

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

Auteurs: Annie Girard
Authors:

Date: 2007

Type: Rapport / Report

Référence: Girard, A. (2007). Identification des coefficients aérodynamiques et commande de vol non-linéaire. (Rapport technique n° EPM-RT-2007-02).
Citation: <https://publications.polymtl.ca/3158/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/3158/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés / All rights reserved
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EPM-RT-2007-02
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:

(3 0 ± 5 7 ±

, '(1 7 ,) , & \$ 7 , 2 1 ' (6 & 2 ()) , & , (1 7 6
\$ e 5 2 ' < 1 \$ 0 , 4 8 ((7 & 2 0 0 \$ 1 ' (' (9 2 /
1 2 1 / , 1 e \$, 5 (

\$ Q Q H * L U D U G
' p S D U W H P H Q W G H * p Q L H p O H F W U L T X H
e F R O H 3 Q D T W H I F G K H 0 R Q W U p D O

- X L Q

Poly

(03 57

, '(17,), & \$7,21 '(6 & 2()), &, (176 \$e52' < 1 \$0,48 ((7
& 200\$1'('(92/ 121 /, 1 e\$,5(

\$QQLH *LUDUG
'pSDUWHPHQW GH *pQLH pOHFWULTXH «
eFROH 3RO\WHFKQLTXH GH ORQWUpDO

-XLQ

¤
\$ QQLH *LUDUG
7RXV GURLWV UpVHUYpV

'pS{W OpJDO
%LEOLRKqTXH QDWLