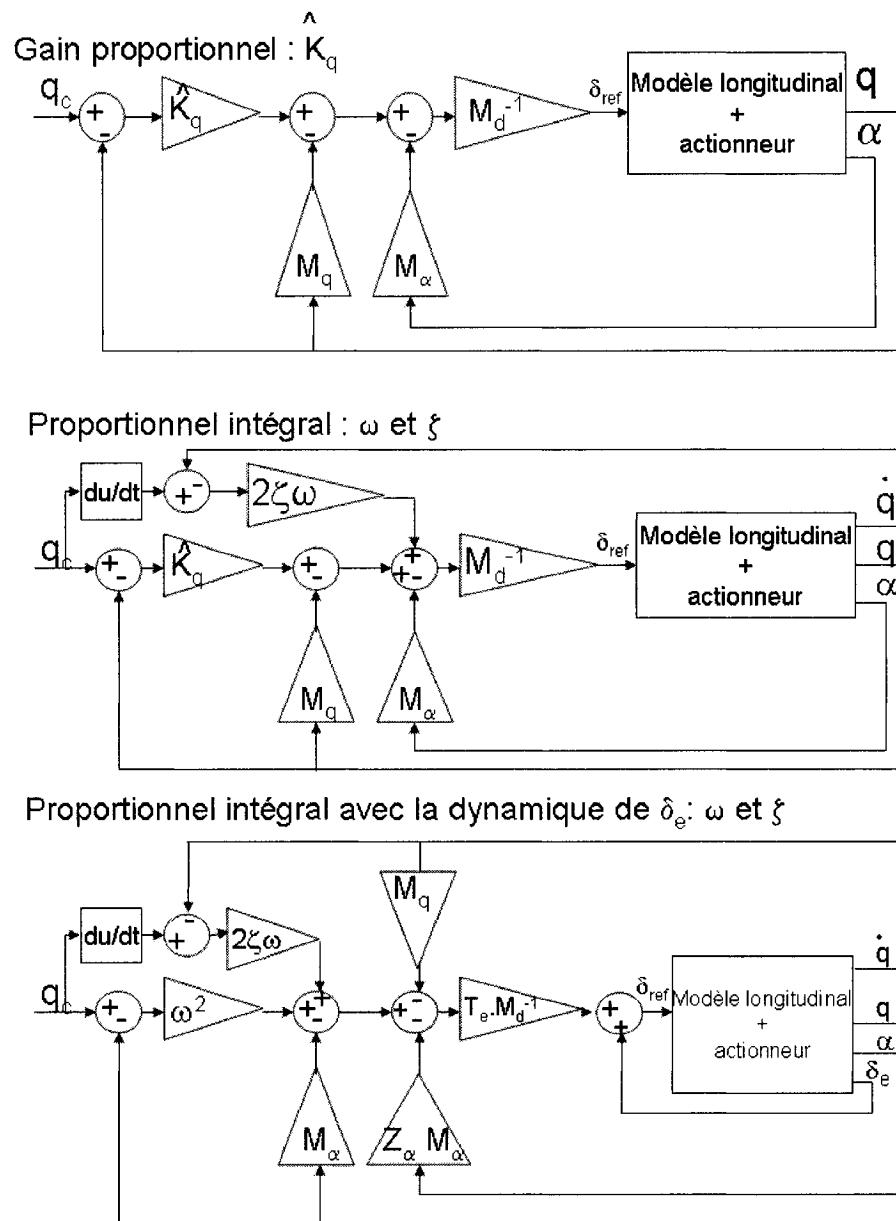


Figure 4.6 Schémas bloc comparant trois configurations de commande par inversion dynamique

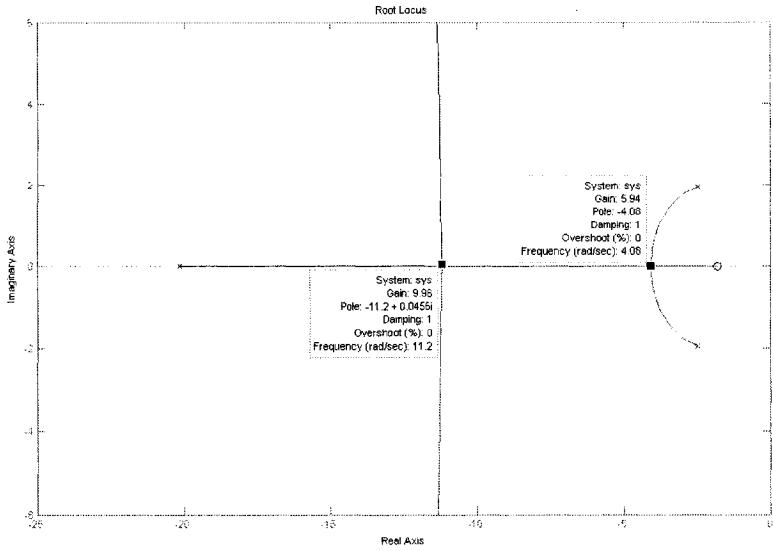


Figure 4.7 Lieu des racines pour la première méthode

La figure 4.7 met en évidence le lieu des racines. Les valeurs désirée de pulsation propre et d'amortissement ne peuvent pas être atteintes simultanément, un compromis est donc fait.

Cependant le calcul en ligne du gain K_p n'est pas évident, car il est associé à un retour de sortie et non à un retour d'état. Baba, Takano et Sano (1995) proposent de négliger Z_α afin d'estimer le gain K_q par \hat{K}_q . La fonction de transfert entre le taux de tangage q et le taux de tangage commandé q_c devient alors :

$$\frac{q(s)}{q_c(s)} = \frac{\hat{K}_q}{T_e s^2 + (1 - M_q T_e) s + (\hat{K}_q - M_\alpha T_e)} \quad (4.33)$$

Si ζ_{mr} est l'amortissement désiré pour le mode rapide, par identification du dénominateur normalisé de l'équation 4.33 et de l'expression $s^2 + 2\zeta_{mr}\omega_{mr}s + \omega_{mr}^2$ on obtient

$$\hat{K}_q = T_e \left(M_\alpha + \frac{1}{4\zeta_{mr}^2} \left(\frac{1}{T_e} - M_q \right)^2 \right) \quad (4.34)$$

On remarque cependant que la pulsation propre de la fonction de transfert est alors imposée et ne peut pas être librement fixée à une valeur désirée :

$$\omega_{mr} = \frac{1}{2\zeta_{mr}} \left(\frac{1}{T_e} - M_q \right) \quad (4.35)$$

En qualité de manœuvrabilité de niveau I, le CAP doit être inférieur à 3.6. La pulsation propre maximale dans la condition considérée est donc $15,65\text{rad.s}^{-1}$. Cependant le gain calculé avec l'équation 4.34 induit une pulsation propre de $16,66\text{rad.s}^{-1}$. Il est donc nécessaire de modifier le choix de l'amortissement : $\zeta_{mr_{des,mod}} = 1$, ce qui entraîne $\omega_{mr_{des,mod}} = 11,67\text{rad.s}^{-1}$.

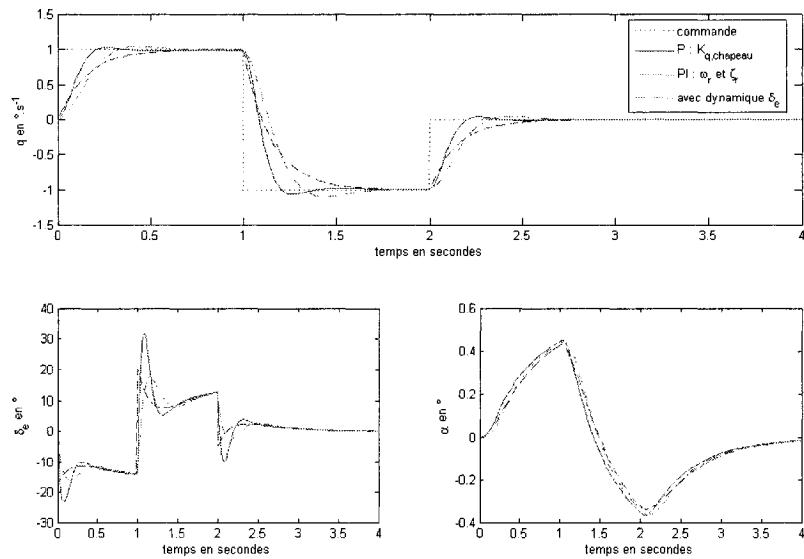


Figure 4.8 Réponse en tangage suite à un doublet de $1^\circ.\text{s}^{-1}$ en commande

On observe (voir figure 4.8) la réponse en tangage ainsi que l'effort de contrôle associé, suite à un doublet d'une amplitude de $1^\circ.\text{s}^{-1}$ et d'une durée de 2s sur la commande en tangage.

Comme attendu, la réponse lorsque le contrôle se fait par le gain K_q est plus rapide que lorsque le contrôle comprend la pulsation propre et l'amortissement désirés.

Cependant elle présente une erreur non nulle en régime permanent

$$e_{perm} = \frac{-M_q}{Z_\alpha K_q + M_q} \quad (4.36)$$

L'effort de contrôle est également plus important avec la commande établie par le gain K_q , avec une amplitude un tiers plus grande lors du pic faisant suite au changement de consigne à $t = 1s$. L'utilisation d'une commande de type proportionnel + intégral est donc préférée.

Le temps de réponse du système, lorsque la dynamique des actionneurs est intégrée dans la loi de commande, est plus rapide que lorsqu'elle ne l'est pas. De plus l'amplitude et la vitesse de la déflection de l'actionneur sont réduites.

Ces considérations motivent le choix d'utiliser une loi de commande de type proportionnel + intégral et d'incorporer la dynamique des actionneurs.

Application à la boucle externe

Deux solutions sont envisageables: soit l'inversion de commande telle que décrite au paragraphe 4.2.4, soit une boucle d'inversion dynamique ayant pour commande les états de la boucle interne.

Pour l'inversion de la commande, supposons que le pilote fournisse directement la commande en assiette de tangage θ_c . La commande en taux de tangage q_c est alors directement obtenue par l'équation suivante :

$$q_c = \frac{d}{dt} \theta_c \quad (4.37)$$

Cependant la dérivation d'une consigne doit être calculée. Or la consigne présente des discontinuités, au moment de changement de consigne par exemple, qui implique que la dérivée n'est pas définie. Cette méthode n'est donc pas retenue.

Lorsqu'une boucle d'inversion externe est utilisée, il est usuel de supposer que la

dynamique du taux de tangage est “très rapide”, de sorte que le taux de tangage q peut être remplacé par le taux de tangage commandé q_c dans l’équation de la dynamique de la variable lente. La dynamique de l’angle d’attaque α est alors :

$$\dot{\alpha} = Z_\alpha \alpha + q_c \quad (4.38)$$

L’inversion de la dynamique donne :

$$q_c = \dot{\alpha}_{des} - Z_\alpha \alpha \quad (4.39)$$

où la dynamique désirée de l’angle d’attaque peut être choisie de type proportionnel ou proportionnel + intégral :

$$\begin{cases} \dot{\alpha}_{des} = K_\alpha(\alpha_c - \alpha) \\ \dot{\alpha}_{des} = K_\alpha(\frac{1}{2}\alpha_c - \alpha) + \frac{K_\alpha^2}{4s}(\alpha_c - \alpha) \end{cases} \quad (4.40)$$

Il a été mis en évidence (voir 4.2.3), que ces deux commandes donnent une fonction de transfert du premier ordre de constante de temps ω_α égale à K_α ($0,5K_\alpha$ respectivement). Le temps de réponse en angle d’attaque devant être plus lent que celui en taux de tangage pour que l’hypothèse de séparation temporelle soit valide, on choisit ω_α inférieur à la pulsation propre ω_q mais toujours compris dans la plage des pulsations propres admissibles donnée par les spécifications du mode rapide.

Ces deux types de génération de la dynamique désirée sont testés par simulation sur le même exemple que précédemment.

On observe la réponse en angle d’attaque suite à un doublet de 5° et d’une durée de $4s$ (voir figure 4.9). La commande en taux de tangage q_c , ainsi que le taux de tangage q sont également observés, de même que l’effort sur l’actionneur.

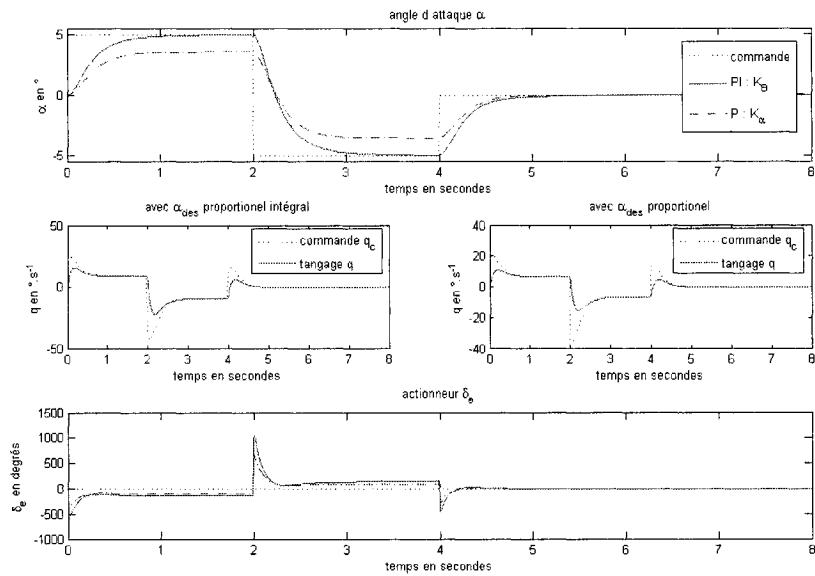


Figure 4.9 Réponse suite à un doublet de 5° en commande

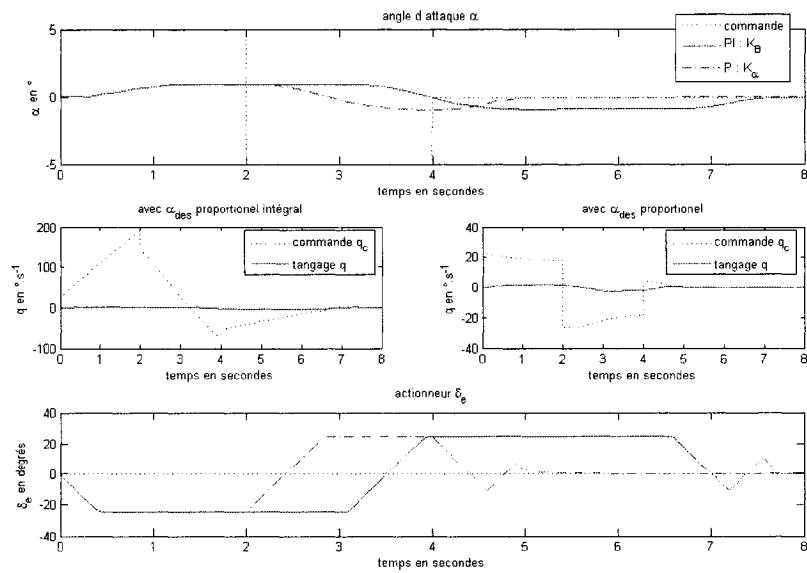


Figure 4.10 Réponse avec saturation suite à un doublet de 5° en commande

L'utilisation du cas proportionnel + intégral a l'avantage d'annuler l'erreur en régime permanent présente dans le cas proportionnel. Toutefois c'est au prix d'un effort de contrôle plus important. L'élévateur ne peut pas dépasser 25° sur le F-16, cette limite est cependant atteinte avec les deux types de contrôle. Le module de simulation est alors modifié pour tenir compte de la saturation de l'actionneur. On observe que les deux types de commandes entraînent une saturation de l'actionneur (voir figure 4.10). Cet effet a une importance capitale qu'il ne faut pas négliger lors de la conception du contrôleur.

En faisant abstraction de la saturation, les considérations ci-dessus mènent à choisir une dynamique désirée de type proportionnel + intégral.

4.2.8 Application au modèle à 6 degrés de liberté

L'inversion dynamique est désormais appliquée au modèle non linéaire à 6 degrés de liberté d'un aéronef. La boucle interne est d'abord conçue : elle prend les vitesses angulaires de roulis p , de tangage q et de lacet r en entrée et fournit la commande à appliquer sur les surfaces de contrôle de l'avion (δ_e pour l'élévateur, δ_a pour les ailerons et δ_r pour le gouvernail). La boucle externe est ensuite conçue. Elle a pour entrée les angles, d'attaque α et de dérapage β , et l'assiette de roulis ϕ ; les sorties sont les commandes en vitesses angulaires p , q et r .

Conception de la boucle interne

Les équations non linéaires qui régissent les vitesses angulaires sont données en annexe I, et sont rappelées ici :

$$\begin{aligned}\dot{p} &= (c_1r + c_2p - c_4h_{moteur})q + \bar{q}Sb(c_3C_l + c_4C_n) \\ \dot{q} &= (c_5p + c_7h_{moteur})r - c_6(p^2 - r^2) + c_7\bar{q}S\bar{c}C_m \\ \dot{r} &= (c_8p - c_2r - c_9h_{moteur})q + \bar{q}Sb(c_4C_l + c_9C_n)\end{aligned}\tag{4.41}$$

Les coefficients aérodynamiques y sont obtenus par le module d'identification décrit à la section 3, et se décomposent de la façon suivante :

$$\begin{aligned} C_l &= C_{l_0}(\alpha, \beta, \delta_a, \delta_r) + \frac{pb}{2V_t} C_{l_p}(\alpha, \beta, \delta_a, \delta_r) + \frac{rb}{2V_t} C_{l_r}(\alpha, \beta, \delta_a, \delta_r) \\ C_m &= C_{m_0}(\alpha, \beta, \delta_e) + \frac{q\bar{C}}{2V_t} C_{m_q}(\alpha, \beta, \delta_e) \\ C_n &= C_{n_0}(\alpha, \beta, \delta_a, \delta_r) + \frac{pb}{2V_t} C_{n_p}(\alpha, \beta, \delta_a, \delta_r) + \frac{rb}{2V_t} C_{n_r}(\alpha, \beta, \delta_a, \delta_r) \end{aligned} \quad (4.42)$$

Les qualités de manœuvrabilité suggèrent de choisir une dynamique désirée équivalente à un premier ordre en roulis et lacet obtenu grâce à la seconde formulation du type proportionnel + intégral (voir 4.2.3) et de second ordre en tangage :

$$\begin{aligned} \dot{p}_{des} &= \omega_p(p_c - p) \\ \dot{q}_{des} &= 2\zeta_q \omega_q(q_c - q) + \omega_q^2 \int (q_c - q) \\ \dot{r}_{des} &= \omega_r(r_c - r) \end{aligned} \quad (4.43)$$

où ω_p est conforme à la bande passante du mode de roulis amorti, ω_r à celle du roulis hollandais et (ω_q, ζ_q) aux caractéristiques du mode rapide longitudinal.

Le système est rendu affine en la commande selon la seconde méthode retenue au paragraphe 4.2.5 (équation 4.19). A cette fin, la dynamique des actionneurs est supposée du premier ordre :

$$\begin{aligned} \dot{\delta}_e &= \frac{1}{T_e} (\delta_{e_c} - \delta_e) \\ \dot{\delta}_a &= \frac{1}{T_a} (\delta_{a_c} - \delta_a) \\ \dot{\delta}_r &= \frac{1}{T_r} (\delta_{r_c} - \delta_r) \end{aligned} \quad (4.44)$$

Lorsque les déflections des surfaces de contrôle sont prises comme entrée et que la

dynamique des actionneurs (voir équation 4.44) est prise en compte, en dérivant les équations 4.41 par rapport au temps, les commandes δ_{e_c} , δ_{a_c} et δ_{r_c} apparaissent. L'inversion des équations obtenues et le remplacement de \ddot{p} , \ddot{q} et \ddot{r} par les dynamiques désirées donne la relation suivante entre les commandes et les dynamiques désirées :

$$\begin{bmatrix} \delta_{e_c} \\ \delta_{a_c} \\ \delta_{r_c} \end{bmatrix} = \begin{bmatrix} 0 & \frac{p_1}{T_a} & \frac{p_2}{T_r} \\ \frac{q_1}{T_e} & 0 & 0 \\ 0 & \frac{r_1}{T_a} & \frac{r_2}{T_r} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \ddot{p}_{des} \\ \ddot{q}_{des} \\ \ddot{r}_{des} \end{bmatrix} - \begin{bmatrix} p_3 & p_4 & p_5 & p_6 & p_7 & 0 & -\frac{p_1}{T_a} & -\frac{p_2}{T_r} \\ q_2 & q_3 & 0 & q_4 & q_5 & -\frac{q_1}{T_q} & 0 & 0 \\ r_3 & r_4 & r_5 & r_6 & r_7 & 0 & -\frac{r_1}{T_a} & -\frac{r_2}{T_r} \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\delta}_e \\ \dot{\delta}_a \\ \dot{\delta}_r \end{bmatrix} \right\} - \begin{bmatrix} p_8 \\ q_6 \\ r_8 \end{bmatrix} \quad (4.45)$$

où

$$\begin{aligned} p_1 &= \bar{q}Sb(c_3C_{l_{\delta_a}} + c_4C_{n_{\delta_a}}) \\ p_2 &= \bar{q}Sb(c_3C_{l_{\delta_r}} + c_4C_{n_{\delta_ar}}) \\ p_3 &= \bar{q}Sb \left(c_3 \frac{\partial C_{l_0}}{\partial \alpha} + c_4 \frac{\partial C_{n_0}}{\partial \alpha} \right) \\ p_4 &= \bar{q}Sb \left(c_3 \frac{\partial C_{l_0}}{\partial \beta} + c_4 \frac{\partial C_{n_0}}{\partial \beta} \right) \\ p_5 &= \frac{\bar{q}Sb^2}{2V_t} (c_3C_{l_p} + c_4C_{n_p}) \\ p_6 &= -c_4h_{moteur} \\ p_7 &= \frac{\bar{q}Sb^2}{2V_t} (c_3C_{l_r} + c_4C_{n_r}) \end{aligned} \quad (4.46)$$

$$\begin{aligned}
p_8 = & \bar{q}Sb \left\{ c_3 \left(\frac{pb}{2V_t} \left(\frac{\partial C_{l_p}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{l_p}}{\partial \beta} \dot{\beta} \right) + \frac{rb}{2V_t} \left(\frac{\partial C_{l_r}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{l_r}}{\partial \beta} \dot{\beta} \right) \right) \right. \\
& + c_4 \left(\frac{pb}{2V_t} \left(\frac{\partial C_{n_p}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{n_p}}{\partial \beta} \dot{\beta} \right) + \frac{rb}{2V_t} \left(\frac{\partial C_{n_r}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{n_r}}{\partial \beta} \dot{\beta} \right) \right) \left. \right\} \\
& + c_1 (\dot{r}q + r\dot{q}) + c_2 (\dot{p}q + p\dot{q})
\end{aligned}$$

$$\begin{aligned}
q_1 &= \bar{q}S\bar{c}c_7 \left(\frac{\partial C_{m_0}}{\partial \delta_e} + \frac{q\bar{c}}{2V_t} \frac{\partial C_{m_q}}{\partial \delta_e} \right) \\
q_2 &= \bar{q}S\bar{c}c_7 \frac{\partial C_{m_0}}{\partial \alpha} \\
q_3 &= \bar{q}S\bar{c}c_7 \frac{\partial C_{m_0}}{\partial \beta} \\
q_4 &= \frac{\bar{q}S\bar{c}^2 c_7}{2V_t} C_{m_q} \\
q_5 &= c_7 h_{moteur} \\
q_6 &= \frac{\bar{q}S\bar{c}^2 c_7}{2V_t} \left(\frac{\partial C_{m_q}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{m_q}}{\partial \beta} \dot{\beta} \right) + c_5 (\dot{p}r + p\dot{r}) - 2c_6 (p\dot{p} - r\dot{r})
\end{aligned}$$

$$\begin{aligned}
r_1 &= \bar{q}Sb (c_4 C_{l_{\delta_a}} + c_9 C_{n_{\delta_a}}) \\
r_2 &= \bar{q}Sb (c_4 C_{l_{\delta_r}} + c_9 C_{n_{\delta_r}}) \\
r_3 &= \bar{q}Sb \left(c_4 \frac{\partial C_{l_0}}{\partial \alpha} + c_9 \frac{\partial C_{n_0}}{\partial \alpha} \right) \\
r_4 &= \bar{q}Sb \left(c_4 \frac{\partial C_{l_0}}{\partial \beta} + c_9 \frac{\partial C_{n_0}}{\partial \beta} \right) \\
r_5 &= \frac{\bar{q}Sb^2}{2V_t} (c_4 C_{l_p} + c_9 C_{n_p}) \\
r_6 &= -c_9 h_{moteur} \\
r_7 &= \frac{\bar{q}Sb^2}{2V_t} (c_4 C_{l_r} + c_9 C_{n_r}) \\
r_8 = & \bar{q}Sb \left\{ c_4 \left(\frac{pb}{2V_t} \left(\frac{\partial C_{l_p}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{l_p}}{\partial \beta} \dot{\beta} \right) + \frac{rb}{2V_t} \left(\frac{\partial C_{l_r}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{l_r}}{\partial \beta} \dot{\beta} \right) \right) \right. \\
& + c_9 \left(\frac{pb}{2V_t} \left(\frac{\partial C_{n_p}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{n_p}}{\partial \beta} \dot{\beta} \right) + \frac{rb}{2V_t} \left(\frac{\partial C_{n_r}}{\partial \alpha} \dot{\alpha} + \frac{\partial C_{n_r}}{\partial \beta} \dot{\beta} \right) \right) \left. \right\} \\
& - c_2 (\dot{r}q + r\dot{q}) + c_8 (\dot{p}q + p\dot{q})
\end{aligned}$$

$$\begin{aligned}
C_{n_{\delta_a}} &= \left(\frac{\partial C_{n_0}}{\partial \delta_a} + \frac{pb}{2V_t} \frac{\partial C_{n_p}}{\partial \delta_a} + \frac{rb}{2V_t} \frac{\partial C_{n_r}}{\partial \delta_a} \right) \\
C_{n_{\delta_r}} &= \left(\frac{\partial C_{n_0}}{\partial \delta_r} + \frac{pb}{2V_t} \frac{\partial C_{n_p}}{\partial \delta_r} + \frac{rb}{2V_t} \frac{\partial C_{n_r}}{\partial \delta_r} \right) \\
C_{l_{\delta_a}} &= \left(\frac{\partial C_{l_0}}{\partial \delta_a} + \frac{pb}{2V_t} \frac{\partial C_{l_p}}{\partial \delta_a} + \frac{rb}{2V_t} \frac{\partial C_{l_r}}{\partial \delta_a} \right) \\
C_{l_{\delta_r}} &= \left(\frac{\partial C_{l_0}}{\partial \delta_r} + \frac{pb}{2V_t} \frac{\partial C_{l_p}}{\partial \delta_r} + \frac{rb}{2V_t} \frac{\partial C_{l_r}}{\partial \delta_r} \right)
\end{aligned} \tag{4.47}$$

Cependant, autant les accélérations angulaires \dot{p} , \dot{q} et \dot{r} sont mesurées, autant les dérivées en angles d'attaque $\dot{\alpha}$ et de dérapage $\dot{\beta}$ ne le sont pas.

Celles-ci pourraient être obtenues par filtrage, mais cette approche ajouteraient 2 variables d'état supplémentaires au système qui compte déjà les 12 variables d'état "classiques" et les 3 variables d'état liées aux surfaces de contrôle. Elles sont donc calculées:

$$\begin{aligned}
\dot{\alpha} &= -\frac{1}{V_t \cos \beta} (-a_z \cos \alpha + a_x \sin \alpha) + q - \tan \beta (p \cos \alpha + r \sin \alpha) \\
&\quad + \frac{g}{V_t \cos \beta} (\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha) \\
\dot{\beta} &= \frac{1}{V_t} (-a_x \cos \alpha \sin \beta + a_y \cos \beta - a_z \sin \alpha \sin \beta) + p \sin \alpha - r \cos \alpha \\
&\quad + \frac{g}{V_t} (\cos \beta \sin \phi \cos \theta + \sin \beta (\cos \alpha \sin \theta - \sin \alpha \cos \phi \cos \theta))
\end{aligned} \tag{4.48}$$

En outre, la condition nécessaire et suffisante pour que l'inverse de la matrice existe est :

$$p_1 r_2 - p_2 r_1 \neq 0 \iff C_{n_{\delta_a}} C_{l_{\delta_r}} - C_{n_{\delta_r}} C_{l_{\delta_a}} \neq 0 \tag{4.49}$$

A priori rien ne permet d'assurer que ce critère est vérifié en tout temps. L'utilisation d'un pseudo-inverse permettrait de contourner ce problème, seulement afin de ne pas saturer les actionneurs, le déterminant ne devrait pas non plus être dans un voisinage de 0. La séparation de la dynamique a justement pour objectif de garantir cette condition en assurant un meilleur conditionnement de la matrice à inverser.

De plus, selon Sun et Clarke (1994), la condition 4.49 est en général satisfaite sur l'ensemble de l'enveloppe de vol.

Ainsi de nombreux calculs sont nécessaires ; ceux-ci peuvent cependant être allégés. Rappelons que le module d'identification a été conçu afin de pouvoir être adapté à une utilisation par une commande de vol reconfigurable. La commande développée ici n'a pas une telle envergure, de ce fait certaines hypothèses peuvent être faites. En effet, les données utilisées pour l'entraînement des réseaux de neurones sont telles que C_{a_p} , C_{a_q} et C_{a_r} sont fonction uniquement de l'angle d'attaque α , il est donc justifié de faire les approximations suivantes :

$$\frac{\partial C_{a_q}}{\partial \delta_e} = \frac{\partial C_{a_p}}{\partial \delta} = \frac{\partial C_{a_r}}{\partial \delta} = \frac{\partial C_{a_i}}{\partial \beta} = 0 \quad (4.50)$$

où $a = l$, m ou n , $\delta = \delta_a$ ou δ_r et $i = q$, p ou r .

Cette commande a l'avantage de fournir directement la valeur des déflections des surfaces de contrôle à appliquer. Cependant elle présente certains inconvénients :

- le temps de calcul peut être long, et il convient de vérifier si il est compatible avec une implémentation en temps réel
- la matrice à inverser pourrait être singulière pour certaines configurations de vol ou bien mal conditionnée et entraîner des efforts en commande importants
- la saturation des actionneurs n'est pas prise en compte lors du design
- si l'aéronef présente des surfaces de contrôle supplémentaires, celles-ci ne peuvent pas être commandées directement et une allocation entre les commandes classiques (δ_e , δ_a , δ_r) et l'ensemble des commandes disponibles est nécessaire

L'inversion par l'équation 4.45 n'est donc pas retenue dans le cadre de cette étude.

Lorsque les coefficients aérodynamiques sont considérés comme entrée, l'inversion est directe et on obtient à partir de l'équation I.7:

$$\begin{bmatrix} C_{l_c} \\ C_{m_c} \\ C_{n_c} \end{bmatrix} = \begin{bmatrix} \bar{q}Sbc_3 & 0 & \bar{q}Sbc_4 \\ 0 & \bar{q}S\bar{c}c_7 & 0 \\ \bar{q}Sbc_4 & 0 & \bar{q}Sbc_9 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p}_{des} \\ \dot{q}_{des} \\ \dot{r}_{des} \end{bmatrix} - \begin{bmatrix} 0 & -c_4h_{moteur} & 0 \\ 0 & 0 & c_7h_{moteur} \\ 0 & -c_9h_{moteur} & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} c_1rq + c_2pq \\ c_5pr - c_6(p^2 - r^2) \\ c_8pq - c_2rq \end{bmatrix} \right\} \quad (4.51)$$

Cette méthode présente des avantages non négligeables par rapport à la méthode précédente :

- elle nécessite peu de calculs
- l'existence de l'inverse de la matrice peut être étudiée lors de la conception, et étant invariante aucune singularité ne peut apparaître au cours du vol
- elle fait intervenir la dynamique désirée de la dérivée première des variables contrôlées (et non les dérivées secondes)
- elle ne fait pas intervenir, à ce niveau, les estimées des coefficients aérodynamiques qui sont la principale source d'incertitude du modèle

Un module d'allocation du contrôle est cependant nécessaire ; il sera explicité en 4.3. Celui-ci utilise les coefficients aérodynamiques identifiés par le module d'identification développé au chapitre 3, incorpore la dynamique des actionneurs et permet d'ailleurs de tenir compte de leurs contraintes de saturation. Il résout ensuite un problème d'optimisation : celui d'obtenir la valeur des déflections des surfaces de contrôle donnant les valeurs commandées des moments aérodynamiques.

Cette méthode est donc retenue.

Conception de la boucle externe

Les équations non linéaires qui régissent l'angle d'attaque α , l'angle de dérapage β et l'assiette de roulis ϕ sont données en annexe I et rappelées ici:

$$\begin{aligned}
 \dot{\alpha} &= -\frac{\bar{q}S}{mV_t \cos \beta} (-C_z \cos \alpha + C_x \sin \alpha) + q - \tan \beta (p \cos \alpha + r \sin \alpha) \\
 &\quad + \frac{g}{V_t \cos \beta} (\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha) - \frac{T \sin \alpha}{mV_t \cos \beta} \\
 \dot{\beta} &= \frac{\bar{q}S}{mV_t} (-C_x \cos \alpha \sin \beta + C_y \cos \beta - C_z \sin \alpha \sin \beta) + p \sin \alpha - r \cos \alpha \\
 &\quad + \frac{g}{V_t} \cos \beta \sin \phi \cos \theta + \frac{\sin \beta}{V_t} \left(g \cos \alpha \sin \theta - g \sin \alpha \cos \phi \cos \theta - \frac{T \cos \alpha}{m} \right) \\
 \dot{\phi} &= p + \tan \theta (q \sin \phi + r \cos \phi)
 \end{aligned} \tag{4.52}$$

L'inversion des équations 4.52 en y incorporant la dynamique désirée, en remplaçant les vitesses angulaires par leur commande et en exploitant la linéarité des coefficients aérodynamiques par rapport aux vitesses angulaires donne :

$$\begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix} = \begin{bmatrix} -\tan \beta \cos \alpha & 1 + a_1 & -\tan \beta \sin \alpha \\ \sin \alpha + b_1 & b_2 & -\cos \alpha + b_3 \\ 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{\alpha}_{des} \\ \dot{\beta}_{des} \\ \dot{\phi}_{des} \end{bmatrix} - \begin{bmatrix} a_2 + \frac{g}{V_t \cos \beta} (\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha) \\ b_4 + \frac{g}{V_t} \cos \beta \sin \phi \cos \theta + \frac{\sin \beta}{V_t} (g \cos \alpha \sin \theta - g \sin \alpha \cos \phi \cos \theta) \\ 0 \end{bmatrix} \right\} \tag{4.53}$$

où

$$\begin{aligned}
 a_1 &= \frac{\bar{q}S}{2mV_t^2 \cos \beta} (C_{zq} \cos \alpha - C_{xq} \sin \alpha) \\
 a_2 &= \frac{\bar{q}S}{mV_t \cos \beta} \left(C_{z0} \cos \alpha - (C_{x0} + \frac{T}{\bar{q}S}) \sin \alpha \right)
 \end{aligned} \tag{4.54}$$

$$\begin{aligned}
b_1 &= \frac{\bar{q}Sb}{2mV_t^2} C_{y_p} \cos \beta \\
b_2 &= \frac{\bar{q}S\bar{c}}{2mV_t^2} (-C_{x_q} \cos \alpha \sin \beta - C_{z_q} \sin \alpha \sin \beta) \\
b_3 &= \frac{\bar{q}Sb}{2mV_t^2} C_{y_r} \cos \beta \\
b_4 &= \frac{\bar{q}S}{mV_t} \left(-\left(C_{x_0} + \frac{T}{\bar{q}S} \right) \cos \alpha \sin \beta + C_{y_0} \cos \beta - C_{z_0} \sin \alpha \sin \beta \right)
\end{aligned} \tag{4.55}$$

De même que dans la boucle interne, on émet l'hypothèse que l'inverse de la matrice existe.

D'autre part plutôt que d'estimer T , l'identification du coefficient C_x par le module d'identification à réseaux de neurones permet d'exprimer T en fonction de celui-ci et de la mesure de l'accélération a_x :

$$\frac{T + \bar{q}SC_{x_0}}{m} = a_x - \frac{\bar{q}S}{m} \frac{q\bar{c}}{2V_t} C_{x_q} \tag{4.56}$$

Les qualités de manœuvrabilité suggèrent de choisir une dynamique désirée du deuxième ordre pour les angles d'attaque et de dérapage, et du premier ordre pour l'assiette de roulis :

$$\begin{aligned}
\dot{\alpha}_{des} &= 2\zeta_\alpha \omega_\alpha (\alpha_c - \alpha) + \omega_\alpha^2 \int (\alpha_c - \alpha) dt \\
\dot{\beta}_{des} &= 2\zeta_\beta \omega_\beta (\beta_c - \beta) + \omega_\beta^2 \int (\beta_c - \beta) dt \\
\dot{\phi}_{des} &= \omega_\phi (\phi_c - \phi)
\end{aligned} \tag{4.57}$$

où $(\omega_\alpha, \zeta_\alpha)$ sont conformes aux spécifications du mode rapide avec $\omega_\alpha < \omega_q$, $(\omega_\beta, \zeta_\beta)$ à celles du roulis hollandais avec $\omega_\beta < \omega_r$, et ω_ϕ à celles du mode spiral.

Le contrôle par inversion dynamique de la boucle externe est ainsi moins complexe que celle de la boucle interne.

4.2.9 Conclusion sur l'inversion dynamique

La commande par inversion dynamique est une méthode relativement flexible, qui laisse un certain nombre de choix au niveau de la conception.

Afin d'éviter les singularités liées à l'inversion de la fonction g , le choix de la séparation temporelle a été fait puis le choix d'utiliser les coefficients aérodynamiques en moment comme pseudo-contrôle. La dynamique désirée quant à elle a été établie comme fonction des qualités de manoeuvrabilité. L'utilisation de la structure de type proportionnel intégral permet de minimiser l'impact de l'incertitude du modèle de la dynamique, mais la robustesse du contrôleur reste à vérifier.

Le choix d'utiliser une allocation du contrôle, plutôt que d'intégrer directement la dynamique des actionneurs et leur saturation en amplitude et en taux de rotation dans la conception du contrôleur par inversion dynamique, permet de conserver un maximum de flexibilité, que ce soit par rapport aux nombre de surfaces de contrôle d'un avion ou encore par rapport aux variables dont dépendent les coefficients aérodynamiques.

4.3 Allocation du contrôle

Sur un aéronef classique, les moments de rotation autour de ses trois axes sont obtenus grâce à trois surfaces de contrôle : l'élévateur, les ailerons et le gouvernail. Cependant ces surfaces ne créent pas des moments découplés. Traditionnellement un découplage est effectué mécaniquement grâce à l'utilisation de “aileron-rudder interconnects”.

Toutefois les aéronefs à grande manœuvrabilité deviennent sophistiqués avec des surfaces de contrôle non conventionnelles qui exploitent ces couplages. Dans ce cas, une allocation du contrôle est nécessaire. Celui-ci a pour objectif de distribuer le contrôle sur les différentes surfaces de contrôle. Le F-16, par exemple, possède une surface de contrôle supplémentaire : le bord d'attaque. Bien que dans sa conception originelle, cette surface n'était pas accessible directement, l'utilisation d'une allocation du contrôle permet de la commander.

De plus, l'étude du cas longitudinal simplifié (voir 4.2.7) a mis en évidence l'impact de la saturation des actionneurs sur l'efficacité de la commande. Les contraintes en amplitude et en taux de rotation de chaque surface de contrôle δ_i sont prises en compte dans l'allocation du contrôle :

$$\underline{\delta}_i \leq \delta_i \leq \overline{\delta}_i \text{ et } \underline{\dot{\delta}}_i \leq \dot{\delta}_i \leq \overline{\dot{\delta}}_i \quad (4.58)$$

Après avoir traduit ces contraintes en temps discret, une allocation statique qui néglige la dynamique des actionneurs, puis une allocation dynamique qui prend en compte cette dynamique sont étudiées.

4.3.1 Contraintes en temps discret

Comme les systèmes d'allocation du contrôle sont en temps discret, les contraintes en taux de rotation de l'équation 4.58 peuvent être transposées en une contrainte en amplitude grâce à une approximation du premier ordre :

$$\dot{\delta}_i \approx \frac{\delta_i(t) - \delta_i(t - T)}{T} \quad (4.59)$$

où T est la période d'échantillonage.

En combinant les contraintes en amplitudes de l'équation 4.58 et la transposition de celles en taux de rotation, il vient (Luo et al., 2004):

$$\begin{aligned}\delta_{i_{min}} &\leq \delta_i \leq \delta_{i_{max}} \\ \delta_{i_{min}} &= \max \left\{ \underline{\delta}_i, \delta_i(t - T) + \dot{\underline{\delta}}_i T \right\} \\ \delta_{i_{max}} &= \max \left\{ \overline{\delta}_i, \delta_i(t - T) + \dot{\overline{\delta}}_i T \right\}\end{aligned}\quad (4.60)$$

Ainsi, à chaque itération les contraintes doivent être mises à jour.

4.3.2 Allocation statique

Il est usuel d'utiliser pour l'allocation du contrôle une modélisation linéaire et invariante qui néglige la dynamique des actionneurs :

$$M = B\delta \quad (4.61)$$

où M est le vecteur des coefficients aérodynamiques en moment, B une matrice appelée “control effectiveness matrix” et δ est le vecteur des déflections des surfaces de contrôle.

De nombreuses méthodes existent afin de calculer le vecteur δ donnant les coefficients commandés M_c , telles que la programmation linéaire, la programmation quadratique, le pseudo-inverse, le pseudo-inverse cascadé ou encore le “daisy chaining”. La méthode du pseudo-inverse cascadé est présentée ici (Luo et al., 2004).

L'idée est de minimiser la fonction J par rapport à δ (voir équation 4.62) sous la contrainte $M_c = B\delta$.

$$J = \frac{1}{2} (\delta - \delta_{des})^T W_\delta (\delta - \delta_{des}) \quad (4.62)$$

où W_δ est une matrice de pondération et δ_{des} un contrôle désiré.

Si le vecteur de contrôle n'avait pas de limites, la solution serait :

$$\delta = \delta_{des} + W_{\delta}^{-1} B^T (BW_{\delta}^{-1} B^T)^{-1} (M_c - B\delta_{des}) \quad (4.63)$$

Le principe de la cascade est alors le suivant. Si aucune composante du vecteur de contrôle δ n'atteint ses limites, alors celui-ci est trouvé et les coefficients aérodynamiques en moment commandés M_c sont atteints. Si une composante du vecteur de contrôle δ atteint la saturation, alors celle-ci est fixée à cette valeur, sa contribution est ôtée de la formulation du problème et celui-ci est calculé de nouveau. Si toutes les composantes du vecteur contrôle atteignent leur valeur de saturation, alors les coefficients aérodynamiques en moments commandés M_c ne sont pas atteints.

Le module d'identification utilisé dans cette étude fournit des coefficients aérodynamiques comme fonction non linéaire de certains états de l'avion et des surfaces de contrôle. Deux solutions sont alors offertes : la première est d'adapter la méthodologie précédente à un modèle non linéaire et la seconde est de linéariser les coefficients.

L'adaptation à un modèle non linéaire est relativement directe grâce à la fonction *fmincon* de MatLab[©]. Cependant il conviendrait de vérifier la compatibilité du temps de calcul avec une implémentation en temps réel.

Une linéarisation invariante des coefficients aérodynamiques sur l'ensemble de vol n'est pas envisageable du fait de la grande non linéarité des coefficients à angles d'attaque élevés. Cependant celle-ci peut être effectuée à chaque itération et donner lieu à une allocation dynamique.

4.3.3 Allocation dynamique

L'approximation suivante permet de linéariser les coefficients à chaque instant :

$$C_a \approx C_{a_0} + \sum_{i=1}^m \frac{\partial C_a}{\partial \delta_i} \Delta \delta_i \quad (4.64)$$

où $\Delta \delta_i = \delta_i - \delta_{i_0}$.

Elle peut alors être traduite par la relation 4.65 qui est analogue à la formulation du problème discuté en début du paragraphe 4.3.2, à la différence près que la relation entre les coefficients aérodynamiques C et les déflexions des surfaces de contrôle δ est variante dans le temps :

$$\Delta C(t) = B(t) \Delta \delta(t) \quad (4.65)$$

La fonction suivante doit alors être minimisée à chaque instant t sous la contrainte $\Delta C_c(t) = B(t) \Delta \delta(t)$:

$$J = \frac{1}{2} \Delta \delta(t)^T W_\delta(t) \Delta \delta(t) \quad (4.66)$$

où $W_\delta(t)$ est une matrice de pondération et $\Delta C_c(t)$ la variation des coefficients aérodynamiques désirée.

Un choix initial pour W_δ peut être une matrice diagonale invariante dont chaque élément i , associé à une surface de contrôle δ_i , est défini par :

$$W_\delta(i, i) = \frac{1}{\max(\delta_i)^2} \quad (4.67)$$

En l'absence de contraintes, la solution est alors :

$$\Delta \delta(t) = W_\delta^{-1} (B(t)^T W_\delta^{-1} B(t))^{-1} \Delta C_c(t) \quad (4.68)$$

Différents auteurs ((Luo et al., 2004), (Keviczky & Balas, 2006) et (Kale & Chipperfield, 2002)) étayent les avantages d'utiliser des modèles linéaires variant dans le temps plutôt que des modèles non linéaires, notamment en ce qui concerne la réduction du temps de calcul et la compatibilité avec une application temps réel.

Luo et al. (2004) ont mis en évidence les avantages d'utiliser une allocation par commande prédictive par rapport à d'autres stratégies d'allocation dans le cas où la dynamique des actionneurs est non négligeable par rapport à celle du système. Cependant dans le cas où les constantes de temps des actionneurs sont petites, l'allocation prédictive nécessiterait un temps d'échantillonnage qui peut ne pas être compatible avec le temps de calcul de la commande. Dans ce cas l'allocation précédente est recommandée.

La modélisation des coefficients aérodynamiques est de forme identique à celle de l'équation 4.64. La dynamique des actionneurs est discrétisée au point de fonctionnement à chaque instant d'échantillonage nT_{ech} . Le problème de l'allocation est alors formulé à chaque instant nT_{ech} de la manière suivante :

$$\begin{aligned}\Delta\delta(k+1) &= A_\delta\Delta\delta(k) + B_\delta\Delta\delta_c(k) \\ \Delta\delta_c(k) &= \Delta\delta_c(k-1) + \Delta r(k) \\ \begin{bmatrix} \Delta C(k) \\ \Delta\delta(k) \end{bmatrix} &= \begin{bmatrix} B(nT_{ech}) \\ I \end{bmatrix} \Delta\delta(k)\end{aligned}\tag{4.69}$$

où $\Delta\delta(k=0) = 0$.

Les contraintes sur $\Delta\delta$ sont similaires à celles de l'équation 4.60 :

$$\begin{aligned}\Delta\delta_{i_{min}}(k) &\leq \Delta\delta_i(k) \leq \Delta\delta_{i_{max}}(k) \\ \Delta\delta_{i_{min}}(k) &= \max \left\{ \underline{\delta}_i - (\Delta\delta_i(k-1) + \delta_i((n-1)T_{ech})), \underline{\dot{\delta}}_i T_{ech} \right\} \\ \Delta\delta_{i_{max}}(k) &= \min \left\{ \overline{\delta}_i - (\Delta\delta_i(k-1) + \delta_i((n-1)T_{ech})), \overline{\dot{\delta}}_i T_{ech} \right\}\end{aligned}\tag{4.70}$$

Le problème peut alors être transformé en un problème de programmation quadratique comme indiqué en annexe V avec les homologies suivantes :

$$\begin{aligned} y &= \Delta C \\ z &= \emptyset \\ u &= \Delta \delta \end{aligned} \tag{4.71}$$

Les horizons de contrôle H_c et de prédition H_p , la période d'échantillonage T_{ech} et les matrices de pondération, Q et R , restent à choisir en fonction de l'application.

Comme la trajectoire de référence des coefficients aérodynamiques n'est pas connue, il est suggéré de prendre une prédition dite "one step ahead" avec $H_p = H_c = 1$. La période d'échantillonage T_{ech} doit être plus petite que la constante de temps des actionneurs. Un choix initial pour la matrice de pondération R est identique à la matrice de pondération W_δ du cas d'allocation continue précédent. La matrice de pondération Q peut également être choisie diagonale, avec ses éléments liés aux valeurs maximales des grandeurs mesurées associées.

4.3.4 Conclusion sur l'allocation du contrôle

Le choix de l'allocation du contrôle dépend essentiellement de la validité de l'hypothèse selon laquelle la dynamique des actionneurs peut être négligée.

Généralement celle-ci est valide pour les avions de transport et les avions militaires, dans ce cas une allocation continue, analogue à l'allocation statique, est suffisante. Les UAV ("Unmanned Air Vehicle") quant à eux peuvent avoir des actionneurs dont la dynamique n'est pas négligeable par rapport à celle de l'aéronef, dans ce cas l'utilisation de l'allocation prédictive est préférable.

4.4 Les contraintes sur les états et la commande prédictive

Dans le cas de manœuvres à grande amplitude, l'utilisation d'une dynamique désirée de type proportionnel + intégral pourrait générer des commandes en angles d'attaque et de dérapage important. Les coefficients aérodynamiques ont été identifiés pour des plages d'angles d'attaque et dérapage limitées. Il n'est donc pas souhaitable de les dépasser, car au delà la performance du module d'identification n'est plus garantie.

Cette contrainte pourrait être intégrée au design en remplaçant la dynamique désirée de type proportionnel intégral sur ces états par une commande prédictive (voir annexe V). Celle-ci permet de plus d'anticiper les changements survenants sur la trajectoire de référence $(\alpha_{ref}, \beta_{ref}, \phi_{ref})$ en fournissant une commande $(\tilde{\alpha}_c, \tilde{\beta}_c, \tilde{\phi}_c)$ qui minimise un critère quadratique.

Sous l'hypothèse que le retour de sortie effectue une linéarisation exacte, la dynamique en (α, β, ϕ) de la boucle fermée par inversion dynamique a la forme suivante

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\alpha}_c \\ \tilde{\beta}_c \\ \tilde{\phi}_c \end{bmatrix} \quad (4.72)$$

Ce modèle présente l'avantage d'être linéaire et invariant. Il est donc bien adapté à la commande prédictive.

Les sorties et les contraintes suivantes sont alors choisies:

$$\begin{aligned} y &= [\alpha, \beta, \phi]^T \\ -10^\circ &\leq \alpha \leq 45^\circ \\ -30^\circ &\leq \beta \leq 30^\circ \end{aligned} \quad (4.73)$$

Le problème est ainsi mis en forme conformément à la méthodologie de l'annexe V et peut être résolu par la programmation quadratique associée. Il convient pour cela de fixer les horizons de prédiction et de contrôle H_p et H_c , le temps d'échantillonnage T_{ech} et les matrices de pondération Q et R .

Le système linéaire résultant de l'inversion dynamique est découplé. Une stratégie hybride peut alors être envisagée. En effet :

- Pour la coordination, la référence en angle de dérapage β_{ref} doit être prise nulle. Il est alors légitime de conserver la dynamique désirée de type proportionnel+integral et de fixer $\beta_c = 0$. L'intégrateur de la boucle d'inversion dynamique permet de rejeter les perturbations constantes pouvant survenir.
- La commande prédictive peut être appliquée sur les deux variables restantes.

Soulignons toutefois que ni la robustesse ni la stabilité de la commande prédictive appliquée à un système linéarisé par retour de sortie n'ont été étudiées dans le cadre de ce mémoire, et que ces deux points restent à démontrer. De ce fait, elle ne sera abordée que succinctement dans les simulations.

4.5 Une approche pour la robustesse

Il a été souligné en 4.2.1 que l'inversion dynamique nécessite la connaissance parfaite du système à l'étude. Toutefois les coefficients aérodynamiques ne sont pas connus de manière précise. La robustesse de ce contrôleur est donc un point critique à analyser. Cependant une telle étude pourrait faire l'objet d'un mémoire à part entière, celle-ci est donc seulement ébauchée.

Une boucle externe de contrôle peut être utilisée afin de garantir la robustesse de la

boucle fermée en présence d'erreurs de modélisation. Le plus courant est d'utiliser la μ -analyse (Ito et al, 2002), bien que d'autres méthodes de contrôle robuste ne sont pas à exclure. En effet Escande (1997) propose d'utiliser une boucle de rétro-action LQR effectuée sur un nombre réduit de variables d'état.

Une solution plus simple a été adoptée ici. En effet, le module d'identification à base de réseaux de neurones identifie, hors ligne, les coefficients aérodynamiques à 2% près pour les coefficients de moments et 1% près pour ceux de forces. Ainsi en l'absence d'apprentissage et si aucune modification majeure ne survient aux coefficients (suite à un dégât par exemple), le terme intégral des dynamiques désirées permet d'assurer une erreur statique nulle en présence de ces erreurs de modélisation et/ou de perturbations constantes.

Lorsque l'apprentissage en ligne est programmé, la loi d'adaptation des poids se doit d'être robuste. Différentes lois sont proposées par Farell et Polycarpou (2006). Parmi elles, on trouve la ϵ -modification, la σ -modification, l'utilisation de zones mortes et la projection. L'apprentissage en ligne dépassant le cadre de cette étude, le lecteur est invité à se reporter à la référence.

4.6 Conclusion sur la commande de vol

La structure du contrôleur a été mise en évidence dans ce chapitre et est représentée par la figure 4.11.

L'inversion dynamique dans les boucles interne et externe permet d'assurer un comportement du système conforme aux qualités de manœuvrabilité. Bien que le retour linéarisant nécessite un modèle parfait, l'utilisation d'une commande désirée de type proportionnel intégral permet d'annuler l'effet de petites erreurs de modélisation ou de perturbations constantes. Le fait d'utiliser le module d'identification par

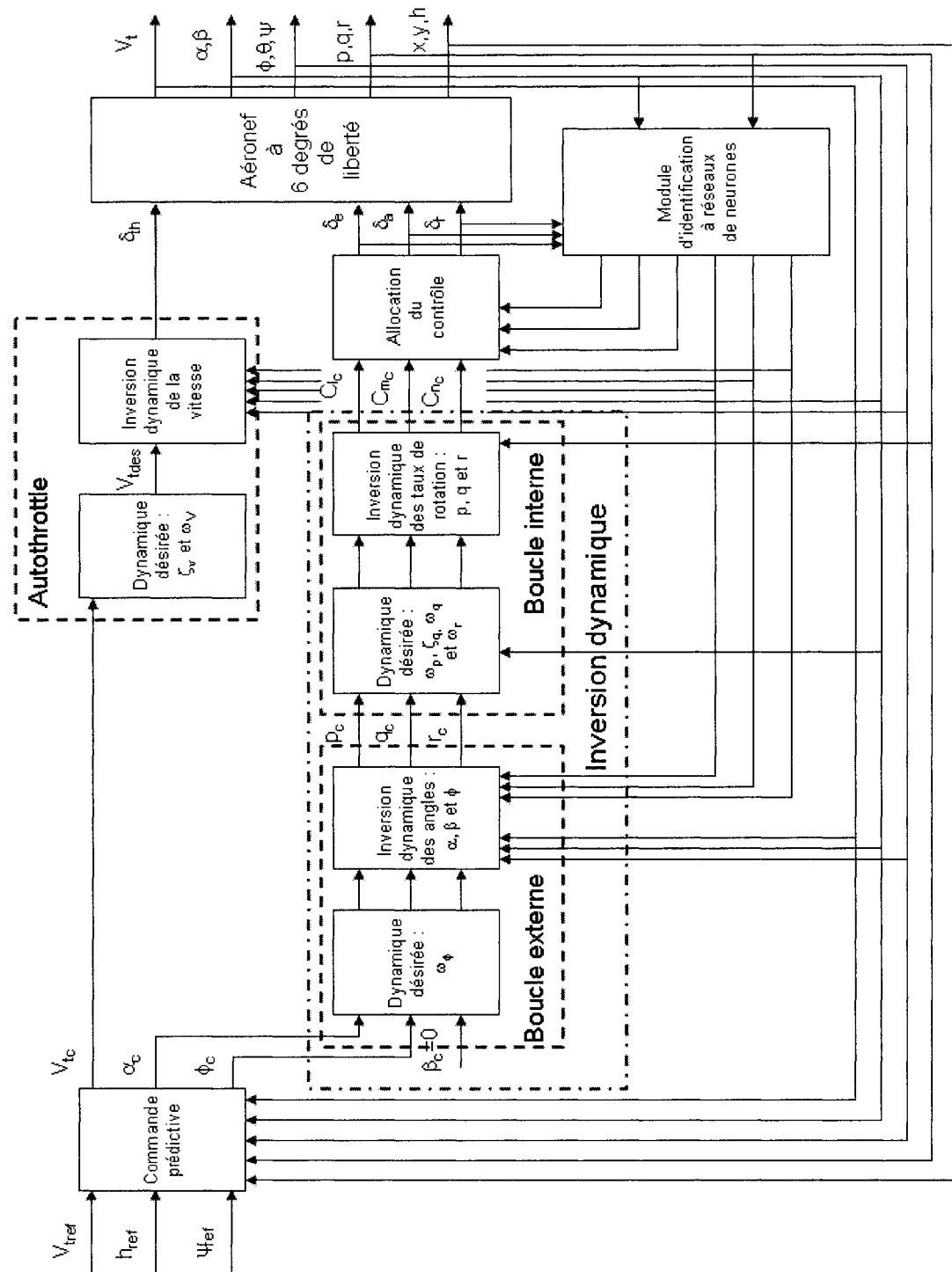


Figure 4.11 Architecture du contrôleur

réseaux de neurones développé précédemment permet notamment de garantir que, en condition de vol normale, les erreurs de modélisation sont limitées.

Afin de ne pas rencontrer de singularité au sein de l'inversion dynamique, les coefficients aérodynamiques en moment ont été choisis comme pseudo-contrôle. Une allocation du contrôle permet ensuite de transformer ces commandes en des commandes sur les actionneurs. Cette allocation peut être statique ou dynamique. L'allocation dynamique n'a pas de restriction d'application, tandis que l'allocation statique ne doit pas être utilisée si la dynamique des actionneurs n'est pas négligeable devant celle du système. Le module d'identification par réseau de neurone fournit directement les dérivées nécessaires au calcul de l'allocation.

Une commande prédictive permet d'anticiper les changements de consigne de la trajectoire planifiée ($\alpha_{ref}, \beta_{ref}, \phi_{ref}$). De plus, elle incorpore directement les limites en angle d'attaque. Toutefois la robustesse de cette commande, ainsi que celle de l'ensemble du contrôleur, restent matière à une recherche future.

CHAPITRE 5

RÉSULTATS DE LA SIMULATION NON LINÉAIRE

Le simulateur non linéaire développé pour la validation est explicité dans l'annexe VI. Les résultats de la simulation en boucle ouverte sont d'abord présentés. Le contrôleur est ensuite adapté à la configuration du F-16. Les simulations en boucle fermée, avec le choix des gains, sont finalement répertoriées.

5.1 Simulation en boucle ouverte

La position du centre de gravité est prise en avant de sa position de référence, de sorte que la dynamique en boucle ouverte du F-16 soit stable.

Deux manœuvres différentes sont simulées, à savoir un vol rectiligne à ailes horizontales et un virage coordonné. Pour chacune d'elles, le comportement de l'avion en boucle ouverte est analysé, et l'identification des coefficients aérodynamiques est mise en évidence.

5.1.1 Manœuvre : vol à ailes horizontales

La manœuvre considérée est une trajectoire rectiligne à vitesse et altitude fixées : $V_t = 150m.s^{-1}$ et $h = 5000m$,

En premier lieu, les variables de commande et d'état trimmés sont calculés grâce à la fonction *trim* de MatLab: $u_0 = [0,34 \ -1,8 \ 0 \ 0]$ et $x_0 = [150 \ 0,0147 \ 0 \ 0 \ 0,0147 \ 0 \ 0 \ 0 \ 0 \ 0 \ 5000 \ -1,8 \ 1,38 \ 22,28 \ 0 \ 0 \ 0,84]$. Les arrondis entraînent que ces vecteurs ne sont pas strictement identiques à la configuration de l'avion en régime permanent,

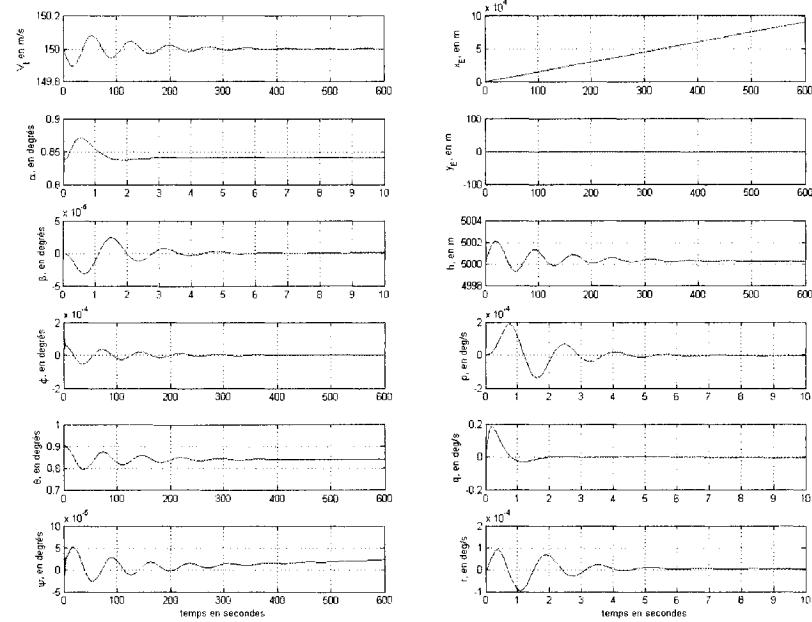
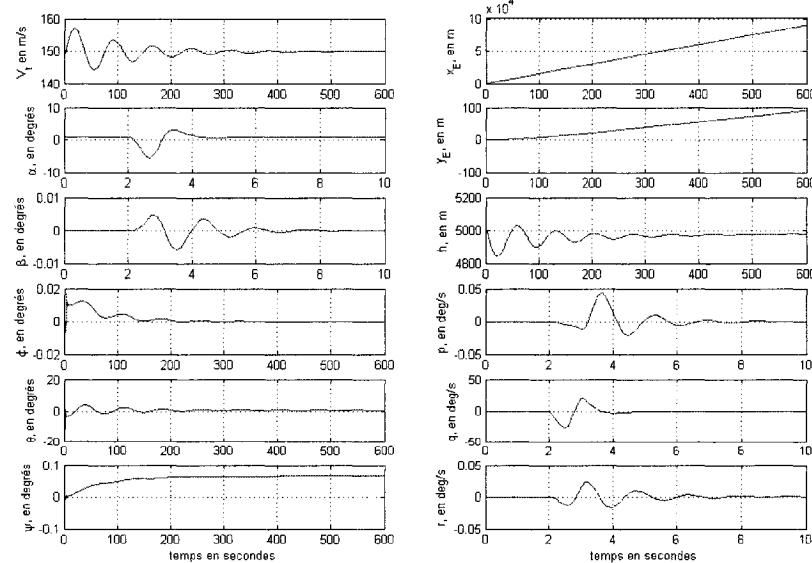


Figure 5.1 Vol à ailes plates - évolution des variables d'état

Figure 5.2 Évolution des variables d'état suite à une perturbation sur δ_e

d'où l'évolution des états illustrée par la figure 5.1. Celle-ci correspond bien à une trajectoire rectiligne à altitude constante.

On observe une oscillation lente de la vitesse totale de l'avion V_t et de son altitude h autour des valeurs de la condition de vol. La vitesse V_t oscille à la pulsation $\omega_V = 0,08 \text{ rad.s}^{-1}$ et a un dépassement maximal de 0,055%. L'altitude h oscille à la pulsation $\omega_h = 0,08 \text{ rad.s}^{-1}$ et a un dépassement maximal de 0,042%.

Ceci est caractéristique du mode phugoïde.

L'angle de dérapage est nul, tandis que l'angle d'attaque vaut $\alpha_0 = 0,0147 \text{ rad} = 0,84^\circ$ afin de créer une force aérodynamique qui maintienne le F-16 en équilibre. L'angle de tangage θ oscille de manière analogue à la vitesse, et se stabilise à $\theta_0 = \alpha_0 = 0,84^\circ$. La pente ("path angle") $\gamma = \theta - \alpha$ est ainsi nul. Les deux autres angles d'Euler, de même que les taux de rotations, sont nuls.

Afin de vérifier la stabilité du système dans cette position d'équilibre, un doublet d'amplitude 5° et d'une durée de $1s$ est appliquée à $t = 2s$ sur l'élévateur. L'évolution des variables d'état est illustrée par la figure 5.2.

Le dépassement maximal en vitesse V_t augmente à 4,59%, mais le temps d'amortissement reste du même ordre de grandeur. De manière analogue le dépassement maximal en altitude h est de 3,13% et ses oscillations sont amorties en près de 300 secondes.

L'angle d'attaque α et le taux de tangage q reviennent à leur point d'équilibre en $2s$; l'angle de dérapage β et les taux de roulis p et de lacet r y reviennent en $6s$. L'angle de roulis ϕ revient à sa valeur d'équilibre en $200s$. L'angle de lacet ψ converge vers une nouvelle valeur $\psi = 0,8^\circ$ en $200s$, ce qui implique une modification de la trajectoire de l'avion. Celle-ci est illustrée par la figure 5.3, la trajectoire y est également projetée sur le plan $x0y$ pour une meilleure visibilité.

Le couplage entre les modes latéraux et longitudinaux est mis en exergue : une perturbation sur l'élévateur entraîne une perturbation non seulement sur la vitesse de tangage q , mais aussi sur les vitesses de roulis p et de lacet r .

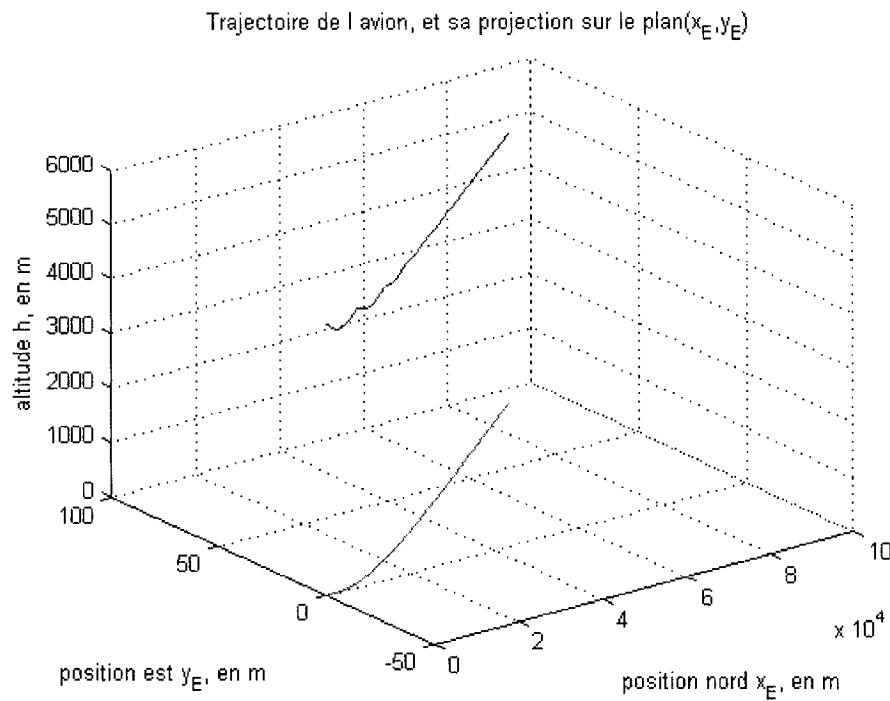


Figure 5.3 Trajectoire de l'avion suite à une perturbation sur δ_e

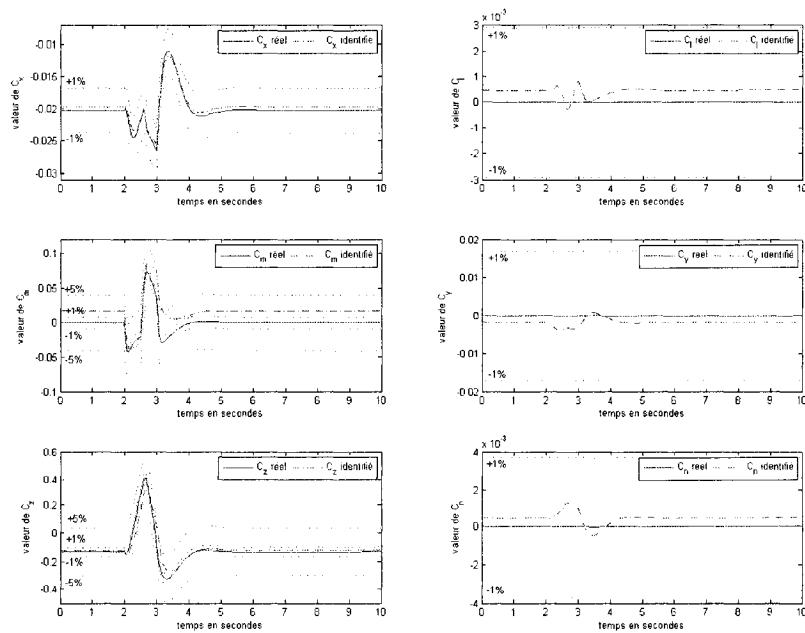


Figure 5.4 Vol à ailes plates - coefficients aérodynamiques

La figure 5.4 met en évidence l'identification des coefficients aérodynamiques par le module d'identification développé au chapitre 3. L'identification est globalement satisfaisante : les valeurs identifiées présentent, à l'exception du coefficient C_m , une erreur inférieure à 1%.

5.1.2 Mancœuvre : virage coordonné à $\dot{\psi} = 0,5^\circ.s^{-1}$

Pour un virage à $0,5^\circ.s^{-1}$ à $V_t = 150m.s^{-1}$ et $h = 5000m$, les variables d'état et de contrôle trimmées sont : $u_0 = [0,34 \ -1,8 \ -0,03 \ 0,19]$ et $x_0 = [150 \ 0,015 \ 0,002 \ 0,142 \ 0,015 \ 0 \ -0,0001 \ 0,001 \ 0,009 \ 0 \ 0 \ 5000 \ -1,8 \ 1,39 \ 22,30 \ 0,20 \ -0,02 \ 0,86]$.

L'évolution des variables d'état est illustrée par la figure 5.5.

Les oscillations en vitesse V_t et en altitude h , caractéristiques du mode phugoïde, sont visibles et identiques à celles observées pour le vol rectiligne à ailes horizontales.

L'angle d'attaque α a une valeur finale non nulle afin de créer les forces aérodynamiques nécessaires au maintien en équilibre de l'aéronef. L'angle de dérapage β est nul, car le virage est coordonné.

Afin de vérifier la stabilité du système dans cette position d'équilibre, un doublet de 5° d'une durée de 1s est appliqué aux ailerons à $t = 2s$, et un autre de mêmes caractéristiques à $t = 2,5s$ au gouvernail.

L'évolution des variables d'état est illustrée par la figure 5.6.

La trajectoire n'est quasiment pas modifiée par cette perturbation (voir figure 5.7).

De même que pour la manœuvre précédente, l'identification des coefficients aérodynamiques est présenté à la figure 5.8. Les coefficients de force sont identifiés à 1% près, tandis que les coefficients de moment ne le sont qu'à 2% près.

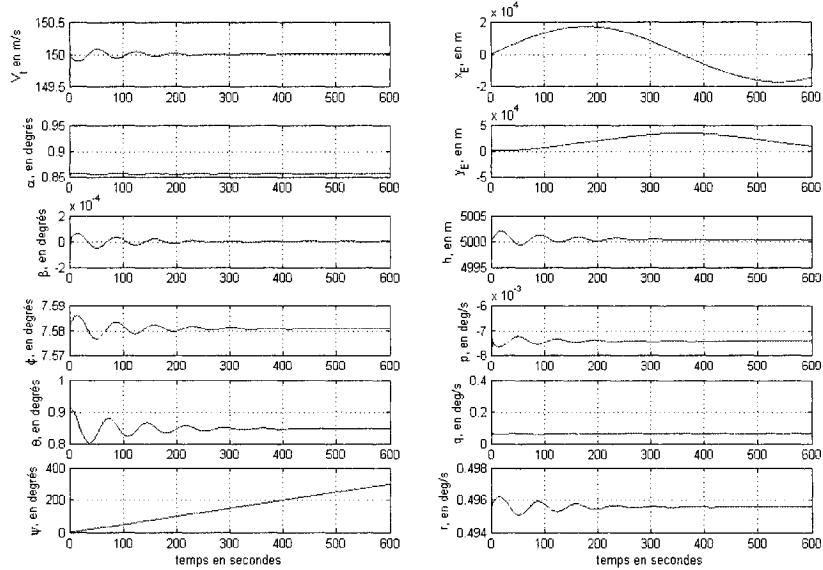
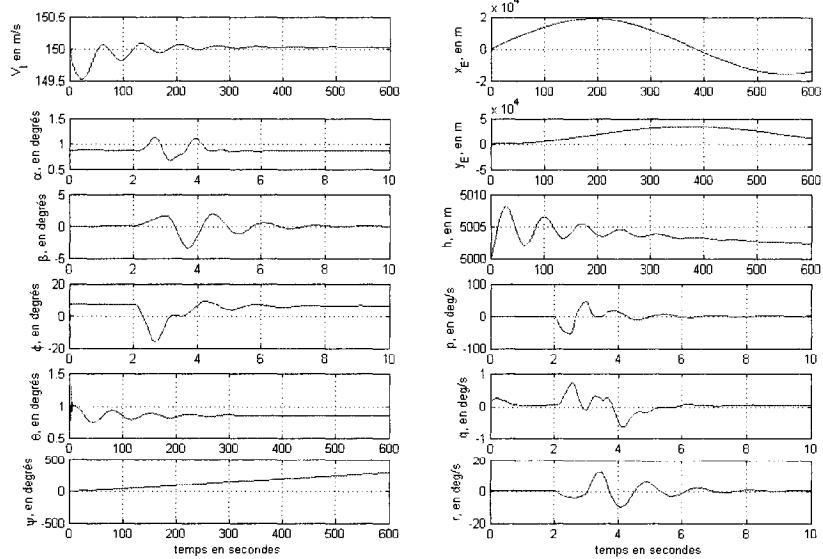


Figure 5.5 Virage - évolution des états

Figure 5.6 Évolution des états suite à une perturbation sur δ_a et δ_r

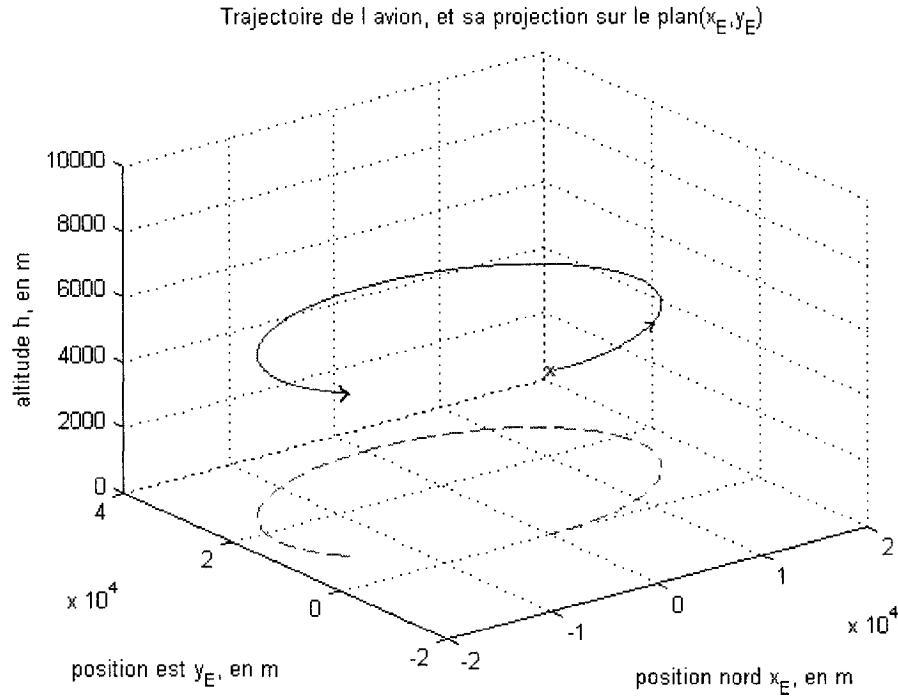


Figure 5.7 Trajectoire de l'avion suite à une perturbation sur δ_a et δ_r

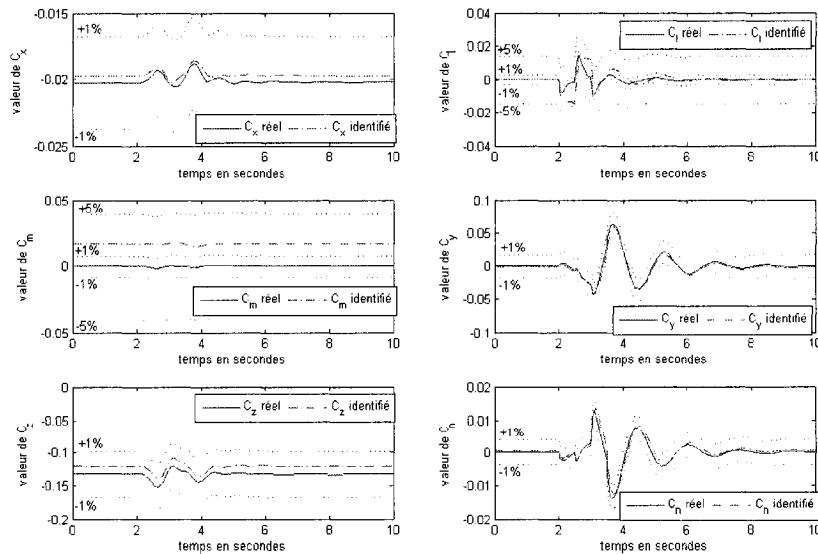


Figure 5.8 Virage - coefficients aérodynamiques

5.2 Application de la structure du contrôleur au F-16

La structure du contrôleur développé au chapitre IV est appliquée au F-16. La commande en vitesse est à préciser, de même que le choix des gains de l'inversion dynamique et celui des paramètres de l'allocation du contrôle.

5.2.1 La commande en vitesse

La modélisation du moteur du F-16 est donnée en annexe VI. L'autothrottle peut ici se décomposer en quatre étapes, comme indiqué sur la figure 5.9.

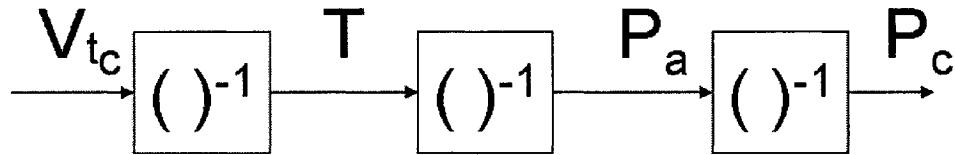


Figure 5.9 Décomposition de l'autothrottle

La première étape est l'inversion de la commande en vitesse pour obtenir la poussée, telle que décrite en 4.2.6. L'amortissement ζ_V et la pulsation propre ω_V doivent être fixés conformément aux qualités de manœuvrabilité.

Pour la dynamique de la vitesse V_t , (ω_V, ζ_V) sont caractéristiques du mode phugophile. On désire fixer l'amortissement de ce mode à $\zeta_{ph} = 0,7$, d'où $\zeta_V = \zeta_{ph} = 0,7$. On choisit aussi $\omega_V = 1$.

La seconde étape consiste à trouver la valeur de la puissance du moteur P_a en fonction de la poussée T . Cette étape est détaillée ci après.

La troisième étape consiste à calculer la valeur de la puissance commandée P_c en fonction de la puissance actuelle P_a . Le modèle de la dynamique du moteur étant complexe, l'hypothèse suivante est faite : $P_c = P_a$. La validité d'une telle hypothèse

Tableau 5.1 Caractéristiques des réseaux de neurones

Poussée identifiée	T_{idle}	T_{mil}	T_{max}
nb de neurones	9	2	2
erreur d'entraînement	$9,65 \cdot 10^{-4}$	$4,10 \cdot 10^{-4}$	$3,18 \cdot 10^{-4}$

est cependant à vérifier par la simulation, car le temps de réponse du moteur n'est pas négligeable en général. Si nécessaire, on pourra considérer pour les simulations que la poussée est commandée directement.

La puissance du moteur P_a peut être calculée (équation 5.1) comme une fonction de la valeur de la poussée T totale de l'avion fournie par l'étape précédente et des poussées T_{idle} , T_{mil} et T_{max} . Celles-ci sont fonction de la vitesse V_t et de l'altitude h de l'avion, et sont identifiées par des réseaux de neurones dont les caractéristiques sont récapitulées dans le tableau 5.1.

$$P_a = \begin{cases} 50 \frac{T - T_{idle}}{T_{mil} - T_{idle}} & \text{si } T < T_{mil} \\ 50 \frac{T - T_{mil}}{T_{max} - T_{mil}} + 50 & \text{sinon} \end{cases} \quad (5.1)$$

La figure 5.10 illustre la performance de l'autothrottle développé sous les hypothèses précédentes, c'est-à-dire dont la puissance P_c est la commande, suite à un changement de consigne de la vitesse. L'instabilité de la réponse en vitesse est dûe à l'hypothèse faite, selon laquelle la constante de temps du moteur est négligeable. On choisit alors pour les simulations de commander directement la poussée induite par le moteur. La figure 5.11 illustre la réponse en vitesse suite au même changement de consigne que précédemment. Celle-ci est conforme aux attentes, le dépassement et le temps de réponse ayant été fixés par le choix de l'amortissement ζ_V et de la pulsation propre ω_V .

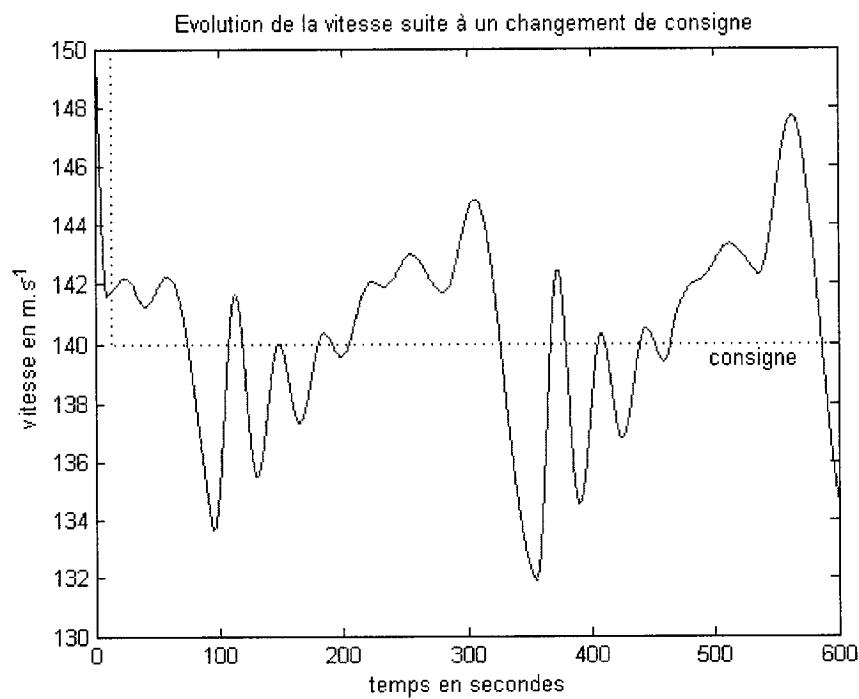


Figure 5.10 Commande en P_c - Réponse à un changement de consigne

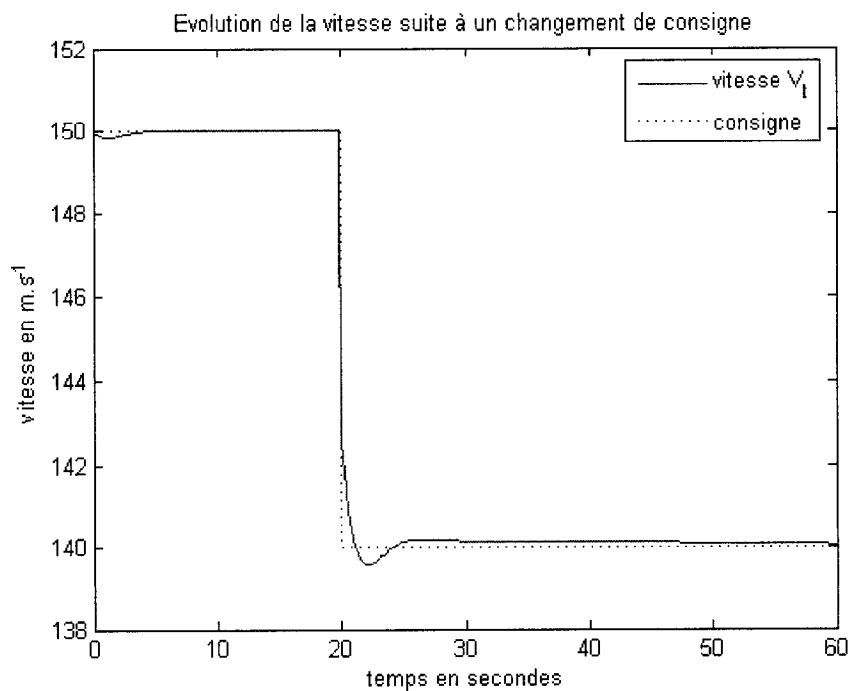


Figure 5.11 Commande en T - Réponse à un changement de consigne

5.2.2 L'allocation du contrôle

Différentes stratégies d'allocation du contrôle ont été mises en évidence dans le chapitre 4.3. Une d'entre elle est sélectionnée et détaillée.

Soulignons que l'allocation du contrôle nécessite une connaissance parfaite du modèle. La robustesse dépassant le cadre de ce projet, le modèle aérodynamique de l'avion est donc pris identique à celui identifié au moyen du module d'identification par réseaux de neurones.

Le modèle aérodynamique choisi est non linéaire, donc les stratégies statiques ne sont pas retenues. Le temps de réponse des actionneurs étant plus rapide que celui du système, leur dynamique peut être négligée et l'allocation continue et adaptative, similaire à l'allocation statique, est utilisée.

Cette allocation nécessite de connaître la matrice de “control effectiveness”, qui traduit la relation entre les coefficients aérodynamiques et les déflections des surfaces de contrôle, et de choisir une matrice de pondération W_δ .

Le module d'identification par réseaux de neurones permet de calculer la matrice de “control effectiveness” $B(t)$:

$$B(t) = \begin{bmatrix} \frac{\partial C_l}{\partial \delta_e} & \frac{\partial C_l}{\partial \delta_a} & \frac{\partial C_l}{\partial \delta_r} \\ \frac{\partial C_m}{\partial \delta_e} & 0 & 0 \\ \frac{\partial C_n}{\partial \delta_e} & \frac{\partial C_n}{\partial \delta_a} & \frac{\partial C_n}{\partial \delta_r} \end{bmatrix} \quad (5.2)$$

où les dérivées sont calculées conformément à l'équation 4.17.

Lorsque les contraintes ne sont pas prises en compte, la commande en déflexion des surfaces de contrôle δ_c est calculée directement à chaque instant :

$$\delta_c(t) = \delta_{mes}(t^-) + W_\delta^{-1} (B(t)^T W_\delta^{-1} B(t))^{-1} (C_c(t) - C_{mes}(t^-)) \quad (5.3)$$

où $\delta_{mes}(t^-)$ est la position mesurée et $C_c(t^-)$ la commande en coefficients aérodynamiques.

On remarque que, comme $B(t)$ est carrée, lorsqu'elle est inversible, la matrice de pondération W_δ ne joue aucun rôle. C'est le cas pour des angles d'attaques faibles. La matrice de pondération est donc choisie égale à celle préconisée au chapitre précédent, à savoir :

$$W_\delta = \begin{bmatrix} \frac{1}{25^2} & 0 & 0 \\ 0 & \frac{1}{20^2} & 0 \\ 0 & 0 & \frac{1}{30^2} \end{bmatrix} \quad (5.4)$$

Lorsque les contraintes sont prises en compte, l'utilisation d'une S-fonction, sans variable d'état et dont la sortie appelle la fonction MatLab *quadprog*, permet de calculer la commande :

```
function sys=mdlOutputs(t,x,u,Wdelta,delta_max,delta_min)
% fonction de cout a minimiser J = u'Wdelta u :
H = Wdelta;
f = zeros(3,1);
% contraintes d'inegalite :
A = [1 0 0;
      -1 0 0;
      0 1 0 ;
      0 -1 0 ;
      0 0 1 ;
      0 0 -1];
B = [delta_max(1) - u(1); % ou u(1) = delta_e_mes
      -delta_min(1) + u(1);
      delta_max(2) - u(2); % ou u(2) = delta_a_mes
      -delta_min(2) + u(2);
      delta_max(3) - u(3); % ou u(3) = delta_r_mes
      -delta_min(3) + u(3)];
% contraintes d'egalite : Delta_C = B(t)*Delta_delta
Aeq = [u(5) u(6) u(7); % B(t)
        u(9) 0 0;
        u(11) u(12) u(13)];
Beq = [u(15)-u(18); % Delta_C = C_c - C_mes
        u(16)-u(19);
        u(17)-u(20)];
% programmation quadratique :
r = quadprog(H,f,A,B,Aeq,Beq);
sys = r;
```

Pour cette étude, on émet l'hypothèse que les manœuvres étudiées n'entraînent pas de saturation, les contraintes ne sont donc pas explicitement prises en compte.

Le F-16 possède, outre les trois surfaces de contrôle usuelles, une surface complémentaire : celle du bord d'attaque. Traditionnellement celle-ci est commandée automatiquement, mais on peut envisager de la commander à partir de l'allocation. La matrice de “control effectiveness” devient alors :

$$B(t) = \begin{bmatrix} \frac{\partial C_l}{\partial \delta_e} & \frac{\partial C_l}{\partial \delta_a} & \frac{\partial C_l}{\partial \delta_r} & \frac{\partial C_l}{\partial \delta_{lef}} \\ \frac{\partial C_m}{\partial \delta_e} & 0 & 0 & \frac{\partial C_l}{\partial \delta_{lef}} \\ \frac{\partial C_n}{\partial \delta_e} & \frac{\partial C_n}{\partial \delta_a} & \frac{\partial C_n}{\partial \delta_r} & \frac{\partial C_l}{\partial \delta_{lef}} \end{bmatrix} \quad (5.5)$$

Dans ce cas la matrice $B(t)$ n'est pas carrée. La matrice de pondération prend alors toute son importance. La figure 5.12 illustre son influence pour les différents choix suivants de W_δ .

$$W_1 = \begin{bmatrix} \frac{1}{25^2} & 0 & 0 & 0 \\ 0 & \frac{1}{20^2} & 0 & 0 \\ 0 & 0 & \frac{1}{30^2} & 0 \\ 0 & 0 & 0 & \frac{1}{25^2} \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad W_4 = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

Pour l'allocation sur l'ensemble des quatre surfaces de contrôle, le choix $W_\delta = W_4$ permet d'obtenir la meilleure performance parmi les 4 possibilités illustrées. Cette allocation est légèrement plus oscillante que celle sur les trois surfaces de contrôle classiques. En terme de régulation, ces deux stratégies donnent des performances comparables.

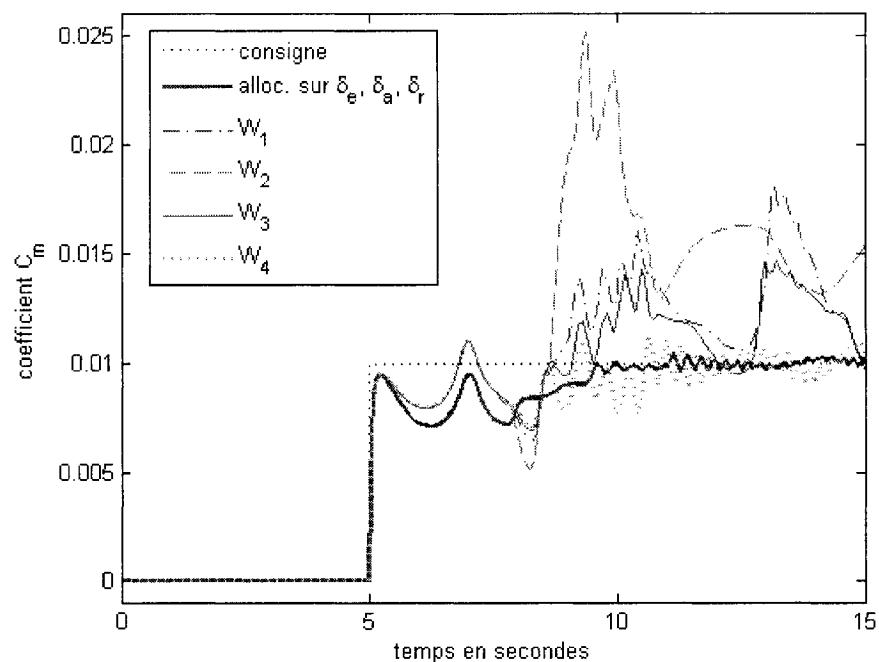
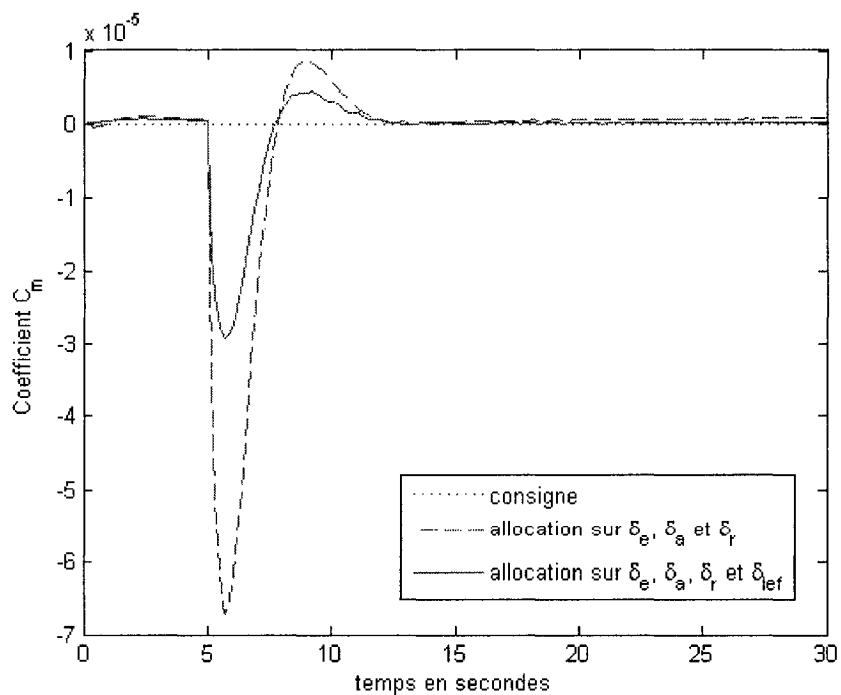
Figure 5.12 Influence de W_δ sur l'allocation

Figure 5.13 Allocation du contrôle - rejet de perturbation

L'interaction entre l'allocation du contrôle et l'autothrottle est alors étudiée. La valeur de consigne des coefficients aérodynamiques en moment est prise nulle, tandis que celle de la vitesse subit un échelon à $t = 5s$ la faisant passer de $150m.s^{-1}$ à $140m.s^{-1}$.

Le changement de vitesse entraîne une perturbation sur la valeur des coefficients aérodynamiques en moment. La figure 5.13 illustre le comportement des deux stratégies. On observe que l'utilisation de l'ensemble des surfaces de contrôle permet une meilleure réjection de la perturbation d'un facteur d'ordre 2.

L'ensemble de ces considérations permet de conclure que, dans le cas du F-16, une allocation sur l'ensemble de ses quatre surfaces de contrôle est préférable.

5.2.3 Choix des gains des dynamiques désirées

Les gains du contrôle par inversion dynamique sont pris de manière à satisfaire les qualités de manœuvrabilité. Le F-16 fait partie des avions de la classe IV. Les manœuvres considérées pour les simulations font partie des phases de vol de catégorie B. Seul le cas de vol nominal est considéré dans cette étude, les qualités de manœuvrabilité doivent donc être de niveau I.

Pour la dynamique du taux de tangage q , (ω_q, ζ_q) sont caractéristiques du mode rapide longitudinal. On désire fixer l'amortissement du mode rapide à $\zeta_{mr} = 1$ et le CAP à $CAP_{des} = 2$, d'où

$$\zeta_q = \zeta_{mr} = 1 \text{ et } \omega_q = \max \left\{ \sqrt{CAP_{des} \frac{n_z}{\alpha}}, 1 \right\}$$

Pour la dynamique du taux de roulis p , ω_p est caractéristique du mode roulis amorti. Pour les phases de vol de catégorie B, il faut $\omega_{ra} = \frac{1}{1,4}$; pour celles de catégorie A, $\omega_{ra} = 1$. Alors ω_p est choisi égal à ω_{ra} .

Pour la dynamique du taux de lacet r , ω_r est caractéristique du roulis hollandais. On désire fixer la pulsation propre du roulis hollandais à $\omega_{rh} = 3$, l'amortissement étant fixé à $\zeta_{rh} = 1$ avec le type de dynamique désirée choisi. D'où $\omega_r = \omega_{rh}$.

Pour la dynamique de l'angle d'attaque α , $(\omega_\alpha, \zeta_\alpha)$ sont caractéristiques du mode rapide longitudinal. Les caractéristiques ont été fixées pour la dynamique du taux de tangage, cependant l'hypothèse de séparation temporelle de la dynamique doit être respectée. On choisit alors

$$\zeta_\alpha = \zeta_{mr} = 0,7 \text{ et } \omega_\alpha = 2$$

Pour la dynamique de l'angle de dérapage β , $(\omega_\beta, \zeta_\beta)$ sont caractéristiques du roulis hollandais. L'amortissement du roulis hollandais est choisi égal à $\zeta_{rh} = 0,7$ ici. On a alors

$$\zeta_\beta = \zeta_{rh} \text{ et } \omega_\beta = \omega_{rh}$$

Pour la dynamique de l'assiette de roulis ϕ , ω_ϕ est caractéristique du mode spiral. Pour les phases de vol de catégorie B, il faut $T_{2_s} = 20$; pour celles de catégorie A, $T_{2_s} = 12$. Alors ω_r est choisi égal à $\omega_\phi = T_{2_s} \ln 2$.

Tout en étant conformes aux qualités de manœuvrabilité, les valeurs des amortissements et des pulsations propres ont été choisies arbitrairement ici. Celles-ci sont directement liées aux gains des fonctions de transfert des différentes variables d'état. Le problème de la robustesse n'a pas été étudié dans le cadre de ce mémoire, cependant il est suggéré d'adapter la méthodologie développée par Saussié et Saydy (2003) afin d'obtenir un choix plus pertinent de gains.

La performance de cette commande est mise en évidence dans le paragraphe suivant au moyen de simulations effectuées sous MatLab et Simulink[©].

5.3 Simulations en boucle fermée

Dans cette section quelques résultats de simulation sont présentés. Ils mettent en évidence les performances du contrôleur à inversion dynamique augmentée du retour proportionnel + intégral qui a été développé dans cette étude.

La trajectoire à suivre est la suivante :

$$\begin{aligned}
 V_t &= 150m.s^{-1}, & \alpha_{ref} &= \alpha_{trim} \text{ } ^\circ, \phi_{ref} = 0 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 0 \leq t < 2 \\
 V_t &= 150m.s^{-1}, & \alpha_{ref} &= \alpha_{trim} \text{ } ^\circ, \phi_{ref} = 30 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 2 \leq t < 5 \\
 V_t &= 150m.s^{-1}, & \alpha_{ref} &= 4 \text{ } ^\circ, \phi_{ref} = 30 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 5 \leq t < 10 \\
 V_t &= 150m.s^{-1}, & \alpha_{ref} &= 4 \text{ } ^\circ, \phi_{ref} = 0 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 10 \leq t < 15 \\
 V_t &= 150m.s^{-1}, & \alpha_{ref} &= -1 \text{ } ^\circ, \phi_{ref} = 0 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 15 \leq t < 20 \\
 V_t &= 140m.s^{-1}, & \alpha_{ref} &= -1 \text{ } ^\circ, \phi_{ref} = 0 \text{ } ^\circ, & \beta_{ref} &= 0 \text{ } ^\circ, \text{ pour } 20 \leq t < 30
 \end{aligned}$$

5.3.1 Cas d'un modèle parfait

Considérons tout d'abord le cas où le modèle aérodynamique de l'avion est parfait. A cette fin l'interpolation cubique permettant d'obtenir les coefficients aérodynamiques de l'avion est remplacée par le module d'identification à base de réseaux de neurones. Les figures 5.14 à 5.18 illustrent les réponses des différentes variables.

On observe que, à l'exception de la vitesse V_t et de l'angle de dérapage β , toutes les variables suivent parfaitement les consignes qui leur sont données. Ces consignes sont obtenues soit directement par le choix de la trajectoire pour la vitesse V_t , l'angle d'attaque α , l'angle de dérapage β et l'angle de roulis ϕ , soit par l'étape précédente du contrôle pour les vitesses en roulis p , en tangage q et en lacet r , les coefficients aérodynamiques de moment C_l , C_m et C_n et les déflections des surfaces de contrôle δ_e , δ_a , δ_r et δ_{lef} .

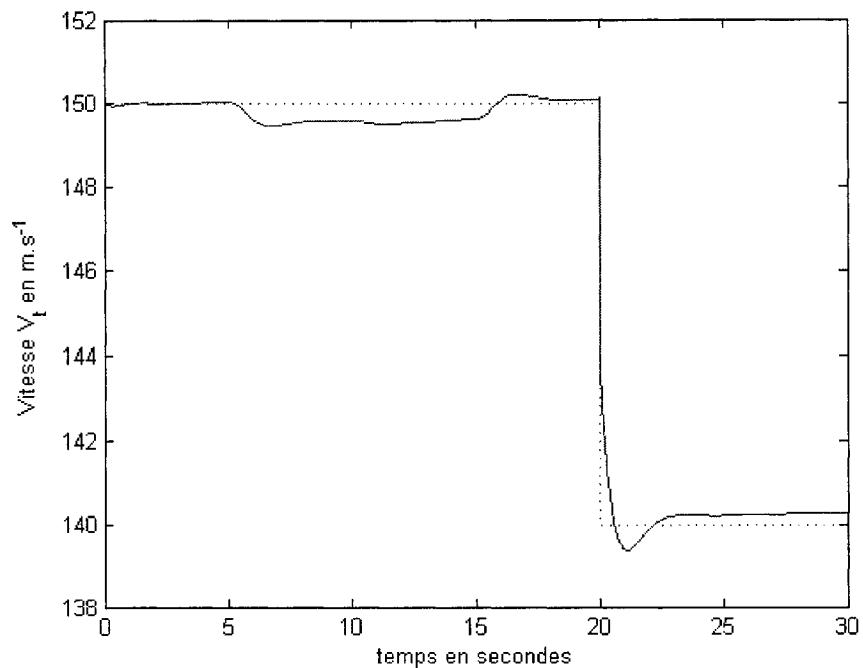


Figure 5.14 Suivi d'une trajectoire - réponse en vitesse V_t

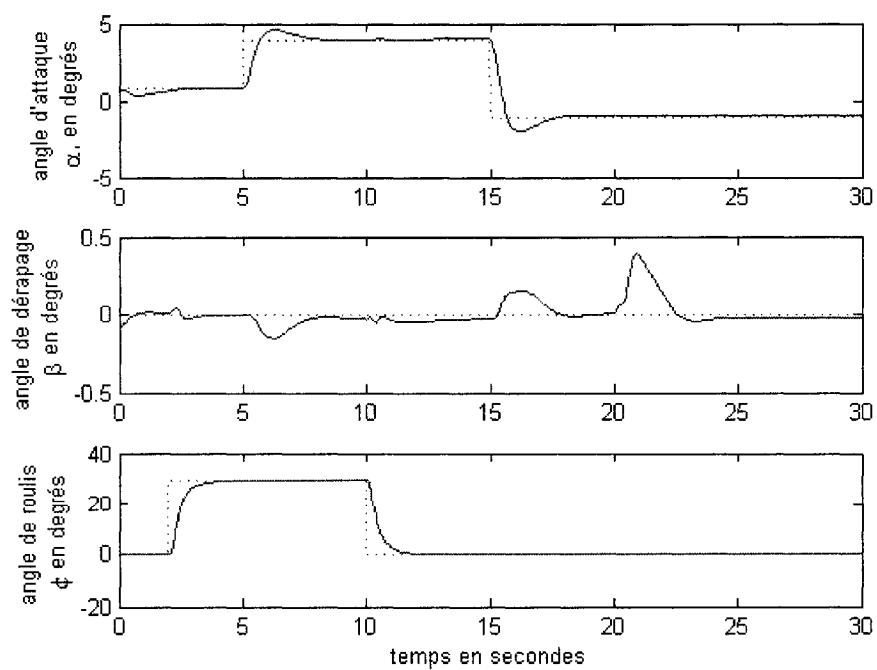


Figure 5.15 Suivi d'une trajectoire - réponse en angles α , β et ϕ

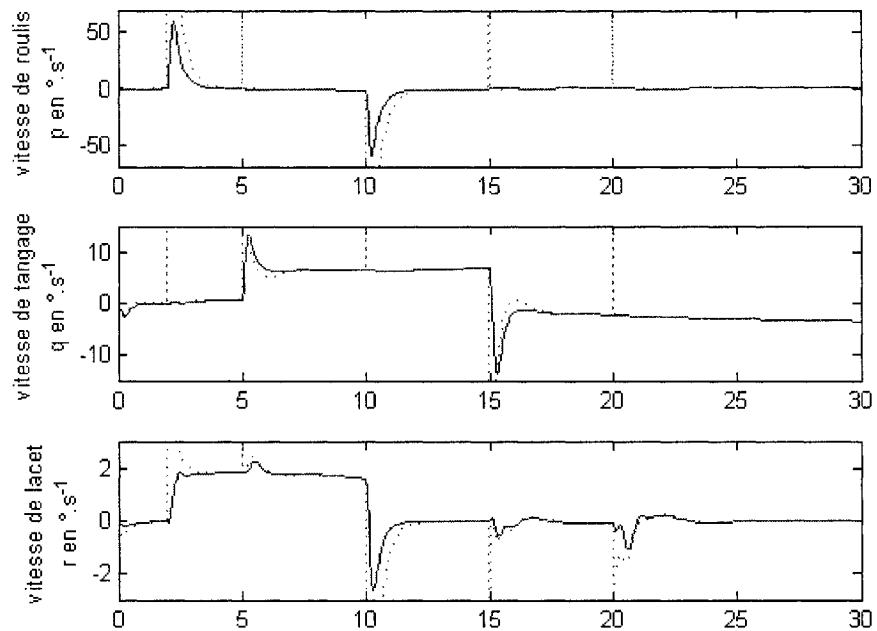


Figure 5.16 Suivi d'une trajectoire - réponse en vitesse de rotation p , q et r

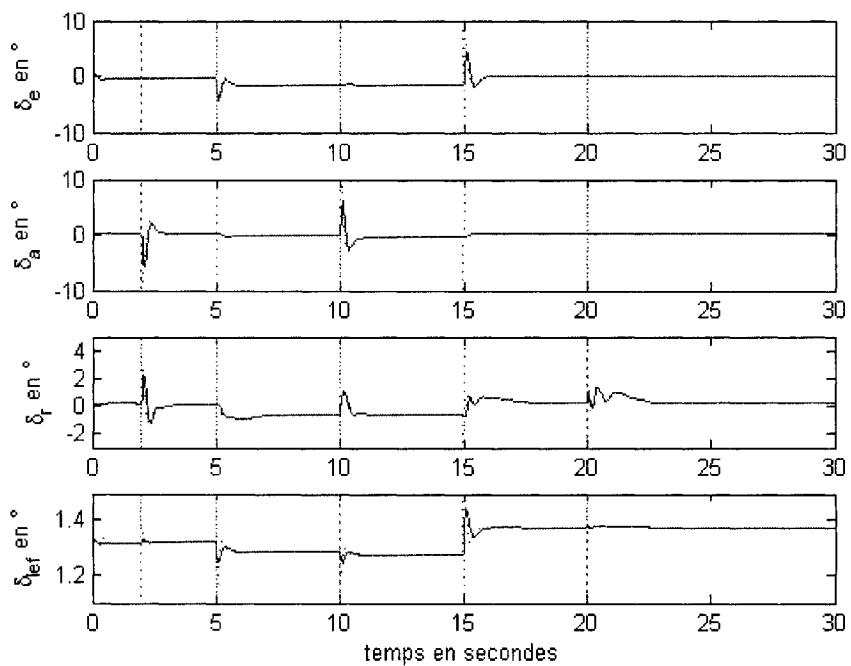


Figure 5.17 Suivi d'une trajectoire - déflections des surfaces δ_e , δ_a , δ_r et δ_{lef}

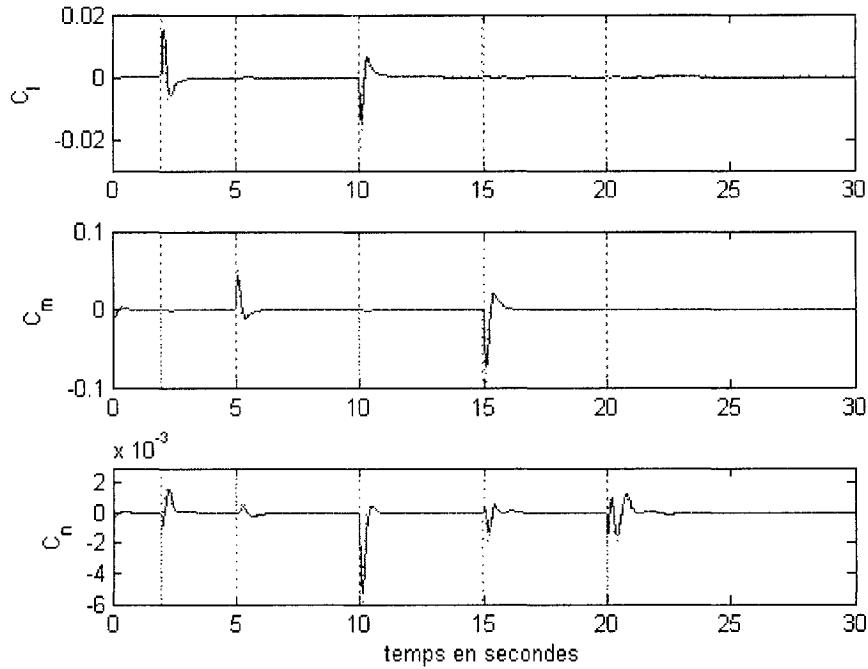


Figure 5.18 Suivi d'une trajectoire - coefficients aérodynamiques C_l , C_m et C_n

La vitesse V_t présente de légères différences avec sa consigne. Cela est dû au fait que l'autothrottle n'est pas totalement découplé du reste du contrôle. Mais cette perturbation reste très faible : inférieure à 0,005%. L'angle de dérapage β reste dans un voisinage de zéro ($< 0,5^\circ$), sa performance est donc jugée satisfaisante. Ainsi le contrôleur assure un bon suivi de trajectoire lorsque le modèle est parfait.

5.3.2 Cas d'un modèle “réel”

Observons désormais le comportement du système dans le cas où le modèle aérodynamique n'est pas parfait. L'identification des coefficients aérodynamiques est faite par le module à réseaux de neurones, tandis que le modèle “réel” des coefficients utilisé dans le simulateur du F-16 est obtenu par interpolation cubique des données provenant des tables du rapport de la NASA (Nguyen et al, 1979).

Le suivi de la trajectoire est quasiment aussi bon qu'avec le modèle parfait pour la vitesse V_t et l'angle d'attaque α . Le temps de réponse de l'angle de roulis est augmenté, de plus on observe une légère erreur statique. L'angle de dérapage reste dans un voisinage de zéro mais ses fluctuations sont plus importantes.

L'effort de contrôle est quasiment inchangé sur l'élévateur et les ailerons. La déflexion du bord d'attaque quant à elle est légèrement plus importante.

L'augmentation du contrôleur à inversion dynamique par des correcteurs de type proportionnel + intégral et l'identification des coefficients aérodynamiques à 1% près assurent ainsi une certaine robustesse. On peut donc conclure que globalement les performances du contrôleur sont satisfaisantes, même en présence d'un modèle non parfait.

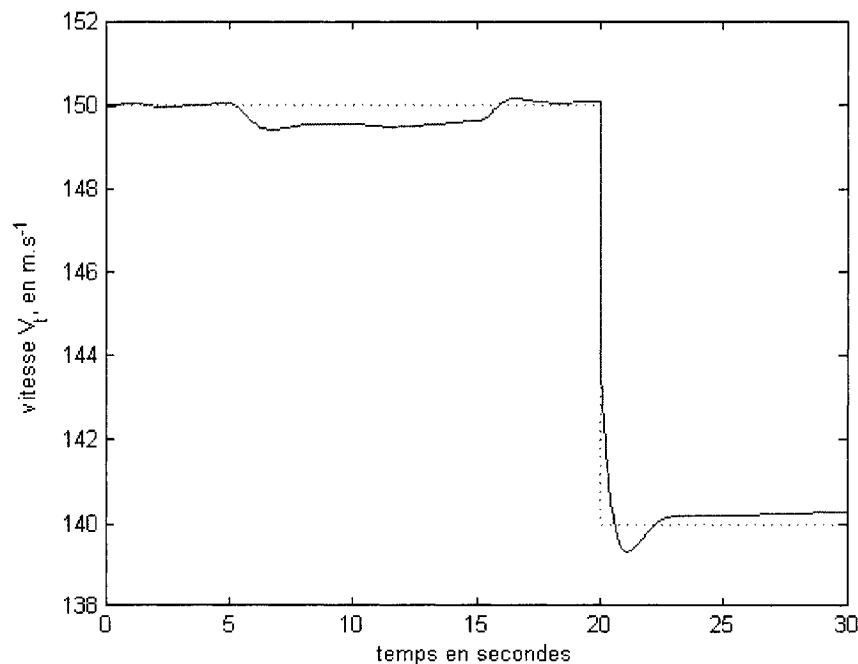


Figure 5.19 Suivi d'une trajectoire - réponse en vitesse V_t

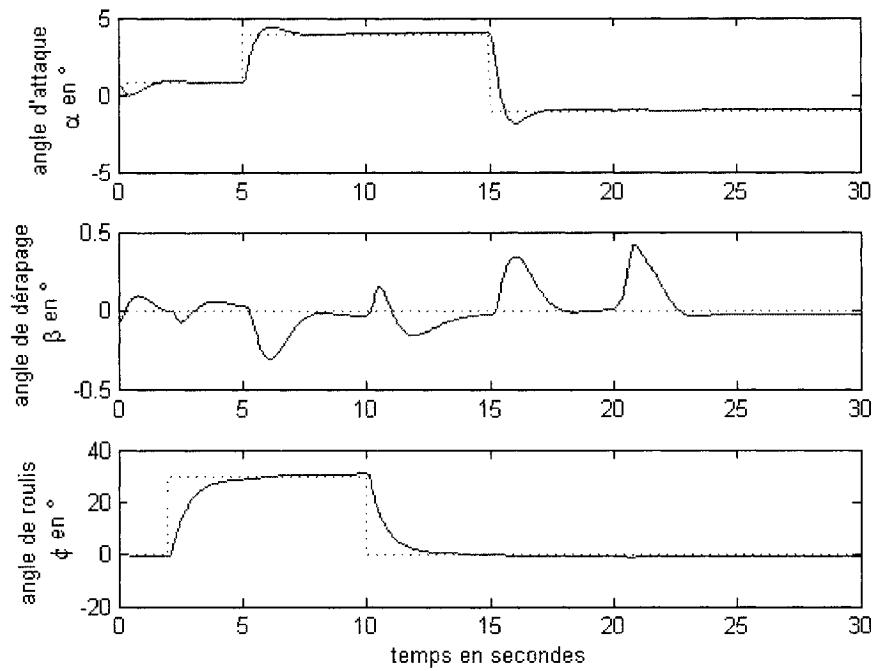


Figure 5.20 Suivi d'une trajectoire - réponse en angles α , β et ϕ

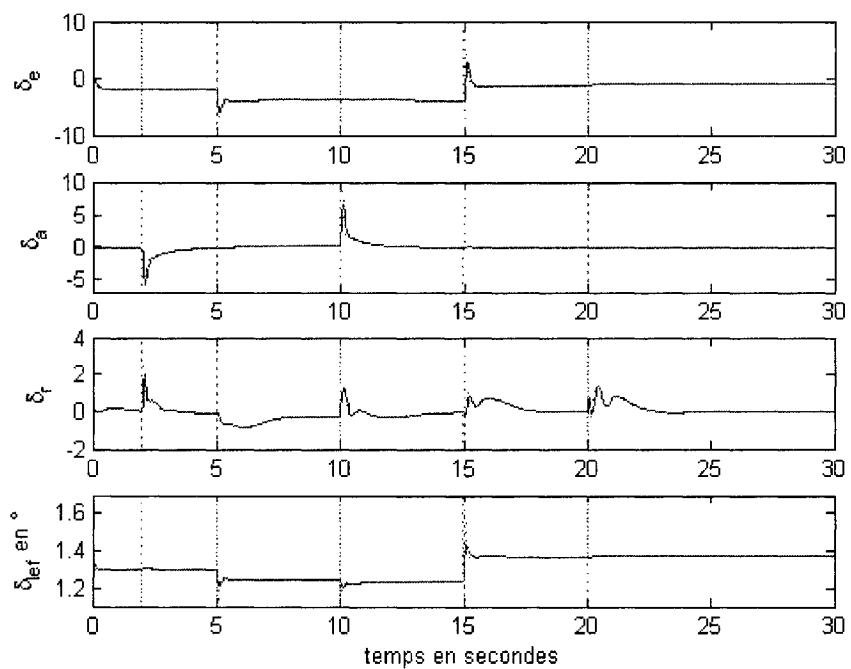


Figure 5.21 Suivi d'une trajectoire - déflection des surfaces δ_e , δ_a , δ_r et δ_{lef}

5.4 Une approche succincte à la commande prédictive

Comme indiqué au chapitre précédent, la commande prédictive est abordée succinctement ici. Dans un premier temps, l'influence des différents paramètres de l'algorithme est mis en évidence. La commande prédictive est ensuite appliquée au F-16, dont la boucle interne linéarise le modèle par retour de sortie.

5.4.1 Influence des différents paramètres

Différents paramètres sont à choisir dans la commande prédictive. Leur influence sur le comportement du système en boucle fermée est étudiée sur un second ordre de pulsation propre $\omega_n = 3$ et d'amortissement $\zeta = 0, 7$.

Le choix de la matrice R est à associer avec l'effort de contrôle. Plus R est grand, plus l'effort de contrôle est restreint. Cela se traduit par une anticipation plus longtemps à l'avance dans la commande prédictive (voir figure 5.22). Ce temps est toutefois limité par l'horizon de prédiction.

La matrice Q quant à elle affecte la sortie commandée du système. Plus Q est grande, plus le système est rapide et moins l'erreur permanente est importante (voir figure 5.23).

La figure 5.24 illustre l'influence de l'horizon de prédiction pour un temps d'échantillonage fixé. Plus l'horizon est grand, plus le changement de consigne est anticipé. Cette anticipation permet d'ailleurs de réduire l'effort de contrôle.

L'algorithme présenté à l'annexe V est aisément modifiable pour y incorporer une différence entre la période d'échantillonage et la durée pendant laquelle la commande est maintenue constante. Cette modification se traduit sur la matrice

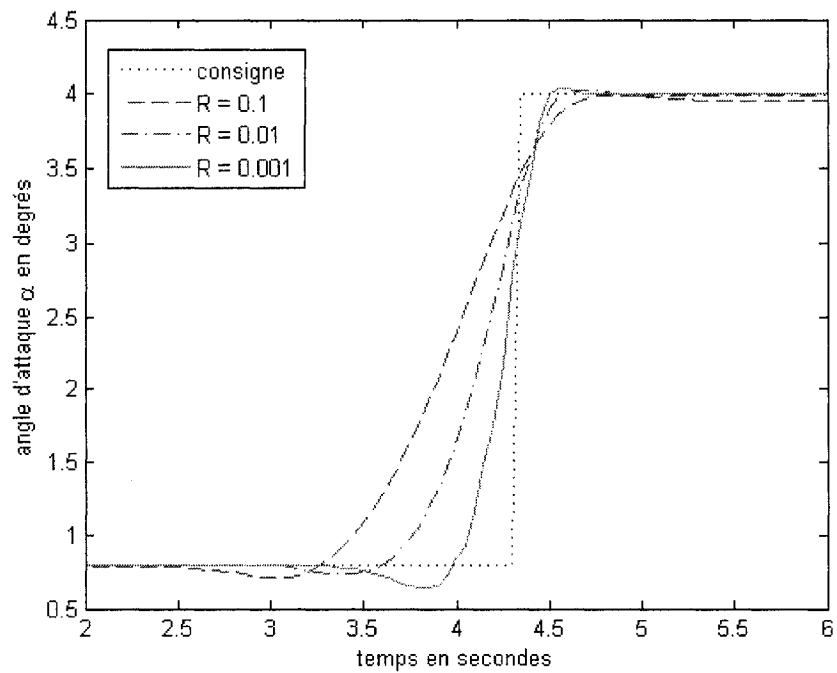


Figure 5.22 Influence de la matrice de pondération R

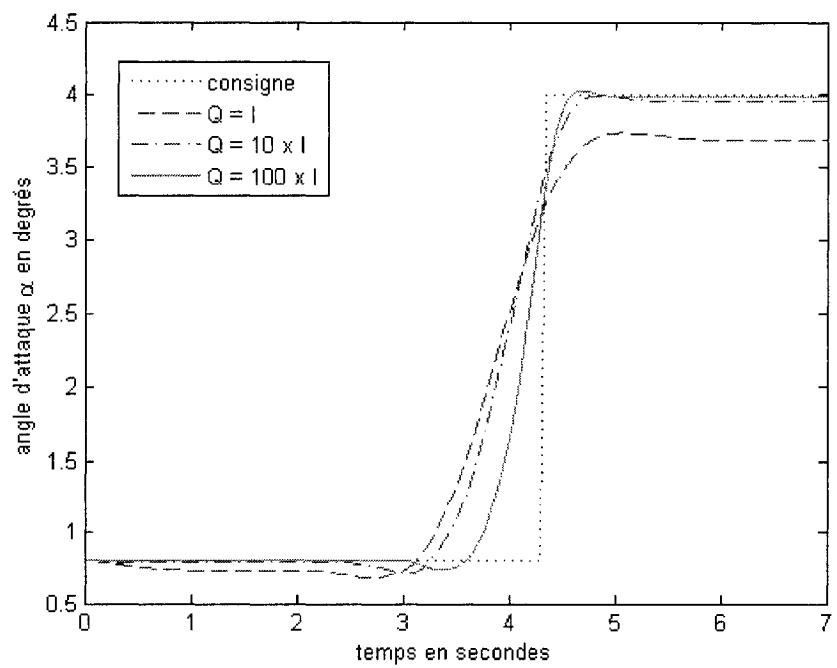


Figure 5.23 Influence de la matrice de pondération Q

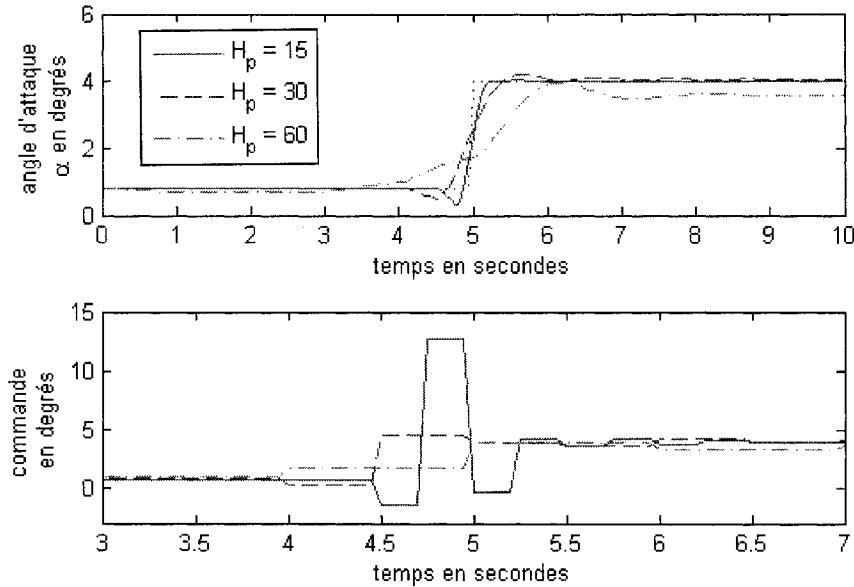


Figure 5.24 Influence de l'horizon de prédition

Bbkbar, dont la dimension est modifiée. La nouvelle matrice est obtenue quasiment en sommant les colonnes originelles et en augmentant les puissances en *Akbar*.

La durée sur laquelle la consigne est maintenue constante est essentiellement liée à la dynamique de l'actionneur. S'il est suffisamment rapide, des changements rapprochés de consigne peuvent être tolérés ; sinon mieux vaut augmenter la durée δH_p sur laquelle la consigne est maintenue constante. On remarque sur la figure 5.25 que plus δH_p est grand, plus la consigne est lisse.

L'intégration d'un modèle de perturbation simple par l'algorithme décrit à l'annexe V permet de rejeter des perturbations constantes. En effet, une perturbation constante de 1° est appliquée à l'angle d'attaque à $t = 4s$. La commande prédictive l'enraye rapidement comme le montre la figure 5.26.

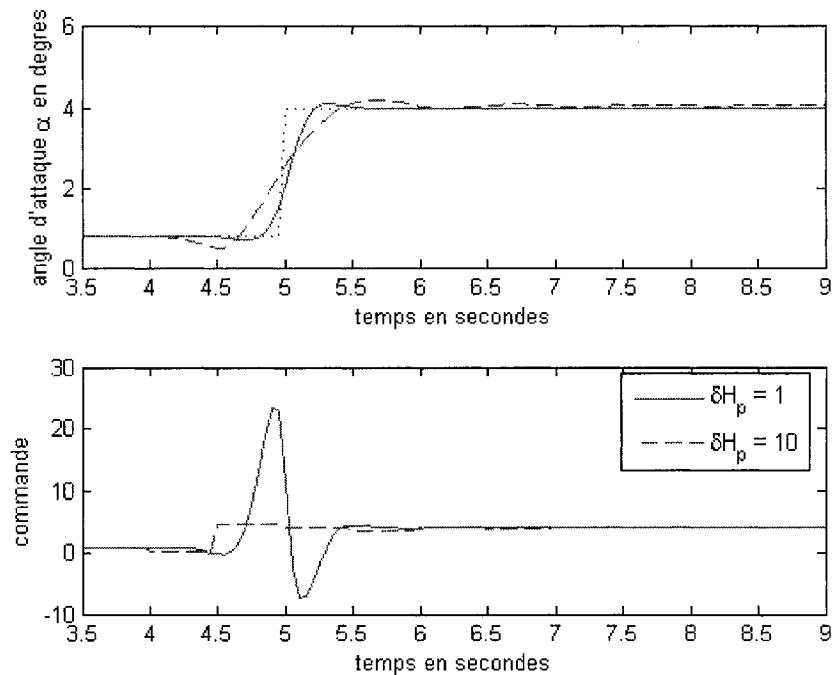


Figure 5.25 Influence du temps de maintien de la commande

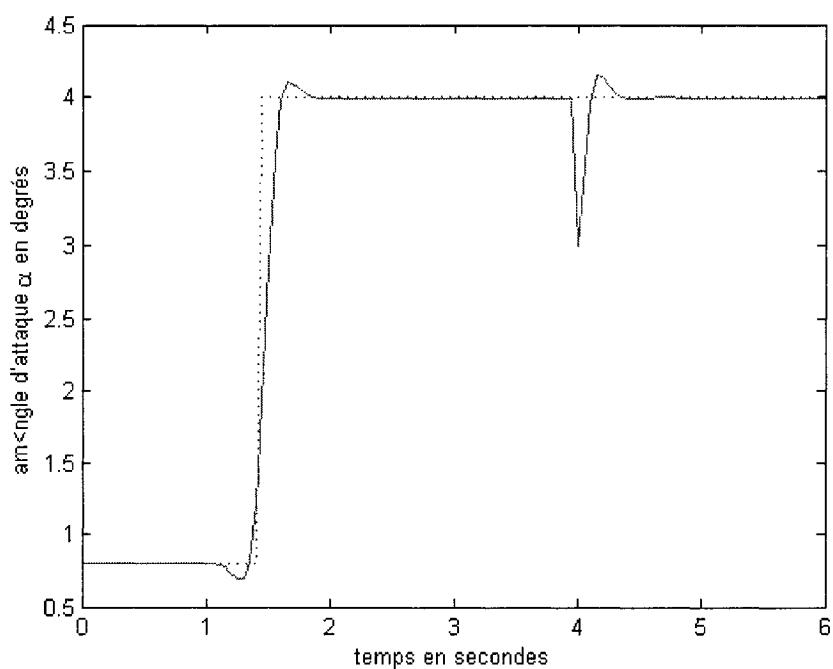


Figure 5.26 Rejet de perturbation constante

5.4.2 Simulation - suivi de trajectoire

La dynamique désirée de type proportionnel + intégral est remplacée par la commande prédictive pour l'ensemble des trois angles commandés : l'angle d'attaque α , l'angle de dérapage β et l'angle de roulis ϕ .

L'étude de l'influence des paramètres et leur application au modèle obtenu si l'inversion était parfaite mène aux choix suivants :

- $T_{ech} = 0,05s$
- $H_p = 1,5s$
- $\delta H_p = 0,5s$
- $Q = 10 \times I_3$
- $R = 0,01 \times I_3$

Les résultats de la simulation, pour la même trajectoire désirée qu'au paragraphe précédent, sont illustrés dans les figures 5.27 à 5.29.

Le suivi en vitesse, gérée par l'autothrottle, n'est pas affecté par ce changement.

On remarque que l'angle d'attaque, bien que stable, ne suit pas la trajectoire désirée et ce malgré la propagation des perturbations dans le modèle de la commande prédictive. On émet l'hypothèse que cela provient du fait que la linéarisation par retour de sortie n'est pas exactement vérifiée. En effet en l'absence du bouclage de la dynamique désirée de type proportionnel + intégral, la dynamique de l'angle d'attaque linéarisé par le retour de sortie devrait être la suivante :

$$\dot{\alpha} = \dot{\alpha}_{des} \quad (5.6)$$

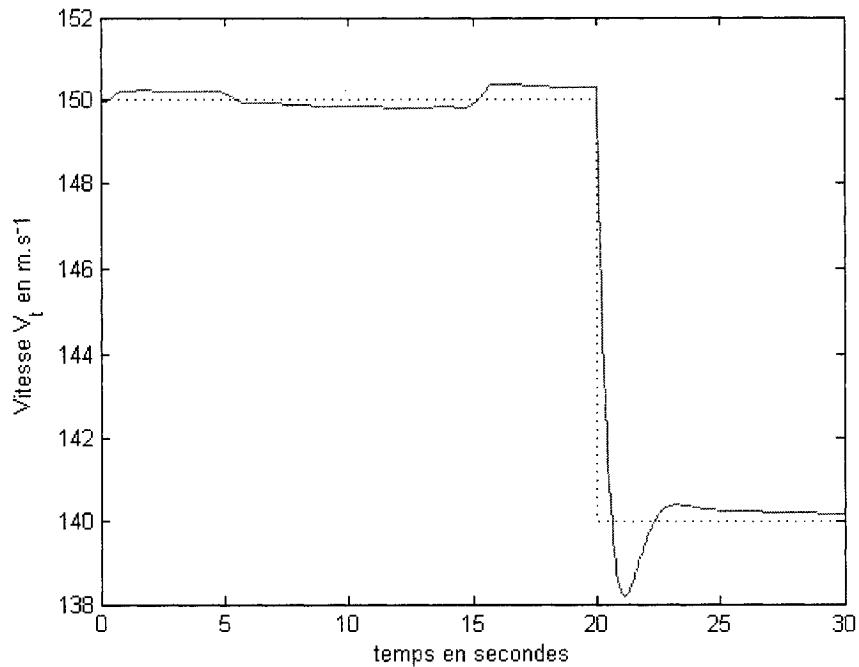


Figure 5.27 Suivi de trajectoire - réponse en vitesse V_t

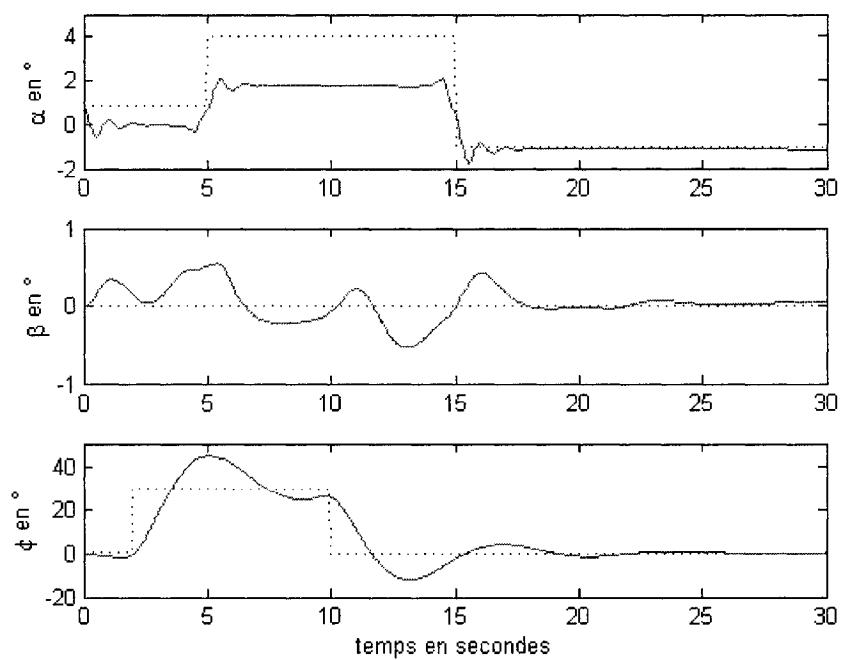


Figure 5.28 Suivi de trajectoire - réponse en α , β et ϕ

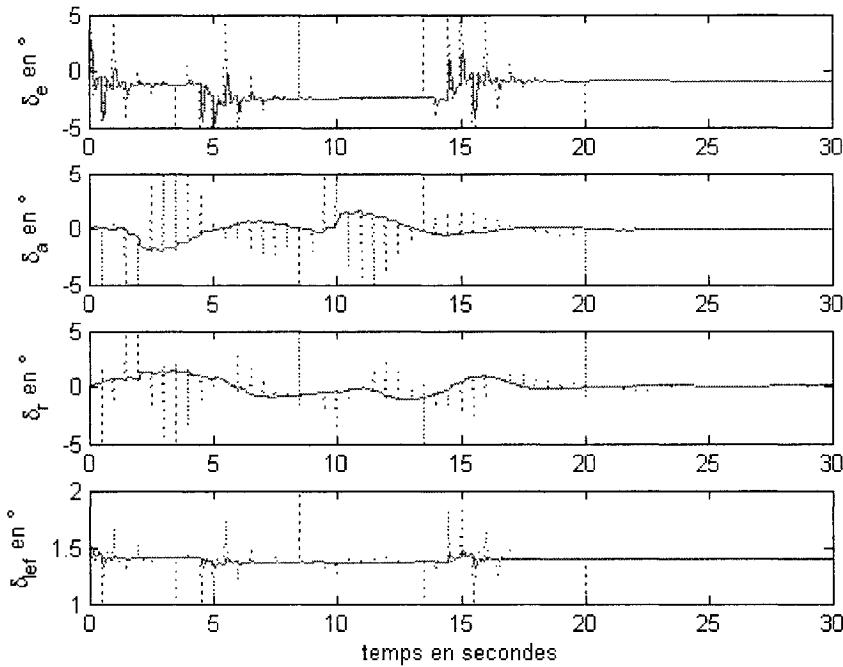


Figure 5.29 Suivi de trajectoire - réponse en δ_e , δ_a , δ_r et δ_{ref}

La figure 5.30 illustre la réponse du F-16 linéarisé par retour de sortie, dont l'entrée est la dérivée désirée d'angle d'attaque, à deux singlets d'amplitude 3° et d'une durée de $1,5s$. Celle-ci confirme le fait que la linéarisation n'est pas parfaite

$$\dot{\alpha} \neq v \quad (5.7)$$

On conclut de ces simulations que la commande prédictive telle qu'appliquée ici n'est pas suffisamment robuste. L'utilisation d'algorithmes robustes de commande prédictive permettrait de lever cet inconvénient. Toutefois une analyse plus poussée de la robustesse dépasse le cadre de ce projet.

Ainsi en comparaison, l'utilisation d'une dynamique désirée de type proportionnel + intégral assure de meilleures performances, et un meilleur suivi de la trajectoire. Un autre avantage de la dynamique désirée est qu'elle permet de fixer aisément le

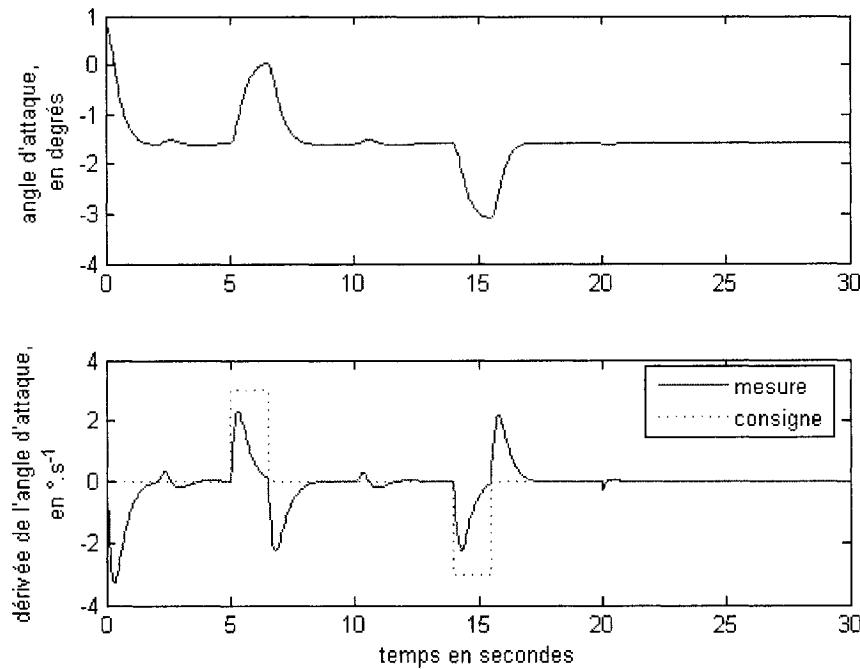


Figure 5.30 Performance du système linéarisé par retour de sortie sans PI

comportement en boucle fermée du système, afin de garantir les différentes qualités de manœuvrabilité.

5.5 Conclusion sur la simulation non linéaire

Un simulateur non linéaire du F-16 a été développé pour ces simulations. Le modèle de la dynamique en boucle ouverte du F-16 utilisé dans ce simulateur reflète de manière hautement fidèle le modèle mis en évidence dans un rapport de la NASA (Nguyen et al, 1979).

La simulation en boucle ouverte a permis de vérifier le comportement global de l'avion en boucle ouverte et de mettre en évidence des couplages entre les mouvements latéraux et longitudinaux.

Le module d'identification par réseau de neurones développé dans le chapitre 3 est incorporé dans le simulateur comme partie intégrante du contrôleur.

Le contrôleur par inversion dynamique est développé étape par étape. Cette construction pas à pas met en évidence d'une part certaines difficultés, telle que l'inversion de la dynamique du moteur, et d'autre part la relative simplicité de ce contrôleur par rapport à l'ensemble des contrôleurs linéaires locaux nécessaires par la technique traditionnelle de séquencement des gains.

La simulation effectuée d'abord avec un modèle aérodynamique parfait a validé le bon fonctionnement du contrôleur pour la trajectoire spécifiée. Les exemples trouvés dans la littérature sur les utilisations de l'inversion dynamique pour la commande de vol soulignaient le manque de robustesse de ce type de contrôle. Or les simulations effectuées ici avec un modèle "réel" de l'avion donnent des résultats conformes aux spécifications. Ces bons résultats s'expliquent par la combinaison de la génération de dynamiques désirées par un retour d'état de type proportionnel + intégral et du module d'identification neuronal. En effet, contrairement aux modèles linéaires utilisés classiquement, le module d'identification par réseaux de neurones développé dans le cadre de cette recherche fournit des approximations précises des coefficients et ce sur l'ensemble de l'enveloppe de vol, permettant de limiter les erreurs de modélisation et d'accroître la précision de l'inversion.

Les essais effectués en remplaçant la dynamique désirée des angles d'attaque, de dérapage et de roulis par de la commande prédictive ont souligné l'importance de la robustesse du contrôle le plus externe. En effet, si la linéarisation par retour de sortie avait été parfaite, les résultats obtenus auraient été ceux obtenus lors de l'étude de l'influence des paramètres ; or ils ne sont pas aussi concluants. L'autothrottle quant à elle permet une commande efficace de la vitesse, et ce qu'elle que soit la commande utilisée pour les angles (α, β, ϕ) .

On conclut que l'utilisation du module d'identification par réseaux de neurones, de l'inversion dynamique et d'une dynamique désirée de type proportionnel + intégral permet de commander les angles significatifs de l'avion sans avoir recours au séquencement des gains.

La tâche la plus importante en terme de temps de conception est le module d'identification, et celle-ci peut être automatisée grâce aux différents algorithmes mis en évidence au chapitre précédent. Au niveau du contrôleur, le nombre de gains à déterminer est limité à 12, et ce pour une commande valable sur l'ensemble de l'enveloppe de vol. Ces considérations mettent en évidence les avantages de la conception d'une telle commande par rapport aux centaines de contrôleurs linéaires locaux nécessaires dans la commande de vol classique.

CHAPITRE 6

VERS DES TRAVAUX FUTURS

Quelques pistes pour des travaux futurs sont présentées ici. Elles ont été abordées pour une première approche sous des hypothèses très restrictives. Mais cette étude préliminaire met en avant les points d'intérêt pour des travaux futurs.

En pratique, dans le cadre d'autopilotes, le pilote ne fournit pas une commande en (α, β, ϕ) , mais une commande sur la trajectoire à suivre, par exemple : V_t , h et ψ . La vitesse est directement commandée par l'autothrottle, tel qu'il a été vu au cours de cette étude. La référence en angle de dérapage β_{ref} est souvent prise nulle pour la coordination. L'utilisation d'une commande prédictive robuste permettrait alors de transformer la trajectoire désirée en une commande sur l'angle d'attaque α_c et l'assiette de roulis ϕ_c .

Sous l'hypothèse que le retour de sortie effectue une linéarisation exacte, la dynamique en (α, β, ϕ) de la boucle fermée par inversion dynamique, augmenté par une dynamique désirée de type proportionnel + intégral, a la forme suivante:

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_\alpha & -2\zeta_\alpha\omega_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\omega_\beta & -2\zeta_\beta\omega_\beta & 0 \\ 0 & 0 & 0 & 0 & -\omega_\phi \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \omega_\alpha^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \omega_\beta^2 & 0 \\ 0 & 0 & \omega_\phi \end{bmatrix} \begin{bmatrix} \alpha_c \\ \beta_c \\ \phi_c \end{bmatrix} \quad (6.1)$$

L'altitude h , l'assiette de tangage θ et l'azimut ψ sont régis par les équations

suivantes :

$$\begin{aligned}\dot{h} &= V_t \cos \alpha \cos \beta \sin \theta - V_t \sin \beta \cos \theta \sin \phi - V_t \sin \alpha \cos \beta \cos \theta \cos \phi \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= \frac{q \sin \phi + r \cos \phi}{\cos \theta}\end{aligned}\tag{6.2}$$

Les taux de rotation q et r sont supposés atteindre leur régime permanent très rapidement et sont donc assimilés à leur valeur de commande q_c et r_c . Celles-ci s'expriment, d'après les équations d'inversion, en fonction des coefficients aérodynamiques de force, des angles α , β et ϕ , de leur dérivées, de la vitesse V_t et de la poussée T . De la même manière, la poussée s'exprime en fonction de la vitesse, sa dérivée et les angles α , β , ϕ et θ .

La combinaison de ces équations permet d'exprimer le système sous la forme :

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \\ \dot{\phi} \\ \dot{h} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = f \left(\begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \\ \phi \\ h \\ \theta \\ \psi \end{bmatrix}, \begin{bmatrix} \alpha_c \\ \beta_c \\ \phi_c \end{bmatrix}, C_x, C_y, C_z \right)\tag{6.3}$$

Pour des manœuvres de faible amplitude, Keviczky et Balas (2006) établissent que l'utilisation de la commande prédictive avec un unique modèle linéaire permet de contrôler le mouvement longitudinal de l'avion de manière performante pour des manœuvres de faible amplitude.

On peut envisager d'appliquer cette technique sur le modèle décrit par l'équation 6.3. Cette approche permet de réduire le nombre d'états considérés, contrairement à l'approche de Keviczky et Balas qui augmentent le mode longitudinal de 5 états.

Comme première approche, on se limite au contrôle de l'altitude h à vitesse V_t fixée, dans le cas longitudinal soit $\beta = \phi = 0$. Seules les manœuvres de faibles amplitudes sont considérées, on peut donc légitimement faire l'hypothèse que l'angle d'attaque est faible. L'approximation au premier ordre donne alors :

$$\cos \alpha = 1 \text{ et } \sin \alpha = \alpha$$

De même on suppose que l'angle de tangage est petit. Dans ce cas, les équations en altitude h et en angle de tangage θ du système deviennent :

$$\begin{aligned} \dot{h} &= V_t \theta - V_t \alpha \\ \dot{\theta} &= q_c \end{aligned} \tag{6.4}$$

Les équations d'inversion donnent :

$$\begin{aligned} q_c &= \frac{1}{1 + \frac{\bar{q}S\bar{c}}{2mV_t} (C_{zq} \cos \alpha - C_{xq} \sin \alpha)} \times \left\{ \dot{\alpha} - \frac{g}{V_t} (\cos \theta \cos \alpha + \sin \theta \sin \alpha) \right. \\ &\quad - \frac{\bar{q}S}{mV_t} \left(C_{z0} \cos \alpha - \sin \alpha (C_{x0} + \frac{m}{\bar{q}S \cos \alpha} \dot{V}_t - C_x - C_z \tan \alpha - \right. \\ &\quad \left. \left. \frac{m}{\bar{q}S \cos \alpha} g (\cos \theta \sin \alpha - \sin \theta \cos \alpha)) \right) \right\} \end{aligned} \tag{6.5}$$

Choisissons la condition de référence comme étant $V_t = 150m.s^{-1}$ et $h = 5000m$. On y linéarise q_c de sorte à obtenir :

$$q_c \approx -\dot{\alpha} + Z_\theta \theta - Z_\alpha \alpha \tag{6.6}$$

Dans le cas d'une application au F-16 on a alors $Z_\alpha = 0,017$ et $Z_\theta = 0,065$.

La commande adaptative est alors appliquée au système suivant :

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{h} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_\alpha^2 & -2\omega_\alpha\zeta_\alpha & 0 & 0 \\ -V_t & 0 & 0 & V_t \\ -Z_\alpha & 1 & 0 & Z_\theta \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ h \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_\alpha \\ 0 \\ 0 \end{bmatrix} \alpha_c \quad (6.7)$$

De même que précédemment les différents paramètres sont fixés. La figure 6.1 illustre les performances pour différentes matrices de pondération. L'impact sur la commande générée en angle d'attaque (voir figure 6.2) est d'un intérêt particulier vu les hypothèses faites.

Les choix suivants sont faits :

- $T_{ech} = 0,05s$

- $H_p = 3s$

- $\delta H_p = 1s$

- $Q = \begin{bmatrix} \frac{1}{7500^2} & 0 \\ 0 & \frac{1}{20^2} \end{bmatrix}$

- $R = \frac{1}{10^2}$

Ils permettent un bon compromis entre le temps de réponse, une erreur statique faible et une commande limitée en angle d'attaque.

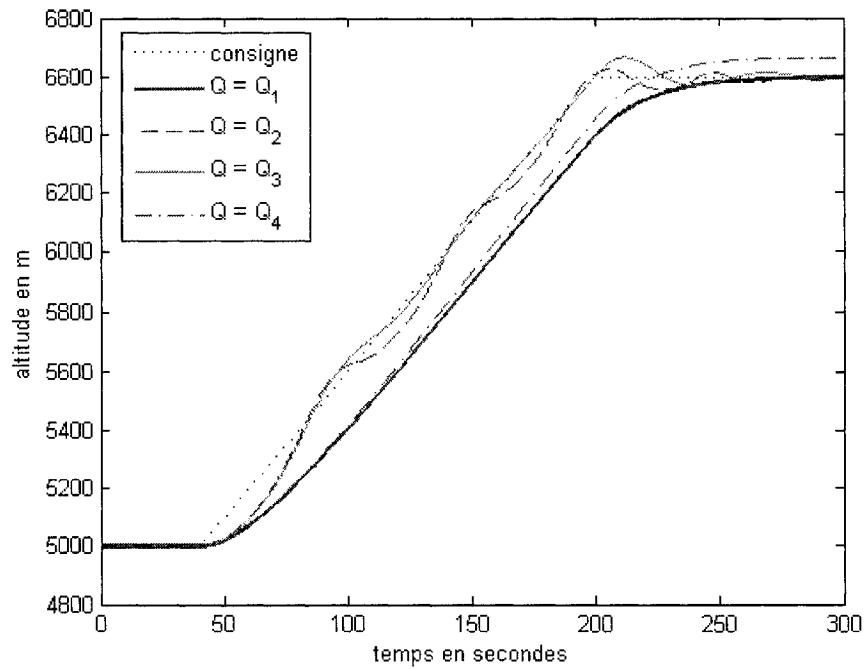
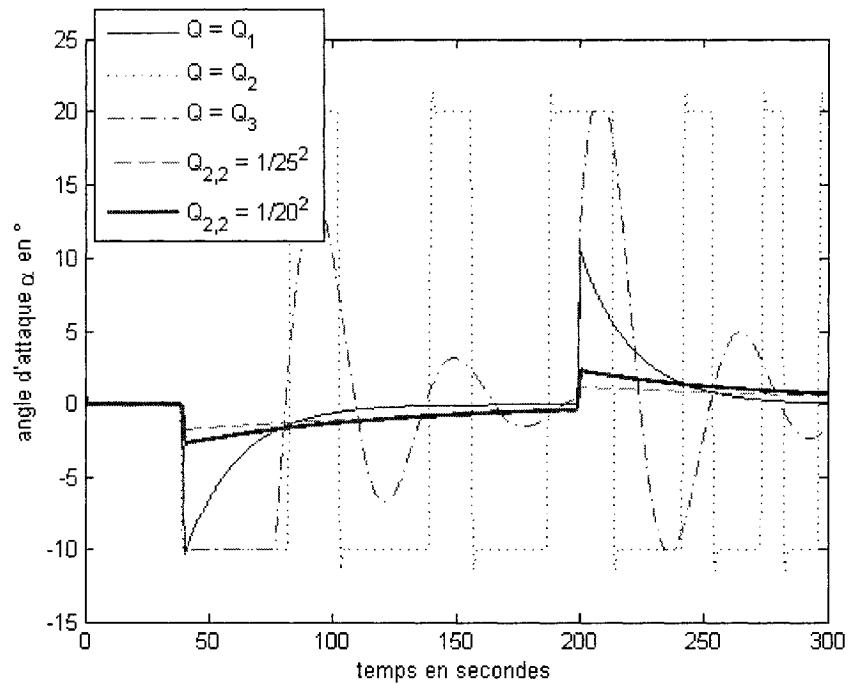
Figure 6.1 Commande en altitude - influence de Q 

Figure 6.2 Commande en altitude - angle d'attaque généré

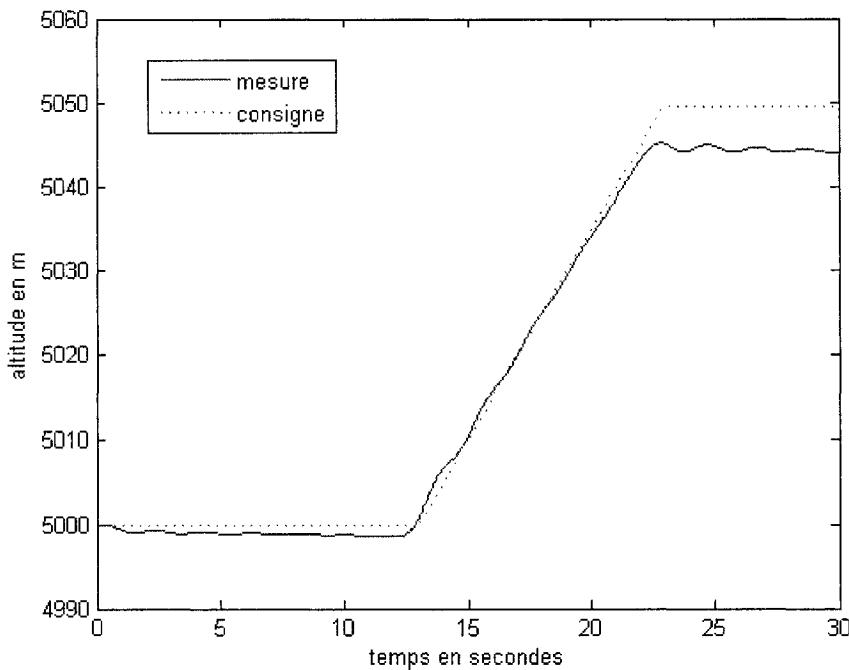


Figure 6.3 Commande en altitude - réponse en altitude

Ce contrôleur est alors appliqué au modèle non linéaire du F-16. Afin d'être dans les conditions des hypothèses faites pour cette étude préliminaire, les consignes en angle de dérapage β et de roulis ϕ sont prises nulles. La consigne en vitesse est prise constante égale à $150m.s^{-1}$.

Les figures 6.3 et 6.4 illustrent les réponses en altitude et en vitesse, suite à une rampe de pente $\dot{h} = 5m.s^{-1}$ appliquée à l'altitude entre les instants $t = 13s$ et $t = 23s$. On observe que la consigne en altitude est correctement suivie. L'erreur statique observée est intimement liée au choix des paramètres de la commande prédictive. Ce choix devrait être affiné. La vitesse conserve sa valeur de consigne à 2% près. Cependant le couplage existant entre l'altitude et la vitesse induit des oscillations sur cette dernière. Afin de réduire ce phénomène, la commande prédictive devrait incorporer la vitesse.

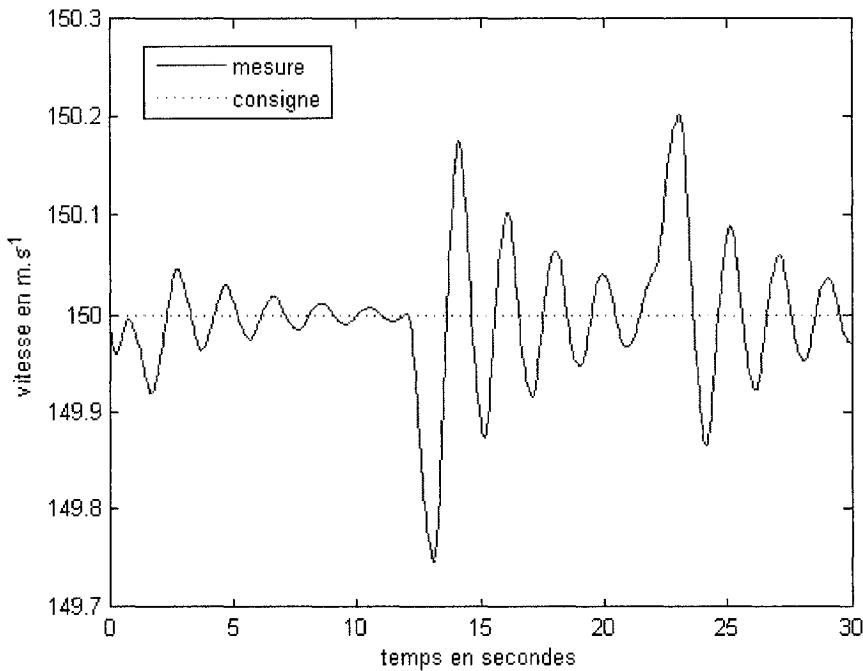


Figure 6.4 Commande en altitude - réponse en vitesse

Afin de vérifier l'affirmation de Keviczky et Balas (2006), selon laquelle le contrôleur développé à partir d'un unique modèle linéaire est valable sur une large part de l'enveloppe de vol, le contrôleur précédent est conservé, mais la condition de vol initiale du F-16 est modifiée. Les figures 6.5 à 6.7 illustrent le comportement du système pour différentes conditions initiales. Les consignes en altitude et en vitesse sont suivies avec précision.

Considérons désormais une manœuvre un peu plus agressive. La rampe d'altitude appliquée n'est plus de pente $\dot{h} = 5m.s^{-1}$ mais de $\dot{h} = 10m.s^{-1}$. On observe une divergence en altitude (voir figure 6.8). Les gains Q et R devraient être choisis de façon à assurer la stabilité du système.

Cette étude préliminaire met en évidence l'importance du choix des gains, notamment pour les répercussions que cela peut avoir sur des variables d'état en principe découplées par l'inversion dynamique et la stabilité du système.

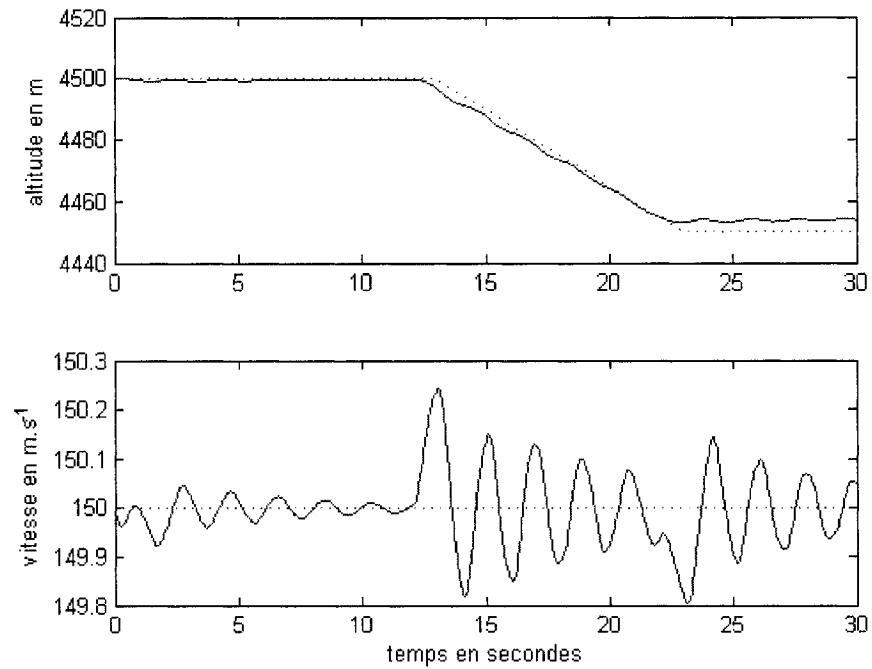


Figure 6.5 Commande en altitude - C.I. $V_t = 150m.s^{-1}$ et $h = 4500m$

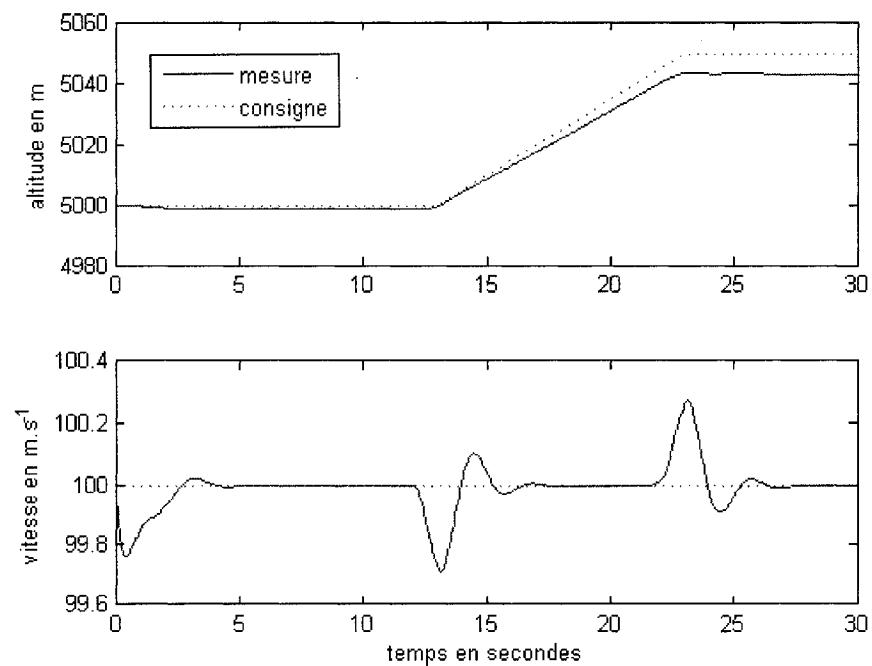


Figure 6.6 Commande en altitude - C.I. $V_t = 100m.s^{-1}$ et $h = 5000m$

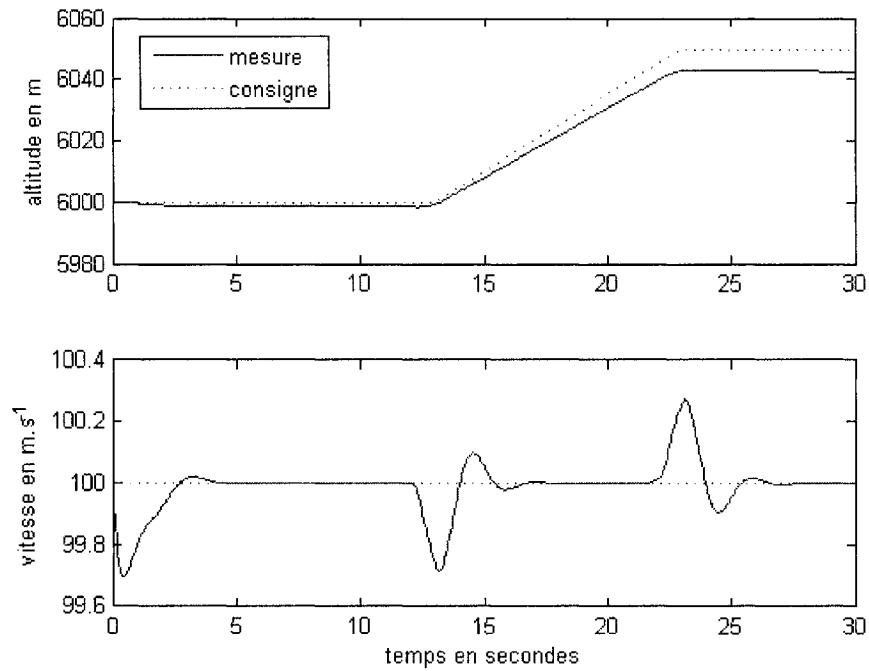


Figure 6.7 Commande en altitude - C.I. $V_t = 100m.s^{-1}$ et $h = 6000m$

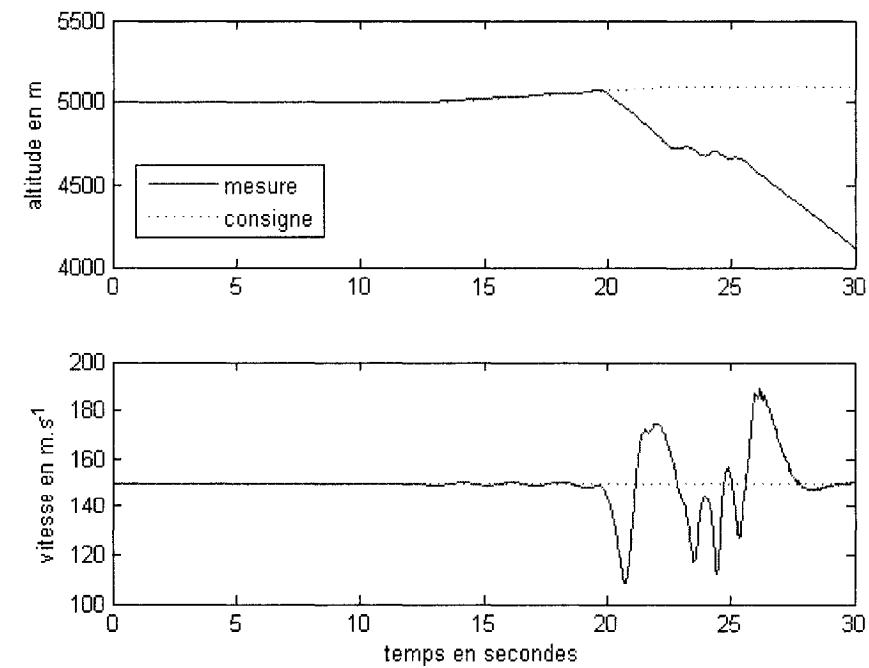


Figure 6.8 Commande en altitude - pente de $10m.s^{-1}$

Un autre point important à approfondir est le temps de calcul. En effet le calcul de la commande prédictive doit être compatible avec les constantes de temps du système.

Le module d'identification par réseaux de neurones doit également être suffisem-
ment rapide pour fournir les données nécessaires aux différentes boucles de contrôle.

L'évolution constante de la rapidité des processeurs et l'utilisation de processeurs parallèles laissent penser que cet aspect ne sera pas problématique.

CONCLUSION

Dans ce mémoire nous avons étudié l'identification des coefficients aérodynamiques d'un aéronef et la commande non linéaire de celui-ci. L'objectif était de se dégager des contraintes du "gain scheduling" grâce à la conception d'une loi de commande non linéaire basée sur un modèle hautement non linéaire d'un avion et de son comportement aérodynamique.

L'étude a été effectuée en deux parties : la première consiste en l'élaboration d'une architecture neuronale pour l'identification des coefficients aérodynamiques, et la seconde en l'élaboration d'une loi de commande non linéaire afin de commander les angles significatifs de l'aéronef (α, β, ϕ).

En s'appuyant sur l'article de De Weerdt (2005), un module d'identification des coefficients aérodynamiques a été élaboré. Ce module devait satisfaire deux objectifs principaux : identifier précisément les coefficients aérodynamiques sur l'ensemble de l'enveloppe de vol et être adapté à un apprentissage en ligne en vue de son utilisation éventuelle par une commande de vol reconfigurable.

Les réseaux de neurones et leur grande capacité d'adaptation ont été exploités de manière intensive dans le design. L'architecture proposée vise à garantir une certaine précision dans l'identification, tout en minimisant les inconvénients majeurs des réseaux de neurones, à savoir restreindre l'impact du phénomène d'oubli.

L'application au F-16 a donné des résultats très satisfaisants : les coefficients aérodynamiques de force ont été identifiés à 1% près par rapport à l'interpolation classiquement utilisée, et ceux en moment une précision de 2%. Le nombre de réseaux de neurones du module d'identification, ainsi que le nombre de neurones par réseau, est suffisamment restreint pour permettre une implémentation sur un processeur neuronal existant. Un frein majeur au design est le temps de simu-

lation pour trouver les neurones. Toutefois ce processus a été automatisé, il ne nécessite plus d'intervention humaine mais requiert un temps de processeur important. Celui-ci est hors ligne, et l'utilisation de processeurs plus rapide que le Pentium III à $450MHz$ utilisé permettrait de le réduire nettement.

Une commande non linéaire basée sur l'inversion dynamique est ensuite adaptée au système. L'innovation consiste à utiliser les coefficients aérodynamiques fournis directement par le module d'identification et non plus d'utiliser un modèle aérodynamique linéarisé. La linéarisation par retour de sortie prend comme consigne la dynamique désirée des variables contrôlées ; l'utilisation d'un type proportionnel + intégral permet de garantir une certaine robustesse au contrôleur.

Les gains de cette dynamique désirée sont directement liés aux qualités de manœuvrabilité, ce qui facilite leur compréhension physique et permet de les fixer arbitrairement dans un premier temps.

L'enjeu est désormais de concevoir la boucle de navigation, avec la commande en vitesse, en altitude et en angle de lacet. Une première approche par commande prédictive a été faite, elle a mis en évidence l'importance du choix de ses paramètres. En effet, ils sont déterminés à partir du modèle linéarisé par inversion dynamique. Or cette inversion et le découplage des variables d'état associées ne sont pas nécessairement exacts. L'utilisation d'algorithmes de commande prédictive robustes et non linéaire semble indiquée.

Les objectifs de ce projet ont ainsi été atteints. Un système de commande d'attitude non linéaire a été développé. Celui ci n'a pas recourt au découplage des mouvements longitudinaux et latéraux et obvie au "gain scheduling".

RÉFÉRENCES

- BABA, Y., TAKANO, H. & SANO, M. (1995). "Design of a nonlinear flight controller based on dynamic inversion", *Proceedings of the SICE Annual Conference*, 1487-1492.
- BODSON, M. & GROSZKIEWICZ, J. E. (1995). "Multivariable adaptive algorithm for reconfigurable flight control", *IEEE Transaction on Control Systems Technology*, 5:2, 217-229.
- CALISE, A. J., LEE, S. & SHARMA, M. (2001). "Development of a reconfigurable flight control law for tailless aircraft", *Journal of Guidance, Control and Dynamics*, 24:5, 896-902.
- CALISE, A. J. & RYSDYK, R. T. (1998). "Nonlinear adaptive flight control using neural networks", *IEEE Control System Magazine*, 18:6, 14-25.
- CALISE, A. J. & RYSDYK, R. T. (1998). "Adaptive nonlinear control for tiltrotor aircraft", *Proceedings of the IEEE Conference on Control Applications*, 2, 980-984.
- DE SANTIS, R. M. (2006). *Commande non linéaire - notes de cours*, 5ème ed., Montral, Qubec : École Polytechnique, 337 p.
- DE WEERDT, E. et al. (2005). "Neural network aerodynamic model identification for aerospace reconfiguration", *Collection of Technical PaPers - AIAA Guidance, Navigation, and Control Conference*, 8, 5962-5991.
- DUMUR, D. (2002). *Introduction à la commande adaptative*, Gif-sur-Yvette, France : Supélec, 50 p.

- ESCANDE, B. (1997). *HIRM design challenge presentation document : the non-linear dynamics inversion and linear quadratic approach* (GARTEUR/TP-088-30), France : Onera, 41 p.
- ESLINGER, R. A. & CHANDLER, P. R., 1988, "Self-repairing flight control system program overview", *IEEE Proceedings of the National Aerospace and Electronics Conference*, 504-511.
- ETKIN, B. (1972). *Dynamics of atmospheric flight*, New York : Wiley, 579.
- FARELL, J. A. & POLYCARPOU, M. M. (2006). *Adaptive approximation based control : unifying neural, fuzzy and traditional adaptive approximation approaches*, New York : Wiley, 420.
- GILI, P. A. & BATTIPEDE, M. (2001). "Adaptive neurocontroller for a nonlinear combat aircraft model", *Journal of Guidance, Control and Dynamics*, 24:5, 910-917.
- GROSKIEWICZ, J. E. & BODSON, M. (1997). "Flight control reconfiguration using adaptive methods", *Proceedings of the IEEE Conference on Decision and Control*, 2, 1159-1164.
- HALEY, P. J., SOLOWAY, D. & GOLD, B. (1999) "Real-time adaptive control using neural generalized predictive control", *Proceedings of the American Control Conference*, 6, 4278-4282.
- HAYKIN, S. (1999). *Neural networks : a comprehensive foundation*, 2nd ed., Upper Saddle River, N. J. : Prentice Hall, 842 p.
- HUANG, C. et al (1992). "Neural-network-based nonlinear inverse flight control", *Proceedings of the 31st conference on Decision and Control*, 2, 1982-1986.

- ITO, D. et al (1979). *Reentry vehicle flight controls design guidelines : dynamic inversion*, Lyndon B. Jonhson Space Center, Texas : National Aeronautics and Space Administration, 221 p., NASA/TP-2002-210771.
- JATEGAONKAR, R. V., 2006, *Flight vehicle system identification : a time domain methodology*. AIAA. Reston: Virginia. 534.
- KALE, M. M. & CHIPPERFIELD, A. J. (2002). "Reconfigurable flight control strategies using model predictive control", *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, 43-48.
- KAMALASADAN, S. & GHANDAKLY, A. A. (2005). "A fighter aircraft pitch rate control based on neural network parallel controller", *IEEE International Conference on Computational Intelligence for Measurements Systems and Applications*, 2005, 135-140.
- KEVICZKY, T. & BALAS, G. J. (2006). "Receding horizon control of an F-16 aircraft : A cmparative study", *Control Engineering Practice*, 14, 1023-1033.
- KIM, B. S. & CALISE, A. J. (1993). "Nonlinear flight control using neural networks and feedback linearization", *Proceedings of Aerospace Control Systems*, 176-181.
- KLEIN, V. & MORELLI, E. A., 2006, *Aircraft system identification : theory and practice*. AIAA Education Series. Blacksburg : Virginia, 484.
- LANE, S. H. & STENGEL, R. F. (1998). "Flight control using non-linear inverse dynamics", *Automatica*, 24:4, 471-483.
- LEE, T. & KIM, Y. (2001). "Nonlinear adaptive flight control using backstepping and neural networks controller", *Journal of Guidance, Control and Dynamics*, 24:4, 675-682.

- LUO, Y. et al. (2004). "Model predictive dynamic control allocation with actuator dynamics", *Proceedings of the 2004 American Control Conference*, 1695-1700.
- MACIEJOWSKY, J. M. (2001). *Predictive control with constraints*, London, Great Britain : Prentice Hall, 331 p.
- MENHAJ, M. B. & RAY, S. (2003). "Neuro-based adaptive controller for longitudinal flight control", *Proceedings of the 2003 IEEE International Symposium on Intelligent Control*, 158-163.
- MEYER, G., SU, R. & HUNT, L. R. (1984). "Application of nonlinear transformations to automatic flight control", *Automatica*, 20:1, 103-107.
- NARENDRA, K. S. & PARTHASARATHY, K. (1990). "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, 1:1, 4-27.
- NG, C-K. & CLARKSON, T. G., 1998, textslpRAM-256 VLSI neural network processor pRAM artificial neuron with learning in hardware, Department of Electronic Engineering, King's College, UK : London, 7.
- NGUYEN, L. T. et al (1979). *Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability*, Langley Research Center, VA : National Aeronautics and Space Administration, 221 p., NASA/TP-1979-1538.
- NHO, K. & AGARWAL, R. K. (1999). "Adaptive control of aircrafts dynamics using neural networks", *Proceedings on the International Joint Conference on Neural Networks*, 3, 2076-2080.
- NØRGAARD, M., RAVN, O., POULSEN, N. K. & HANSEN, L. K. (2000). *Neural Networks for modelling and control of Dynamic Systems*. Springer-Verlag. London : Great Britain. 246.

- PAMADI, B. N., 2004, *Performance, stability, dynamics, and control of airplanes*, 2nd ed., AIAA. Reston: Virginia. 780.
- PASHILKAR, A. A., SUNDARARAJAN, N. & SARATCHANDRAN, P. (2006) "Adaptive back-stepping neural controller for reconfigurable flight control systems", *IEEE Transactions on Control Systems Technology*, 14:3, 553-561.
- RICHARDSON, K. A., 1999, Identification of aerodynamic coefficients with a neural network, thèse de doctorat, Princeton University, Princeton : New Jersey, 220.
- SAUSSIE, D., SAYDY, L. & AKHRIF, O. (2003). "Flight Control Design with Robustness and Handling Quality Requirements ", *Proceedings of the 2003 Canadian Conference on Electrical and Computer Engineering*, 3, 1749-1752.
- SAVRAN, A., TASALTIN, R. & BECERIKLI, Y. (2006). "Intelligent adaptive nonlinear control of a high performance aircraft with neural networks", *The Instrumentation, Systems and Automation Society*, 45:2, 225-247.
- SOLOWAY, D. & HALEY, P. J. (2001). "Aircraft reconfiguration using neural generalized predictive control", *Proceedings of the American Control Conference*, 4, 2924-2929.
- SOMAKUMAR, R. & CHANDRASEKHAR, J. (1998). "Neural network based nonlinear inverse dynamics for flight controller design", *Proceedings of the 1998 IEEE International Conference on Control Applications*, 1, 187-191.
- STEVENS, B. L. & LEWIS, F. L. (1992). *Aircraft Control and Simulation*, 2nd ed., Hoboken, NJ : Wiley, 664 p.
- SUN, X-D. & CLARKE, T. (1994). "Advanced aircraft flight control using nonlinear inverse dynamics", *IEEE Proceedings on Control Theory and Applications*, 141:6, 418-426.

- SURESH, S. at al (2005). "Nonlinear lateral command using neural network for F-16 aircraft", *Proceedings of the 2005 American Control Conference*, 4, 2658-2663.
- SURESH, S., OMKAR, S. N. & MANI, V. (2006). "Direct adaptive flight control for F-8 fighter aircraft", *Journal of Guidance, Control and Dynamics*, 29:2, 454-464.
- U.S. Department of Defense Standard, 1997, *Flying qualities of piloted aircraft*, MIL-HDBK-1797A, Washington D.C. : Government printing office.
- WARD, D., MONACO, J. & BODSON, M. (1998). "Development and flight testing of a parameter identification algorithm for reconfigurable control", *Journal of Guidance, Control and Dynamics*, 21:6, 948-956.
- YU, C., ZHU, J. & SUN, Z. (2005). "Nonlinear adaptive internal model control using neural networks for tilt rotor aircraft platform", *2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, 12-16.
- ZAMARREÑO, J. M. & VEGA, P. (1999). "Neural predictive control. Application to a highly non-linear system", *Engineering Applications of Artificial Intelligence*, 12:2, 149-158.

ANNEXE I

RAPPELS DE DYNAMIQUE DU VOL

Les équations de la dynamique du vol issues des principes fondamentaux de Newton sont établies sous les conditions suivantes (Stevens & Lewis, 1992) :

- l'avion est considéré comme un corps rigide
- l'avion est supposé symétrique par rapport à ses axes x-z
- la masse de l'avion est prise constante sur la fenêtre d'observation
- il n'y a pas de déplacement de carburant entraînant une modification de la position du centre de gravité et a fortiori des caractéristiques aérodynamiques de l'avion
- la rotation de la terre est jugée négligeable, et son rayon de courbure infini
- seules les forces aérodynamiques, la force de poussée et celle de gravitation sont prises en compte
- la gravité est estimée constante et indépendante de l'altitude de l'avion.

I.1 Équations cinématiques et de navigation :

L'attitude de l'avion peut être définie par le triplet des angles d'Euler dans le repère NED. Ces angles sont l'angle de lacet ψ , l'angle de tangage θ et l'angle de roulis φ . Ils sont le résultat de trois rotations primaires effectuées à partir des axes du

repère NED pour obtenir ceux du repère avion (Stevens & Lewis, 1992) :

$$\begin{aligned}\dot{\phi} &= p + \tan \theta (q \sin \phi + r \cos \phi) \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= \frac{q \sin \phi + r \cos \phi}{\cos \theta}\end{aligned}\tag{I.1}$$

Les équations de navigation expriment la relation entre les vitesses dans le repère NED et celles dans le repère de l'avion, et permettent de suivre l'évolution de la position du centre de gravité de l'avion. En négligeant l'effet du vent elles s'écrivent:

$$\begin{aligned}\dot{x}_E &= u \cos \psi \cos \theta + v (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) \\ &\quad + w (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ \dot{y}_E &= u \sin \psi \cos \theta + v (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) \\ &\quad + w (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ \dot{h} &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi\end{aligned}\tag{I.2}$$

I.2 Équations des forces et des moments :

La position d'un corps dans l'espace est entièrement déterminée par six coordonnées spatiales, trois déterminant les vitesses suivant les axes du repère avion et trois déterminant les moments autour de ces mêmes axes. Suivant les hypothèses faites ci-dessus, les équations du modèle non linéaire à 6 degrés de liberté dans le système d'axes de l'avion sont (Stevens & Lewis, 1992) :

$$\begin{aligned}\dot{u} &= rv - qw - g \sin \theta + \frac{\bar{q}SC_x + T}{m} \\ \dot{v} &= pw - ru + g \cos \theta \sin \phi + \frac{\bar{q}SC_y}{m} \\ \dot{w} &= qu - pv + g \cos \theta \cos \phi + \frac{\bar{q}SC_z}{m}\end{aligned}\tag{I.3}$$

$$\begin{aligned}
I_x \dot{p} - I_{xz} \dot{r} &= \bar{q} S b C_l - (I_z - I_y) qr + I_{xz} qp \\
I_y \dot{q} &= \bar{q} S c C_m - (I_x - I_z) pr - I_{xz} (p^2 - r^2) + h_{moteur} r \quad (I.4) \\
I_z \dot{r} - I_{xz} \dot{p} &= \bar{q} S b C_n - (I_y - I_x) pq - I_{xz} qr - h_{moteur} q
\end{aligned}$$

où C_x , C_y , C_z , C_l , C_m et C_n sont les coefficients aérodynamiques. La poussée T du moteur est supposée agir selon l'axe des x et passer par le centre de gravité de l'avion, et son moment angulaire de la rotation h_{moteur} est supposé constant et agir selon l'axe des x uniquement.

Rappelons la relation entre (V_t, α, β) et (u, v, w) :

$$\begin{cases} V_t = \sqrt{u^2 + v^2 + w^2} \\ \alpha = \tan^{-1} \left(\frac{w}{u} \right) \\ \beta = \sin^{-1} \left(\frac{v}{V_t} \right) \end{cases} \iff \begin{cases} u = V_t \cos \alpha \cos \beta \\ v = V_t \sin \beta \\ w = V_t \sin \alpha \cos \beta \end{cases} \quad (I.5)$$

En termes de réalisation pratique il est plus judicieux d'utiliser V_t , α et β , plutôt que u , v et w , car les premiers sont directement mesurables sur l'avion au moyen de capteurs. En dérivant les équations à gauche dans I.5 par rapport au temps, on peut donc reformuler les équations I.3 sous la forme suivante :

$$\begin{aligned}
\dot{V}_t &= -\frac{\bar{q} S}{m} (-C_x \cos \alpha \cos \beta - C_y \sin \beta - C_z \sin \alpha \cos \beta) + \frac{T}{m} \cos \alpha \cos \beta \\
&\quad + g (\cos \phi \cos \theta \sin \alpha \cos \beta + \sin \phi \cos \theta \sin \beta - \sin \theta \cos \alpha \cos \beta) \\
\dot{\alpha} &= -\frac{\bar{q} S}{m V_t \cos \beta} (-C_z \cos \alpha + C_x \sin \alpha) + q - \tan \beta (p \cos \alpha + r \sin \alpha) \\
&\quad + \frac{g}{V_t \cos \beta} (\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha) - \frac{T \sin \alpha}{m V_t \cos \beta} \\
\dot{\beta} &= \frac{\bar{q} S}{m V_t} (-C_x \cos \alpha \sin \beta + C_y \cos \beta - C_z \sin \alpha \sin \beta) + p \sin \alpha - r \cos \alpha \\
&\quad + \frac{g}{V_t} \cos \beta \sin \phi \cos \theta + \frac{\sin \beta}{V_t} \left(g \cos \alpha \sin \theta - g \sin \alpha \cos \phi \cos \theta - \frac{T \cos \alpha}{m} \right) \quad (I.6)
\end{aligned}$$

Pour les équations relatives aux moments, il est préférable de découpler les équations I.4 en isolant un seul terme dérivé dans le membre de gauche :

$$\begin{aligned}\dot{p} &= (c_1 r + c_2 p - c_4 h_{moteur}) q + \bar{q} S b (c_3 C_l + c_4 C_n) \\ \dot{q} &= (c_5 p + c_7 h_{moteur}) r - c_6 (p^2 - r^2) + c_7 \bar{q} S \bar{c} C_m \\ \dot{r} &= (c_8 p - c_2 r - c_9 h_{moteur}) q + \bar{q} S b (c_4 C_l + c_9 C_n)\end{aligned}\quad (I.7)$$

où les c_i sont des constantes d'inertie :

$$\begin{aligned}c_1 &= \frac{(I_y - I_z) I_z - I_{xz}^2}{I_x I_z - I_{xz}^2} & c_2 &= \frac{(I_x - I_y + I_z) I_{xz}}{I_x I_z - I_{xz}^2} \\ c_3 &= \frac{I_z}{I_x I_z - I_{xz}^2} & c_4 &= \frac{I_{xz}}{I_x I_z - I_{xz}^2} \\ c_5 &= \frac{(I_z - I_x)}{I_y} & c_6 &= \frac{I_{xz}}{I_y} \\ c_7 &= \frac{1}{I_y} & c_8 &= \frac{(I_x - I_y) I_x - I_{xz}^2}{I_x I_z - I_{xz}^2} \\ c_9 &= \frac{I_x}{I_x I_z - I_{xz}^2}\end{aligned}\quad (I.8)$$

I.3 Facteurs de charge :

En supposant que la poussée du moteur est selon l'axe des x de l'avion et est alignée avec son centre de gravité, le facteur d'accélération en nombre de g selon x, y et z est donné par :

$$\begin{aligned}n_x &= \frac{a_x}{g} = \frac{\bar{q} S C_x + T}{mg} \\ n_y &= \frac{a_y}{g} = \frac{\bar{q} S C_y}{mg} \\ n_z &= \frac{a_z}{g} = \frac{\bar{q} S C_z}{mg}\end{aligned}\quad (I.9)$$

ANNEXE II

DONNÉES AÉRODYNAMIQUES DU F-16

L'ensemble des données provient du rapport de la NASA (Nguyen et al, 1979). Celui-ci fournit également la valeur des différentes contributions aux coefficients aérodynamique sur l'ensemble de l'enveloppe de vol du F-16.

Tableau II.1 Propriétés géométriques et de masse du F-16

Paramètre	Unité	Valeur
poids, W	N	91 188
moment d'inertie suivant l'axe des x, I_x	$kg.m^2$	12 875
moment d'inertie suivant l'axe des y, I_y	$kg.m^2$	75 674
moment d'inertie suivant l'axe des z, I_z	$kg.m^2$	85 552
produit d'inertie, I_{xz}	$kg.m^2$	1 331
envergure des ailes, b	m	9,144
surface des ailes, S	m^2	27,87
corde aérodynamique moyenne, \bar{c}	m	3.45
position de référence du centre de gravité	$\% \bar{c}$	0.35 \bar{c}
moment angulaire du moteur, h_{moteur}	$N.m$	216.9

ANNEXE III

CODE SOURCE MATLAB

```

%
%
% Algorithme de choix optimal du reseau de neurone identifiant le modele
% aerodynamique longitudinal du F-16.
%
% Author : Anne Girard
% Date : 17 janvier 2007
%
% Reference : rapport de la NASA (Nguyen et al., 1979)
%


---


clear all;
close all;
clc;

%charger les donnees aerodynamique dans le workspace
donneesAero;

%charger dans le workspace les donnees geometriques et de masse
donnees;

disp('Algorithme_de_choix_optimal_pour_les_reseaux_de_neurones');
disp('du_modele_aerodynamique_du_F-16')
disp('Auteur : Anne Girard')
disp('Date : 17 janvier 2007')
disp('-----');
disp('Donnees_issues_du_rapport_de_la_NASA_(Nguyen_et_al.,_1979)');
disp('-----');

%----- Choix du modele longitudinal ou lateral -----
disp('Quel modele voulez-vous identifier ?');
disp('1. le modele longitudinal');
disp('2. le modele lateral');
choixLongLat = input('Votre choix : ');

%----- Region -----
disp('Choix de la region d angle d attaque alpha : ');
disp('alpha doit etre compris entre -10 et 45 degres');
alpha_min = input('valeur minimale de la region : ');
alpha_max = input('valeur maximale de la region : ');

%----- Points statiques -----
disp('Choix du point statique :')

switch choixLongLat
    case 1
        disp('1. q = 0 degres/s');
        disp('2. q = 100 degres/s');
        choix_q = input('votre choix : ');
        switch choix_q

```

```

        case 1
            q = 0; % en rad/s
        case 2
            q = 100*pi/180; % en rad/s
    end
    % pour le developpement des reseaux :
    epsilon = 4*10^(-4);
    disp('critere_de_performance_par_defaut : 4*10^(-4)');
case 2
    disp('1. p=r=0 degres/s');
    disp('2. p=50 degres/s, r=0 degres/s');
    disp('3. p=0 degres/s, r=25 degres/s');
    choix_pr = input('votre choix : ')
    switch choix_pr
        case 1
            p = 0; r = 0; % en rad/s
        case 2
            p = 50*pi/180; r = 0; % en rad/s
        case 3
            p = 0; r = 25*pi/180; % en rad/s
    end
    % pour le developpement des reseaux :
    epsilon = 1*10^(-4);
    disp('critere_de_performance_par_defaut : 1*10^(-4)');
end
v = 150; % en m/s

%----- Modele high-fidelity du F-16 -----
% 1. creation du vecteur d'entree d'entrainement
%
% les donnees aerodynamiques contenues dans les tables sont donnees pour :
%
% vecteur d'entree et de sortie:
%
% ensemble genere par interpolation des donnees de la NASA (methode
% usuelle, ici consideree comme donnant le modele 'reel')
[beta_i, alpha_i, deltae_i] = meshgrid(-30:1:30, alpha_min:1:alpha_max, ...
-25:1:25);
[tailleAlpha_test, tailleBeta_test, tailleDeltae_test]=size(alpha_i);

switch choixLongLat
    case 1
        [CX0_I, CXlef_I, CXq_I, dCXqLef_I, Cm0_I, Cmlef_I, dCm_I, Cmq_I, ...
        dCmqLef_I, CZ0_I, CZlef_I, CZq_I, dCZqLef_I] = ...
        coeffLongInterpoles(alpha_i, beta_i, deltae_i);
        % generation aleatoire de la valeur de delta_lef pour chaque
        % entree :
        deltaLef_i = 25*rand(1, tailleAlpha_test*tailleBeta_test*...
        tailleDeltae_test);
        % compilation des donnees pour creer les vecteurs d'entree et
        % de sortie
        for i=1:tailleAlpha_test
            for j=1:tailleBeta_test
                for k=1:tailleDeltae_test
                    l = k+(j-1)*tailleDeltae_test+(i-1)*tailleDeltae_test*...
                    tailleBeta_test
                    input(:,l) = [alpha_i(i,j,k); beta_i(i,j,k); ...
                    deltae_i(i,j,k); deltaLef_i(l)];
                    target(:,l) = [CX0_I(i,j,k)+(CXlef_I(i,j)-...
                    CX0_I(i,j,26))*(1-deltaLef_i(l)/25)+(q*cbar)/(2*v)*...
                    (CXq_I(i)+dCXqLef_I(i)*(1-deltaLef_i(l)/25));
                end
            end
        end
    end

```

```

Cm0_I(i,j,k)+(CXlef_I(i,j)-Cm0_I(i,j,26))*...
(1-deltaLef_i(1)/25)+(q*cbar)/(2*v)*(Cmq_I(i)+...
dCmqLef_I(i)*(1-deltaLef_i(1)/25))+dCm_I(i)+...
(xcg_ref - xcg)*(CZO_I(i,j,k)+(CXlef_I(i,j)-...
CZO_I(i,j,26))*(1-deltaLef_i(1)/25)+(q*cbar)/(2*v)*...
(CZq_I(i)+dCZqLef_I(i)*(1-deltaLef_i(1)/25)));
CZO_I(i,j,k)+(CXlef_I(i,j)-CZO_I(i,j,26))*...
(1-deltaLef_i(1)/25)+(q*cbar)/(2*v)*(CZq_I(i)+...
dCZqLef_I(i)*(1-deltaLef_i(1)/25));
end
end
end
case 2
[C10_I,Clelf_I,Clda20_I,Clda20Lef_I,Cldr30_I,Clr_I,dClrLef_I, ...
Clp_I,dClpLef_I,dClbeta_I,CY0_I,CYlef_I,CYda20_I,CYda20Lef_I, ...
CYdr30_I,CYr_I,dCYrLef_I,CYp_I,dCYpLef_I,Cn0_I,Cnlef_I, ...
Cnda20_I,Cnda20Lef_I,Cndr30_I,Cnr_I,dCnrLef_I,Cnp_I,dCnpLef_I, ...
dCnbeta_I] = coeffLatInterpoles(alpha_i,beta_i,deltae_i);
% generation aleatoire de la valeur de delta_lef pour chaque entree :
deltaLef_i = 25*rand(1,tailleAlpha_test*tailleBeta_test*...
tailleDeltae_test);
deltaa_i = 42*rand(1,tailleAlpha_test*tailleBeta_test*...
tailleDeltae_test)-21;
deltar_i = 60*rand(1,tailleAlpha_test*tailleBeta_test*...
tailleDeltae_test)-30;
% compilation des donnees pour creer les vecteurs d'entree et de
% sortie
for i=1:tailleAlpha_test
    for j=1:tailleBeta_test
        for k=1:tailleDeltae_test
            l = k+(j-1)*tailleDeltae_test+(i-1)*tailleDeltae_test*...
tailleBeta_test
            input(:,l) = [alpha_i(i,j,k); beta_i(i,j,k); ...
deltae_i(i,j,k); deltaa_i(l); deltar_i(l); ...
deltaLef_i(l)];
            target(:,l) = [C10_I(i,j,k)+(Clelf_I(i,j)-...
C10_I(i,j,26))*(1-deltaLef_i(1)/25)+(Clda20_I(i,j)-...
C10_I(i,j,26))*(deltar_i(1)/30)+(Clda20Lef_I(i,j)-...
Clelf_I(i,j)-...
(Clda20_I(i,j)-C10_I(i,j,26)))*(1-deltaLef_i(1)/25))*...
(deltaa_i(1)/20)+(p*b)/(2*v)*(Clp_I(i)+dClpLef_I(i)*...
(1-deltaLef_i(1)/25)+(r*b)/(2*v)*(Clr_I(i)+...
dClrLef_I(i)*(1-deltaLef_i(1)/25))+dClbeta_I(i); ...
CY0_I(i,j)+(CYlef_I(i,j)-CY0_I(i,j))*...
(1-deltaLef_i(1)/25)+(CYdr30_I(i,j)-CY0_I(i,j))*...
(deltar_i(1)/30)+(CYda20_I(i,j)-CY0_I(i,j)+...
(CYda20Lef_I(i,j)-CYlef_I(i,j)-(CYda20_I(i,j)-...
CY0_I(i,j)))*(1-deltaLef_i(1)/25))*(deltaa_i(1)/20)+...
(p*b)/(2*v)*(CYp_I(i)+dCYpLef_I(i)*...
(1-deltaLef_i(1)/25)+(r*b)/(2*v)*(CYr_I(i)+...
dCYrLef_I(i)*(1-deltaLef_i(1)/25)); ...
Cn0_I(i,j,k)+(Cnlef_I(i,j)-Cn0_I(i,j,26))*...
(1-deltaLef_i(1)/25)+(Cndr30_I(i,j)-Cn0_I(i,j,26))*...
(deltar_i(1)/30)+(Cnda20_I(i,j)-Cn0_I(i,j,26)+...
(Cnda20Lef_I(i,j)-Cnlef_I(i,j)-(Cnda20_I(i,j)-...
Cn0_I(i,j,26)))*(1-deltaLef_i(1)/25))*...
(deltaa_i(1)/20)+(p*b)/(2*v)*(Cnp_I(i)+dCnpLef_I(i)*...
(1-deltaLef_i(1)/25)+(r*b)/(2*v)*(Cnr_I(i)+...
dCnrLef_I(i)*(1-deltaLef_i(1)/25))+dCnbeta_I(i)-...
(cbar/b)*(xcg_ref - xcg)*(CY0_I(i,j)+(CYlef_I(i,j)-...
CY0_I(i,j))*(1-deltaLef_i(1)/25)+(CYdr30_I(i,j)-...
CY0_I(i,j))*(deltar_i(1)/30)+(CYda20_I(i,j)-...
CY0_I(i,j)+(CYda20Lef_I(i,j)-CYlef_I(i,j)-...

```

```

        (CYda20_I(i,j)-CY0_I(i,j))*(1-deltaLef_i(1)/25))*...
        (deltaa_i(1)/20)+(p*b)/(2*v)*(CYp_I(i)+dCYpLef_I(i)*...
        (1-deltaLef_i(1)/25))+(r*b)/(2*v)*(CYr_I(i)+...
        dCYrLef_I(i)*(1-deltaLef_i(1)/25))]];
    end
end
end
end

[tailleVariables,tailleEntree]=size(input);

% melange du vecteur pour augmenter la variance entre les exemples
R = randperm(tailleEntree);
for i=1:tailleEntree
    P(:,i) = input(:,R(i));
    T(:,i) = target(:,R(i));
end

% pretraitement des donnees :
[pn,meanp,stdp,tn,meant,stdt] = prestd(P,T);

% prelever 20% pour l'ensemble de validation, 20% pour l'ensemble de test:
s = fix(tailleEntree*.6);
t = fix(tailleEntree*.8);

inputTrain = pn(:,1:s);
targetTrain = tn(:,1:s);

inputVal = pn(:,s+1:t);
targetVal = tn(:,s+1:t);
val.P=inputVal;
val.T=targetVal;

inputTest = pn(:,t+1:tailleEntree);
targetTest = tn(:,t+1:tailleEntree);
test.P=inputTest;
test.T=targetTest;

% 2. Algorithme pour reseau "optimal"
%-----
clear classes %nettoyage de la memoire

disp('Initialisation des parametres de l algorithme');
m1 = input('Choisissez un nombre initial de neurones sur la couche cachee : ');
disp('Voulez-vous modifier le critere de performance ?')
disp('1. oui');
disp('2. non');
choixEpsilon = input('votre reponse : ');
switch choixEpsilon
    case 1
        epsilon = input('entrez la valeur desiree du critere');
    case 2
        disp('les valeurs de epsilon par defaut sont utilisees');
end
N = input('Choisissez le nombre de comparaison a faire pour
la validation ...');
nbepochs = 300; %nombre maximum d'epoques d'entrainement

% initialisation des variables de stockage
performance = 10;
reseaux = [];

```

```

perfReseau = [];
perfres = 1;

% tant que le critere d'arret n'est pas verifie :
while (performance>epsilon)
    %initialisation de la variable perfM1 locale a chaque boucle
    %(ie. associee a la valeur de m1)
    perfM1 = 10;
    % on entraîne un reseau a m1 neurones sur la couche cachee donnee
    % fait N fois, car l'initialisation influence la performance
    for i=1:N
        % initialisation des variables de performance locale
        perflocale = 100;
        perfentraimentprecedent=10;

        % validation croisee pour determiner le nombre d'epoque
        % d'entraînement:
        while (perflocale > performance)
            % si la performance du reseau obtenu est superieure a la
            % performance obtenue pour m1 - 1 neurones sur la couche cachee, le
            % reseau est mis de cte et un nouveau est entraîne.

            % creation d'un reseau a m1 neurones sur la couche cachee
            net = newff(minmax(inputTrain),[m1 3],{'tansig' 'purelin'},...
            'trainlm');
            % il est entraîne une epoque a la fois :
            net.trainParam.epochs = 1;
            % compteurs et variables d'arret d'entraînement :
            j = 1;
            cont = 0;
            compteur = 0;
            % tant que le maximum d'epoques d'entraînement n'est pas
            % atteint et que la performance du reseau continue de diminuer
            % celui-ci est entraîne
            while( j < (nbepochs+1) & cont == 0)
                [net,tr]=train(net,inputTrain,targetTrain,[],[],val,test);

                % si la performance de l'ensemble de validation ne change
                % pas, alors le compteur est incremente de 1.
                if (tr.vperf(1,2) >= perfentraimentprecedent)
                    compteur = compteur + 1;
                else compteur = 0;
                end
                % au bout de cinq epoques o la performance ne change pas,
                % l'entraînement prend fin
                if (compteur == 10)
                    cont = 1;
                    display('entraînement_arrete_prematurement');
                end

                %si la performance de l'ensemble de test est plus faible
                %que pour l'epoque precedente, le reseau est sauvegarde
                if (tr.tperf(1,2) < perflocale)
                    res = net;
                    epoq = j;
                    perflocale = mse(poststd(sim(res,pn),meant,stdt)-T);
                    display('reseau_plus_performant_a_m1_donne');
                end
                perfentraimentprecedent = tr.vperf(1,2);
                j=j+1;
            end % on sort de la boucle d'entraînement

        end % on sort de la boucle donc : perflocale <=performance
    end

```

```
% le reseau N a m1 neurones est :  
    resN = res;  
    perfres = perflocale;  
% on choisit le meilleur :  
if (perfres < perfM1)  
    display('choix_du_reseau_a_ml_neurones');  
    resM1 = resN;  
    perfM1 = perfres;  
end  
end  
  
% on les compile pour les garder en memoire :  
reseaux = [reseaux resM1];  
perfReseau = [perfReseau perfM1];  
performance = perfM1;  
% le nombre de neurones sur la couche cachee est augmentee de 1 :  
m1 = m1 + 1;  
  
% le workspace est sauvegarde  
save('reseauxIdentification')  
  
end
```

ANNEXE IV

COMMANDE PAR LINÉARISATION EXACTE

La linéarisation exacte consiste à trouver une loi de commande non linéaire qui permet d'annuler les non linéarités de la dynamique d'un système et de transformer celui-ci en un système linéaire.

On distingue deux méthodes de linéarisation exacte : la linéarisation entrée-état et celle entrée-sortie. Le résumé des notes de cours de M. Desantis (2006) ci-dessous explique ces deux linéarisations.

IV.1 Linéarisation entrée-état

La méthodologie est expliquée dans le cas de systèmes à entrée unique et sortie unique (SISO), mais peut aisément être étendue aux systèmes à entrées multiples et sorties multiples (MIMO).

Considérons le système non linéaire

$$\dot{x} = f(x) + g(x)u \quad (\text{IV.1})$$

où $x \in \mathbb{R}^n$ et $u \in \mathbb{R}$ et les fonctions f et g sont lisses dans un domaine $D \subset \mathbb{R}^n$.

Définition 1 *Le système IV.1 est dit linéarisable entrée-état s'il existe une région Ω de \mathbb{R}^n , un difféomorphisme $\Phi : \Omega \rightarrow \mathbb{R}^n$ et un retour d'état*

$$u = \alpha(x) + \beta(x)v$$

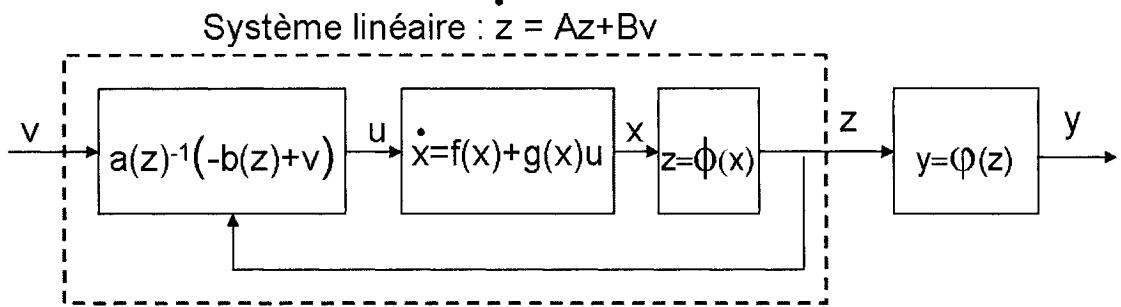


Figure IV.1 Linéarisation entrée-état

tel que le nouveau vecteur d'état $z = \Phi(x)$ et la nouvelle entrée v vérifient la relation

$$\dot{z} = Az + Bv$$

Le schéma bloc du système linéarisé est donné par la figure IV.1. Certaines conditions doivent cependant être satisfaites pour que le système soit linéarisable par un tel retour statique.

Théorème 1 *Le système IV.1 est linéarisable entrée-état, dans une région Ω de \mathbb{R}^n , par retour d'état statique si les deux conditions suivantes sont satisfaites :*

- $r = \text{rang}(g, \text{ad}_f g, \dots, \text{ad}_f^{n-1} g) = n, \forall x \in \Omega$
- $\text{vect}(g, \text{ad}_f g, \dots, \text{ad}_f^{n-1} g)$ est involutive sur Ω

$\text{ad}_f g$ est défini par les crochets de Lie tel que :

$$L_g L_f^j h(x) = L_{\text{ad}_f^j g} h(x) \quad (\text{IV.2})$$

Si ces conditions sont satisfaites alors on a

$$\begin{aligned} z &= [\Phi_1(x), L_f \Phi_1(x), \dots, L_f^{n-1} \Phi_1(x)]^T \\ \alpha(x) &= -\frac{L_f^n \Phi_1(x)}{L_g L_f^{n-1} \Phi_1(x)} \\ \beta(x) &= \frac{1}{L_g L_f^{n-1} \Phi_1(x)} \end{aligned} \quad (\text{IV.3})$$

où $\Phi_1(x)$ est solution du système d'équations suivant :

$$\begin{aligned} L_g L_f^i \Phi_1(x) &= 0 \text{ pour } i = 0, \dots, n-2 \\ L_g L_f^{n-1} \Phi_1(x) &\neq 0 \end{aligned} \quad (\text{IV.4})$$

IV.2 Linéarisation entrée-sortie

Considérons le système suivant, non linéaire MIMO et affine en la commande :

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \quad (\text{IV.5})$$

où $x \in \mathbb{R}^n$ représente les états, $u \in \mathbb{R}$ le contrôle et $y \in \mathbb{R}^p$ les sorties. Les fonctions f et g sont lisses dans un domaine $D \subset \mathbb{R}^n$.

Le degré relatif r de ce système est alors déterminé.

Définition 2 Soit Ω un voisinage de x_0 .

Un système MIMO est dit de degré relatif r_1, r_2, \dots, r_p au point x_0 si

$$\forall x \in \Omega \quad L_{g_j} L_f^k h_i(x) = 0 \quad 1 \leq j \leq m, \quad 1 \leq i \leq p, \quad 0 \leq k \leq r_i - 1$$

et

$$A(x_0) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & \dots & L_{g_m} L_f^{r_2-1} h_2(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_p-1} h_p(x) & \dots & L_{g_m} L_f^{r_p-1} h_p(x) \end{bmatrix}$$

est inversible.

Le degré relatif total d'un système MIMO est alors $r = \sum_{i=1}^p r_i$.

Le difféomorphisme Φ transforme le système original non linéaire et affine en la commande en un système linéaire sous la forme canonique de Brunovsky :

$$z = \Phi(x) = \begin{bmatrix} z_{11} \\ z_{12} \\ \vdots \\ z_{1r_1} \\ z_{21} \\ \vdots \\ z_{pr_p} \end{bmatrix} = \begin{bmatrix} h_1(x) \\ L_f h_1(x) \\ \vdots \\ L_f^{r_1-1} h_1(x) \\ h_2(x) \\ \vdots \\ L_f^{r_p-1} h_p(x) \end{bmatrix} \quad (\text{IV.6})$$

$$\dot{z} = \frac{d}{dt} \begin{bmatrix} z_{11} \\ z_{12} \\ \vdots \\ z_{1(r_1-1)} \\ z_{1r_1} \\ z_{21} \\ \vdots \\ z_{pr_p} \end{bmatrix} = \begin{bmatrix} z_{12} \\ z_{13} \\ \vdots \\ z_{1r_r} \\ L_f^{r_1} h_1(x) + \sum_{k=1}^m L_{g_k} L_f^{r_1-1} h_1(x) u_k \\ z_{22} \\ \vdots \\ L_f^{r_p} h_p(x) + \sum_{k=1}^m L_{g_k} L_f^{r_p-1} h_p(x) u_k \end{bmatrix} \quad (\text{IV.7})$$

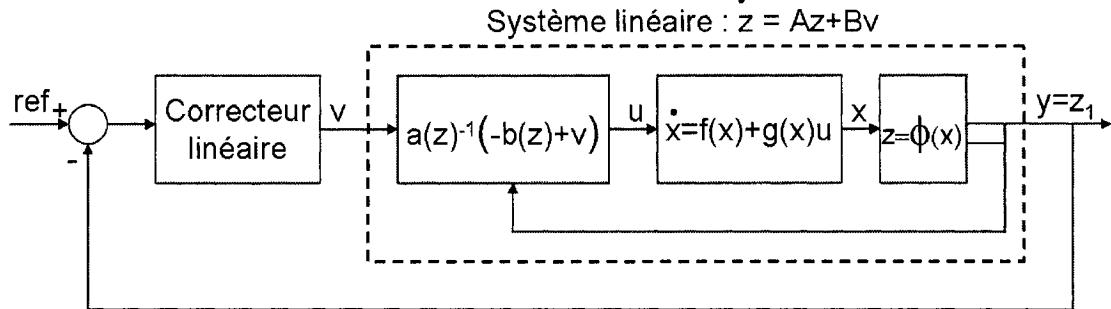


Figure IV.2 Linéarisation entrée-sortie

Le retour linéarisant résultant peut alors être défini par :

$$u = A^{-1}(v - B) \quad (\text{IV.8})$$

où

$$\begin{aligned} A &= \{A_{ij}\} = \{L_{g_j} L_f^{r_i-1} h_i(x)\} \\ B &= \{B_i\} = \{L_f^{r_i} h_i(x)\} \end{aligned} \quad (\text{IV.9})$$

Théorème 2 *Un système non linéaire est complètement linéarisable entrée-sorties dans un domaine $D_0 \in D$ par retour d'état statique si et seulement si le degré relatif est égal à la dimension de l'espace d'entrée pour tout $x \in D_0$.*

Cette méthode de linéarisation est une méthode relativement simple. Une fois le système linéarisé par retour d'état, une correction classique peut être utilisée en cascade, constituant ainsi une boucle externe (voir IV.2).

ANNEXE V

COMMANDÉ PRÉDICTIVE LINÉAIRE

La commande prédictive repose sur les grandes idées suivantes (Dumur, 2002) :

- la création d'un effet anticipatif qui exploite les connaissances sur l'évolution des trajectoires à suivre (connaissance au moins sur quelques points au delà de l'instant présent)
- la minimisation d'un critère (ici quadratique à horizon fini) sur les erreurs de prédiction futures
- l'élaboration d'une séquence de commandes futures, optimale au sens du critère précédent, dont seule la première valeur est appliquée au système

V.1 Formulation du problème

L'ensemble de cette section est basée sur la formulation de Maciejowsky (2001).

Soit le système à commander suivant:

$$\begin{aligned}\dot{x} &= f(x, r) \\ \bar{y} &= h(x, r)\end{aligned}\tag{V.1}$$

où le vecteur d'état $x \in \mathbb{R}^n$, le vecteur de contrôle $r \in \mathbb{R}^m$ et la sortie $\bar{y} \in \mathbb{R}^p$.

Celui-ci est alors linéarisé et discréétisé. A chaque période d'échantillonnage k , on

obtient :

$$\begin{aligned} x(k+1) &= A_k x(k) + B_k r(k) \\ \begin{bmatrix} y(k) \\ z(k) \\ u(k) \end{bmatrix} &= C_k x(k) + D_k r(k) \end{aligned} \quad (\text{V.2})$$

où $y \in \mathbb{R}^{p_1}$ représente les sorties à réguler, $z \in \mathbb{R}^{p_2}$ les sorties liées aux variables d'état qui possèdent des contraintes et $u \in \mathbb{R}^{p_3}$ les sorties liées aux actionneurs qui possèdent des contraintes en amplitude et en vitesse ; $p_1 + p_2 + p_3 = p$, $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{n \times m}$, $C_k \in \mathbb{R}^{p \times n}$ et $D_k \in \mathbb{R}^{p \times m}$.

Le problème d'optimisation est légèrement modifié en choisissant le critère de performance basé sur la fonction de coût quadratique suivante (Keviczky & Balas, 2006) :

$$\begin{aligned} J(k) &= \sum_{i=1}^{H_p} (\hat{y}(k+i|k) - y_{ref}(k+i|k))^T Q (\hat{y}(k+i|k) - y_{ref}(k+i|k)) \\ &+ \sum_{i=0(\delta_{H_c})}^{H_c-1} \Delta r(k+i|k)^T R \Delta r(k+i|k) + \rho \epsilon \end{aligned} \quad (\text{V.3})$$

où $\hat{y}(k+i|k)$ est la prédiction des sorties à i instants après le temps présent k , basée sur les données jusqu'au temps k ; H_p est l'horizon de prédiction, H_c l'horizon de contrôle et δ_{H_c} la période de changement du contrôle.

ϵ représente un état appelé “slack state”. Il permet de transformer des contraintes discontinues en contraintes plus souples. ρ est le gain associé à cet état.

Afin d'obtenir les prédictions, un modèle de prédiction est nécessaire. L'utilisation d'un modèle linéaire conduit à transformer le problème d'optimisation du critère $J(k)$ en un problème de programmation quadratique, pour lequel il existe des algorithmes rapides et fiables. L'état du système est augmenté par m intégrateurs

afin de convertir le changement sur la commande Δr en la valeur de la commande r . La commande prédictive requiert la connaissance parfaite du modèle du système à asservir ; afin de réduire l'impact d'une erreur de modélisation, un modèle simple de perturbations est incorporé. Celui-ci suppose que des perturbations constantes agissent sur les sorties y et z .

Le modèle de prédiction est alors :

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \\ r(k) \end{bmatrix} = \begin{bmatrix} A_k & 0_{n \times (p_1+p_2)} & B_k \\ 0_{(p_1+p_2) \times n} & I_{p_1+p_2} & 0_{(p_1+p_2) \times m} \\ 0_{m \times n} & 0_{m \times (p_1+p_2)} & I_m \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix} + \begin{bmatrix} B_k \\ 0_{(p_1+p_2) \times m} \\ I_m \end{bmatrix} \Delta r(k)$$

$$\begin{bmatrix} \hat{y}(k) \\ \hat{z}(k) \\ \hat{u}(k) \end{bmatrix} = \begin{bmatrix} C_k & \left| \begin{array}{c} I_{p_1+p_2} \\ 0_{p_3 \times (p_1+p_2)} \end{array} \right| & D_k \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix} + D_k \Delta r(k)$$
(V.4)

Afin de simplifier la notation, il est réécrit sous la forme suivante :

$$\begin{aligned} \hat{\xi}(k+1) &= \overline{A_k} \hat{\xi}(k) + \overline{B_k} \Delta r(k) \\ \hat{w}(k) &= \overline{C_k} \hat{\xi}(k) + \overline{D_k} \Delta r(k) \end{aligned} \quad (V.5)$$

V.2 Le problème de programmation quadratique équivalent

Le modèle de prédiction permet de calculer les prédictions utilisées par le critère de performance.

$$\begin{aligned}
& \begin{bmatrix} \hat{w}(k+1|k) \\ \hat{w}(k+2|k) \\ \vdots \\ \hat{w}(k+H_c|k) \\ \hat{w}(k+H_c+1|k) \\ \vdots \\ \hat{w}(k+H_p|k) \end{bmatrix} = \begin{bmatrix} \overline{C_k A_k} \\ \overline{C_k A_k}^2 \\ \vdots \\ \overline{C_k A_k}^{H_c} \\ \overline{C_k A_k}^{H_c+1} \\ \vdots \\ \overline{C_k A_k}^{H_p} \end{bmatrix} \hat{\xi}(k) \\
& + \begin{bmatrix} \overline{C_k B_k} & \overline{D_k} & 0 & \dots & 0 \\ \overline{C_k A_k B_k} & \overline{C_k B_k} & \overline{D_k} & \dots & 0 \\ \vdots & \vdots & \ddots & & \ddots \\ \overline{C_k A_k}^{H_c-1} \overline{B_k} & \overline{C_k A_k}^{H_c-2} \overline{B_k} & \dots & & \overline{C_k B_k} \\ \overline{C_k A_k}^{H_c} \overline{B_k} & \overline{C_k A_k}^{H_c-1} \overline{B_k} & \dots & & \overline{C_k A_k B_k} \\ \vdots & \vdots & \ddots & & \vdots \\ \overline{C_k A_k}^{H_p-1} \overline{B_k} & \overline{C_k A_k}^{H_p-2} \overline{B_k} & \dots & & \overline{C_k A_k}^{H_p-H_c} \overline{B_k} \end{bmatrix} \begin{bmatrix} \Delta r(k|k) \\ \Delta r(k+1|k) \\ \vdots \\ \Delta r(k+H_c-1|k) \end{bmatrix}
\end{aligned} \tag{V.6}$$

Cette relation est réécrite avec des notations compactes :

$$\overline{W}(k) = \overline{Aa_k} \hat{\xi}(k) + \overline{Bb_k} \Delta \overline{R}(k) \tag{V.7}$$

Le critère de performance devient alors :

$$\begin{aligned}
J(k) &= \Delta \overline{R}(k)^T \overline{Bb_{k_y}}^T \overline{Q} \overline{Bb_{k_y}} \Delta \overline{R}(k) + \Delta \overline{R}(k)^T 2 \overline{Bb_{k_y}}^T \overline{Q} \left(\overline{Aa_{k_y}} \hat{\xi}(k) - \overline{Y_{ref}}(k) \right) \\
&+ \left(\overline{Aa_{k_y}} \hat{\xi}(k) - \overline{Y_{ref}}(k) \right)^T \overline{Q} \left(\overline{Aa_{k_y}} \hat{\xi}(k) - \overline{Y_{ref}}(k) \right) + \Delta \overline{R}(k)^T \overline{R} \Delta \overline{R}(k) + \rho \epsilon
\end{aligned} \tag{V.8}$$

où $\overline{Y_{ref}}(k) = \left[y_{ref}(k+1), \dots, y_{ref}(k+H_p) \right]^T$, l'indice y indique que seules les lignes associées aux sorties y sont sélectionnées, \overline{Q} (respectivement \overline{R}) est une ma-

trice avec Q (respectivement R) sur sa diagonale.

Keviczky et Balas (2006) distinguent deux types de contraintes : celles liées aux actionneurs, exprimées sur u , et celles liées aux manœuvres de l'aéronef, exprimées sur z .

Les contraintes en amplitude et en vitesse de rotation sur les actionneurs s'expriment comme des contraintes dites “dures” :

$$\underline{u}_k \leq \hat{u}(k+1|k), \dots, \hat{u}(k+H_p|k) \leq \bar{u}_k \quad (\text{V.9})$$

Les contraintes sur certaines sorties liées aux manœuvres de l'avion doivent être exprimées comme des contraintes “molles”, sinon les perturbations et les erreurs de modélisation pourraient impliquer que le problème n'a pas de solution. À cette fin, la variable additionnelle ϵ est introduite :

$$\underline{z}_k - \epsilon \leq \hat{z}(k+1|k), \dots, \hat{z}(k+H_p|k) \leq \bar{z}_k + \epsilon \quad (\text{V.10})$$

Afin que les contraintes originelles “dures” ne soient pas transgresser inutilement, le gain ρ doit être choisi suffisamment grand.

L'équation V.6 permet d'exprimer les contraintes sur z en fonction de $\Delta R(k)$. En effet $\bar{Z}(k) = [\hat{z}(k+1|k), \dots, \hat{z}(k+H_p|k)]^T$ est régit par l'équation suivante :

$$\bar{Z}(k) = \overline{Aa_{k_z}} \hat{\xi}(k) + \overline{Bb_{k_z}} \Delta R(k) \quad (\text{V.11})$$

où l'indice z indique que seules les lignes associées aux sorties z ont été sélectionnées.

La contrainte V.10 devient alors :

$$\begin{aligned} -\overline{Bb_{k_z}}\Delta\overline{R}(k) - \epsilon &\leq -\underline{z}_k + \overline{Aa_{k_z}}\hat{\xi}(k) \\ \overline{Bb_{k_z}}\Delta\overline{R}(k) - \epsilon &\leq \overline{z}_k - \overline{Aa_{k_z}}\hat{\xi}(k) \end{aligned} \quad (\text{V.12})$$

De manière analogue on peut réécrire les contraintes sur u :

$$\begin{aligned} -\overline{Bb_{k_u}}\Delta\overline{R}(k) &\leq -\underline{u}_k + \overline{Aa_{k_u}}\hat{\xi}(k) \\ \overline{Bb_{k_u}}\Delta\overline{R}(k) &\leq \overline{u}_k - \overline{Aa_{k_u}}\hat{\xi}(k) \end{aligned} \quad (\text{V.13})$$

Le problème de programmation quadratique équivalent à résoudre à chaque itération en fonction de $\Delta R(k)$ et ϵ est le suivant:

$$\begin{aligned} \min_{\Delta R(k), \epsilon} & \left\{ \Delta\overline{R}(k)^T \overline{Bb_{k_y}}^T \overline{Q} \overline{Bb_{k_y}} \Delta\overline{R}(k) + \Delta\overline{R}(k)^T 2\overline{Bb_{k_y}}^T \overline{Q} \left(\overline{Aa_{k_y}}\hat{\xi}(k) - \overline{Y_{ref}}(k) \right) \right. \\ & \left. + \left(\overline{Aa_{k_y}}\hat{\xi}(k) - \overline{Y_{ref}}(k) \right)^T \overline{Q} \left(\overline{Aa_{k_y}}\hat{\xi}(k) - \overline{Y_{ref}}(k) \right) + \Delta\overline{R}(k)^T \overline{R} \Delta\overline{R}(k) + \rho\epsilon \right\} \\ \text{avec} & \begin{bmatrix} -\overline{Bb_{k_u}} & 0 \\ \overline{Bb_{k_u}} & 0 \\ -\overline{Bb_{k_z}} & -1 \\ \overline{Bb_{k_z}} & -1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta R(k) \\ \epsilon \end{bmatrix} \leq \begin{bmatrix} -\underline{u}_k + \overline{Aa_{k_u}}\hat{\xi}(k) \\ \overline{u}_k - \overline{Aa_{k_u}}\hat{\xi}(k) \\ -\underline{z}_k + \overline{Aa_{k_z}}\hat{\xi}(k) \\ \overline{z}_k - \overline{Aa_{k_z}}\hat{\xi}(k) \\ 0 \end{bmatrix} \end{aligned} \quad (\text{V.14})$$

ANNEXE VI

SIMULATEUR NON LINÉAIRE DU F-16

Un simulateur non linéaire du F-16 a été développé sous MatLab - Simulink®. L'architecture générale de celui-ci est illustrée par le schéma de la figure VI.1. Chacun des sous-systèmes le constituant est explicité ci-dessous.

VI.1 Les variables d'état du F-16

Les variables d'état usuelles d'un aéronef sont $[V_t, \alpha, \beta, \phi, \theta, \psi, p, q, r, x_E, y_E, h]$. La modélisation du moteur du F-16 introduit une variable d'état supplémentaire P_a , et celle de la dynamique des actionneurs quatre variables d'état supplémentaires $\delta_{e_{com}}, \delta_{a_{com}}, \delta_{r_{com}}$ et $\delta_{lef_{com}}$. De plus le calcul de la valeur de δ_{lef} en induit un autre $\delta_{lef_{calcul}}$. Le vecteur d'état du F-16 est ainsi :

$$[V_t, \alpha, \beta, \phi, \theta, \psi, p, q, r, x_E, y_E, h, \delta_{e_{com}}, \delta_{lef_{com}}, P_a, \delta_{r_{com}}, \delta_{a_{com}}, \delta_{lef_{calcul}}].$$

VI.2 Les équations du mouvement

Les équations du mouvement présentées à l'annexe I sont implémentées dans les sous-systèmes en jaune de la figure VI.1. On remarque que les équations donnant $\dot{\phi}$ et $\dot{\psi}$ peuvent présenter une singularité lorsque $\theta = \pm 90^\circ$. Une solution consisterait à utiliser les quaternions plutôt que les angles d'Euler. Cependant, cela implique que les angles d'Euler ne sont plus directement accessibles. On préfère donc ici ajouter un sous-système permettant d'obvier à cette difficulté en s'inspirant de la solution proposée par Klein et Morelli : lorsque θ est compris en valeur absolue

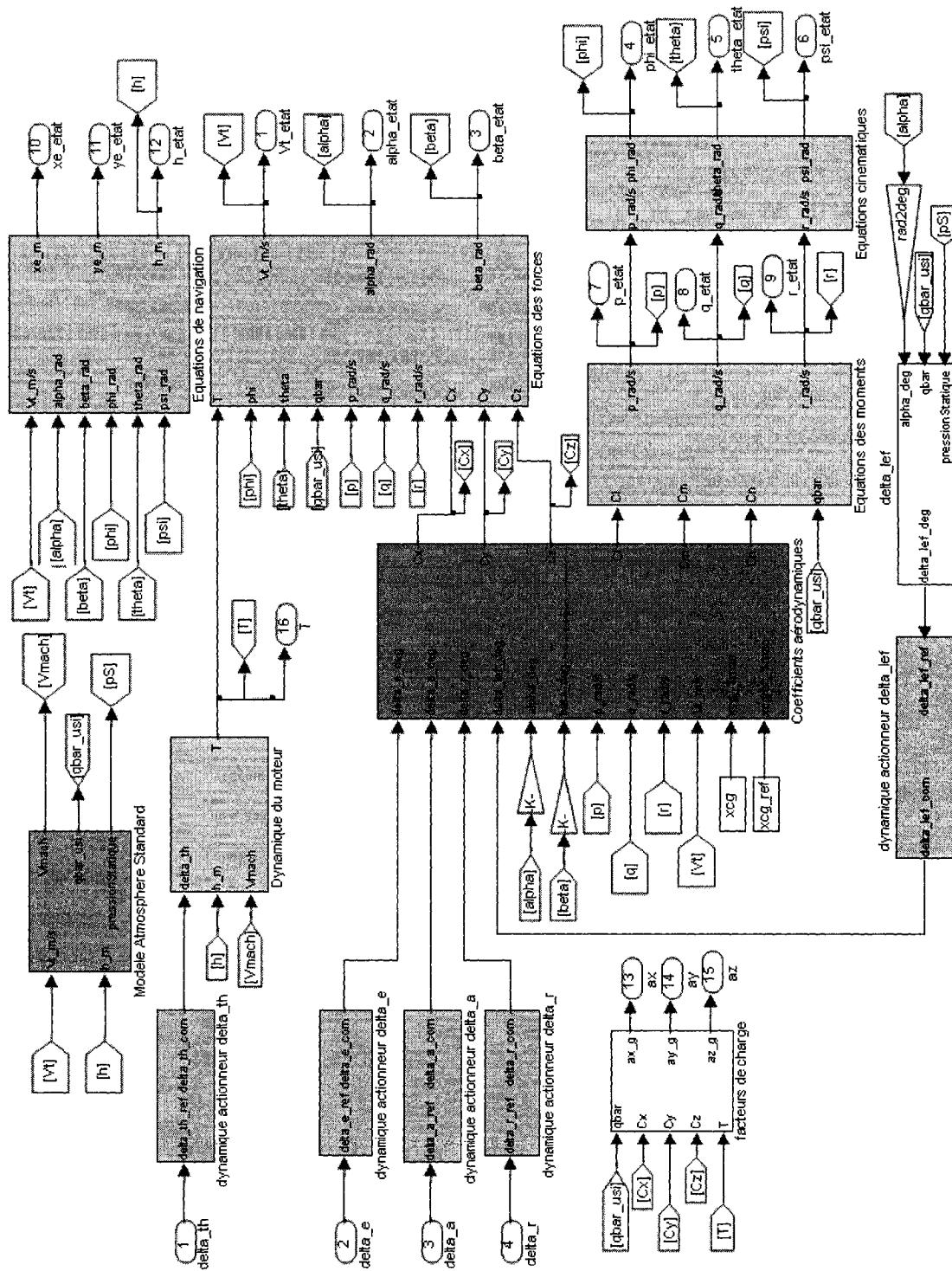


Figure VI.1 Simulateur

entre 89.99° et 90.01° , la valeur utilisée pour le calcul de $\dot{\phi}$ et $\dot{\psi}$ est soit 89.99° soit 90.01° , selon la valeur de q .

VI.3 Le modèle aérodynamique

Le modèle aérodynamique utilisé dépend de la simulation à effectuer, selon qu'il s'agisse du comportement du F-16 “réel” ou du F-16 identifié.

VI.3.1 Pour la simulation du comportement “réel” du F-16

Les données aérodynamiques proviennent d'essais en soufflerie statiques et dynamiques à faible vitesse sur un modèle réduit à 16% du F-16 (Nguyen et al, 1979). Les données sont fournies sous forme de tableaux pour un large spectre d'angle d'attaque et d'angle de dérapage : $-20^\circ < \alpha < 90^\circ$ et $-30^\circ < \beta < 30^\circ$. Pour obtenir une approximation de la valeur des données pour un point ne figurant pas dans les tables, il convient d'interpoler les données connues. Dans le cas du F-16, c'est l'interpolation linéaire qui est classiquement utilisée pour simuler le système réel; cependant il existe d'autres méthodes d'interpolation qui peuvent se prouver plus efficaces. Sachant que ces coefficients interviennent dans l'équation d'état qu'il va falloir intégrer et que les états sont continus, les méthodes d'interpolation donnant des résultats plus lisses seront préférées.

Interpolation du plus proche voisin : La valeur de la fonction en un point donné est égale à celle de son plus proche voisin dans l'espace des variables d'entrées. Bien que cette méthode soit rapide, elle est coûteuse en terme de mémoire. Le résultat de l'interpolation est une fonction constante par morceau qui ne satisfait pas les caractéristiques de continuité désirées.

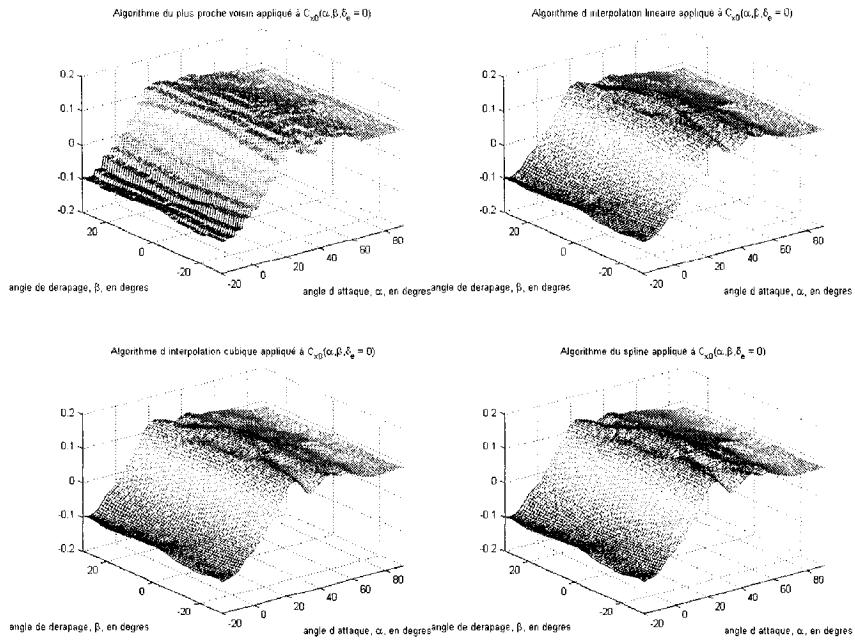


Figure VI.2 Comparaison des différents algorithmes d'interpolation en 2D

Interpolation lin   aire (ou bilin   aire, selon le nombre de param   tres) :

La valeur de la fonction en un point est calcul   e sous l'hypoth   se que celle-ci varie lin   airement entre deux points de l'ensemble des donn   es. Cet algorithme n   cessite moins de m   moire que le précédent, mais l   g   rement plus de temps de calcul. Le r   sultat de l'interpolation est une fonction continue et d   ivable par morceau.

Interpolation cubique (ou bicubique, selon le nombre de param   tres):

Cet algorithme construit une fonction polynomiale par morceau, de sorte que sa d   riv   e soit continue. La d   riv   e seconde peut ne pas    tre continue, elle peut effectivement faire des sauts aux points de l'ensemble des donn   es. Cette m   thode est plus coûteuse     la fois en m   moire et en temps de calcul que les deux pr   c   dentes, cependant le r   sultat de l'interpolation est une fonction continue et de d   riv   e continue.

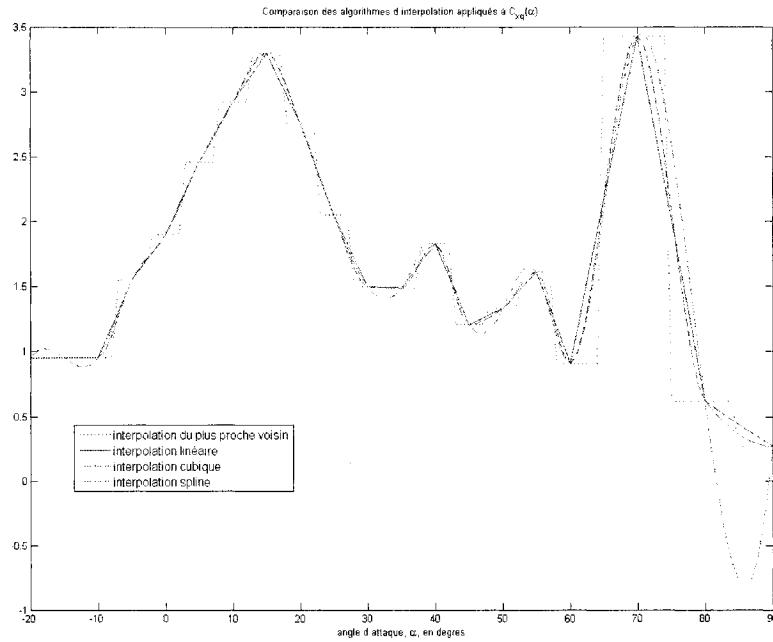


Figure VI.3 Comparaison des différents algorithmes d'interpolation en 1D

Interpolation du spline cubique : Cet algorithme ressemble à l'interpolation cubique, une approximation par un polynôme d'ordre 3 est faite entre chaque point de l'ensemble de données consécutifs, la dérivée seconde de ce polynôme est nulle en ces points. Cette méthode prend plus de temps de calcul que l'interpolation cubique, mais nécessite moins de mémoire, et le résultat de l'interpolation est une fonction continue deux fois dérivable et de dérivée seconde continue.

Les considérations évoquées en début de paragraphe font que l'algorithme d'interpolation est à choisir entre l'interpolation cubique et le spline cubique. Pour les coefficients d'amortissement, où l'interpolation est à une dimension il n'y a que peu de différence entre ces deux algorithmes. Cependant on peut craindre que le spline entraîne un sur-apprentissage des données comme le montre la figure VI.2. L'interpolation en deux dimensions confirme ce résultat (figure VI.3). L'algorithme privilégié est donc celui de l'interpolation cubique.

VI.3.2 Pour la simulation du F-16 identifié

Dans ce cas-ci, le modèle aérodynamique utilisé est le module d'identification à base de réseaux de neurones, dont l'architecture a été établie au chapitre 3.

VI.4 Le modèle du moteur

La puissance commandée est fonction de la position de la manette des gaz δ_{th} . Cette fonction transforme la position de δ_{th} , comprise entre 0 et 1, en un niveau de puissance, compris entre 0 et 100 :

$$P_c(\delta_{th}) = \begin{cases} 64, 94.\delta_{th} & \text{si } \delta_{th} \leq 0.77 \\ 217, 38.\delta_{th} - 117, 38 & \text{si } \delta_{th} > 0.77 \end{cases} \quad (\text{VI.1})$$

La dynamique du moteur est modélisée par une constante de temps du premier ordre entre la réponse en puissance réelle du moteur et la puissance commandée :

$$\dot{P}_a = \frac{1}{\tau_{moteur}} (P_c - P_a) \quad (\text{VI.2})$$

où

$$P_c = \begin{cases} P_c(\delta_{th}) & \text{si } P_c(\delta_{th}) \geq 50 \text{ et } P_a \geq 50 \\ 60 & \text{si } P_c(\delta_{th}) \geq 50 \text{ et } P_a < 50 \\ 40 & \text{si } P_c(\delta_{th}) < 50 \text{ et } P_a \geq 50 \\ P_c(\delta_{th}) & \text{si } P_c(\delta_{th}) < 50 \text{ et } P_a < 50 \end{cases} \quad (\text{VI.3})$$

$$\frac{1}{\tau_{moteur}} = \begin{cases} 5 & \text{si } P_c(\delta_{th}) \geq 50 \text{ et } P_a \geq 50 \\ \frac{1}{\tau_{moteur}^*} & \text{si } P_c(\delta_{th}) \geq 50 \text{ et } P_a < 50 \\ 5 & \text{si } P_c(\delta_{th}) < 50 \text{ et } P_a \geq 50 \\ \frac{1}{\tau_{moteur}^*} & \text{si } P_c(\delta_{th}) < 50 \text{ et } P_a < 50 \end{cases}$$

$$\frac{1}{\tau_{moteur}^*} = \begin{cases} 1 & \text{si } (P_c(\delta_{th}) - P_a) \leq 25 \\ 0,1 & \text{si } (P_c(\delta_{th}) - P_a) \geq 50 \\ 1,9 - 0,036(P_c(\delta_{th}) - P_a) & \text{si } 25 < (P_c(\delta_{th}) - P_a) < 50 \end{cases}$$

La poussée du moteur est ensuite calculée par l'équation VI.4, où T_{idle} , T_{mil} et T_{max} sont obtenues par interpolation linéaire des données fournies sous forme de table pour $0 \leq h \leq 15240m$ et $0 \leq M \leq 1$ dans le rapport de la NASA (Nguyen et al, 1979).

$$T = \begin{cases} T_{idle} + (T_{mil} - T_{idle}) \frac{P_a}{50} & \text{si } P_a < 50 \\ T_{mil} + (T_{max} - T_{mil}) \frac{P_a - 50}{50} & \text{si } P_a \geq 50 \end{cases} \quad (\text{VI.4})$$

VI.5 Le modèle de l'atmosphère

Le modèle de l'Atmosphère utilisé dans le rapport de la NASA (Nguyen et al, 1979) est le modèle standard US de 1976. Celui-ci fournit le nombre de Mach, la pression dynamique et la pression statique en unités impériales. Il a été transcrit ici, de sorte à fournir ces grandeurs en unités standard international.

La température T , en K , à une altitude h , en m , est :

$$T = T_0 - 0,0065h \quad \text{si } h < 11000m$$

$$T = T_0 - 0,0065 \cdot 11000 \quad \text{si } h \geq 11000m \quad (\text{VI.5})$$

où T_0 est la température moyenne au niveau de la mer, soit $288,15K$.

La vitesse du son a , en $m.s^{-1}$, est :

$$a = \sqrt{\gamma RT} \quad (\text{VI.6})$$

où γ est le coefficient adiabatique de l'air ($\gamma = 1,402$ à $P_0 = 1atm = 101325Pa$ et $T = T_0$) et R est la constante spécifique de l'air ($R = 287,05 \text{ J.kg}^{-1}.\text{K}^{-1}$).

Le nombre de Mach M est alors :

$$M = \frac{V}{a} \quad (\text{VI.7})$$

La masse volumique ρ , en $kg.m^{-3}$, est :

$$\rho = \rho_0 e^{\left(-\frac{g}{RT}h\right)} \quad (\text{VI.8})$$

La pression dynamique \bar{q} , en Pa , est alors :

$$\bar{q} = \frac{1}{2} \rho V^2 \quad (\text{VI.9})$$

et la pression statique P_s , en Pa , est :

$$P_s = P_0 \frac{\rho}{\rho_0} \quad (\text{VI.10})$$

VI.6 La dynamique des actionneurs

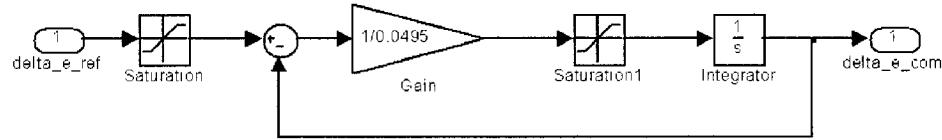


Figure VI.4 Dynamique des actionneurs

Les actionneurs sont modélisés par un premier ordre, tel que le montre la figure VI.4.

Le tableau VI.1 récapitule les limites de leur dynamique.

Tableau VI.1 Dynamique des actionneurs

Actionneur	déflexion maximale	taux de rotation limite	constante de temps T_e
Élévateur	$\pm 25^\circ$	$60^\circ.s^{-1}$	0,0495s
Ailerons	$\pm 21,5^\circ$	$80^\circ.s^{-1}$	0,0495s
Gouvernail	$\pm 30^\circ$	$120^\circ.s^{-1}$	0,0495s
Bord d'attaque	$0 - 25^\circ$	$25^\circ.s^{-1}$	0,16s

VI.7 Les conditions de vol de l'avion

La condition de vol est dite à l'équilibre lorsque toutes les forces et tous les moments dans le repère avion sont constants. Cela implique que les angles aérodynamiques et les taux de rotation sont constants, et donc que leurs dérivées soient nulles :

$$\dot{p} = \dot{q} = \dot{r} = \dot{V}_t = \dot{\alpha} = \dot{\beta} = 0 \quad (\text{VI.11})$$

On distingue ensuite plusieurs manœuvres, qui imposent chacune des contraintes spécifiques.

Vol à l'équilibre avec ailes horizontales :

$$\dot{\phi} = \dot{\psi} = \dot{\theta} = 0 \quad (\text{VI.12})$$

Virage coordonné où $\dot{\psi}$ est le taux de rotation constant :

$$\dot{\phi} = \dot{\theta} = 0 \quad (\text{VI.13})$$

Montée stabilisée où $\dot{\theta}$ est le taux de montée constant :

$$\dot{\phi} = \dot{\psi} = 0 \quad (\text{VI.14})$$

Roulis stabilisé où $\dot{\phi}$ est le taux de roulis constant :

$$\dot{\theta} = \dot{\psi} = 0 \quad (\text{VI.15})$$

Ces conditions permettent de calculer, pour une vitesse et une altitude désirées, les valeurs des variables d'état et des entrées qui assurent de rester à cette condition. Le script MatLab ci-dessous effectue ce calcul.

```

%
%----- Simulateur non lin aire du F-16 -----
%
% Auteur : Anne Girard
% Date : 9 mars 2007
%
% Ce fichier permet de "trimmer" le modele non lin aire du F-16 la
% condition de vol (vitesse, altitude) desiree et selon la manoeuvre
% desiree
%
%
%----- clear all
%----- close all

% charger dans le workspace les donnees aerodynamiques et geometriques du
% F-16 :
donneesAero;
donnees;

load RN_identification_1603;

% interface utilisateur
disp('Logiciel de simulation non lin aire du F-16');
disp('-----');
disp('Le module utilis dans ce logiciel est bas sur celui developp ');
disp('dans le rapport de la NASA (1979)');
disp(' ');
disp('Nous rappelons les limitations physiques du module :');
disp(' ');
disp('----- Units Min Max');
disp('Variables d entree');
disp('Manette des gazs, delta_th----0----1');
disp('Elvateur, delta_e----deg---25.0----25.0');
disp('Ailerons, delta_a----deg---21.5----21.5');
disp('Gouvernail, delta_r----deg---30----30');
disp('Gouverne de profondeur, deg----0----25');
disp('delta_lef');
disp(' ');
disp('Etats du systeme');
disp('Vitesse----mach---0.1----1');
disp('----m---30----340');
disp('Angle d attaque, alpha----deg---10----45');
disp('Angle de derapage, beta----deg---30----30');
disp('Altitude----m----0----15240');
disp(' ');
disp('Veuillez choisir la condition de vol :');
v_des = input('vitesse d s r e en m/s (entre 30 et 240 m/s): ');
h_des = input('altitude d s r e en m (entre 0 et 15240 m): ');
disp(' ');
disp('Quelle manoeuvre souhaitez vous simuler ?');
disp('1. vol ailes horizontales (steady wings-level flight)');
disp('2. virage (steady turning flight)');
disp('3. tire de g (steady pull-up)');
disp('4. roulis (steady roll)');
choix_manoeuvre = input('Votre choix : ');

% calcul de la valeur de Pa initiale
delta_th_0 = 0.7;
if(delta_th_0 > 0.77)
    Pa_0=delta_th_0*P2+P3;
else
    Pa_0=delta_th_0*P1;

```

```

end

% initialisation du modele
x0 = [V_des;0;0; 0;0;0; 0;0;0; 0;0;h_des; 0;0;Pa_0;0;0];
xTrim = x0;
uTrim = [delta_th_0;0;0;0];

% trim en fonction de la manoeuvre effectuée
switch choix_manoeuvre
    case 1
        [x,u,y,dx,options]=trim('F16Reel_trim_cubic',[V_des;0;0; 0;0;0; ...
        0;0;0; 0;0;h_des; 0;0;0;0;0;0],[[],[],[1; 4;6; 7;8;9; 12],...
        [],[],zeros(18,1),[1;2;3; 4;5;6; 7;8;9; 11;12; 13;14;16;17;18]);
    case 2
        psi_dot_des = input('entrez_le_taux_de_rotation_desire_en_deg/s: ');
        psi_dot_des = psi_dot_des*pi/180;
        [x,u,y,dx,options]=trim('F16Reel_trim_cubic',[V_des;0;0; 0;0;0; ...
        0;0;0; 0;0;h_des; 0;0;0;0;0;0],[[],[],[1; 12],...
        [],[],[0;0;0; 0;0;psi_dot_des; 0;0;0; 0;0;0; 0;0;0;0;0;0],...
        [1;2;3; 4;5;6; 7;8;9; 12; 13;14;16;17;18]);
    case 3
        theta_dot_des = input('entrez_le_taux_de_montee_desire_en_deg/s: ');
        theta_dot_des = theta_dot_des*pi/180;
        [x,u,y,dx,options]=trim('F16Reel_trim_cubic',[V_des;0;0; 0;0;0; ...
        0;0;0; 0;0;h_des; 0;0;0;0;0;0],[[],[],[1; 4;6; 12],...
        [],[],[0;0;0; 0;theta_dot_des;0; 0;0;0; 0;0;0; 0;0;0;0;0;0],...
        [1;2;3; 4;5;6; 7;8;9; 13;14;16;17;18]);
    case 4
        phi_dot_des = input('entrez_le_taux_de_roulis_desire_en_deg/s: ');
        phi_dot_des = phi_dot_des*pi/180;
        [x,u,y,dx,options]=trim('F16Reel_trim_cubic',[V_des;0;0; 0;0;0; ...
        0;0;0; 0;0;h_des; 0;0;0;0;0;0],[[],[],[1; 6; 12],...
        [],[],[0;0;0; phi_dot_des;0;0; 0;0;0; 0;0;0; 0;0;0;0;0;0],...
        [1;2;3; 4;5;6; 7;8;9; 11;12; 13;14;16;17;18]);
end

% memorisation des donnees trimmees pour la simulation
xTrim = x;
uTrim = u;

% calcul de la valeur de delta_th trimme
if(xTrim(15)>52.3138)
    uTrim(1)=(xTrim(15)-P3)/P2;
else
    uTrim(1)=xTrim(15)/P1;
end

t_sim = input('Pendant_combien_de_secondes_voulez-vous_simuler_le_F-16... ...
---en_boucle_ouverte_?');
disp('Voulez-vous_creer_des_perturbations_sur_les_surfaces_de_controle_?');
disp('1._oui')
disp('2._non')
choix_perturbation = input('votre_rponse: ');
switch choix_perturbation
    case 1
        disp(' ');
        perturbation_deltae = input('veuillez_entrer_la_perturbation... ...
-----d_sir_e_sur_l_vateur_en_degres: ');
        t_perturbation_deltae = input('veuillez_entrer_le_temps_initial... ...
-----de_cette_perturbation_en_secondes: ');
        delta_perturbation_deltae = input('veuillez_entrer_la_duree_de... ...
-----cette_perturbation_en_secondes: ');
        disp(' ');

```

```
    perturbation_deltaa = input('veuillez entrer la perturbation ...  
d s i r e s u r l e s a i l e r o n s e n d e g r s : ' );  
    t_perturbation_deltaa = input('veuillez entrer le temps initial ...  
de cette perturbation en secondes : ' );  
    delta_perturbation_deltaa = input('veuillez entrer la duree de ...  
cette perturbation en secondes : ' );  
    disp(' ' );  
    perturbation_deltar = input('veuillez entrer la perturbation ...  
d s i r e s u r l e g o u v e r n a i l e n d e g r s : ' );  
    t_perturbation_deltar = input('veuillez entrer le temps initial ...  
de cette perturbation en secondes : ' );  
    delta_perturbation_deltar = input('veuillez entrer la duree de ...  
cette perturbation en secondes : ' );  
    disp(' ' );  
case 2  
    perturbation_deltae = 0;  
    t_perturbation_deltae = 0;  
    delta_perturbation_deltae = 0;  
    perturbation_deltaa = 0;  
    t_perturbation_deltaa = 0;  
    delta_perturbation_deltaa = 0;  
    perturbation_deltar = 0;  
    t_perturbation_deltar = 0;  
    delta_perturbation_deltar = 0;  
end  
[t,x,y] = sim('F16Reel_cubic',t_sim);
```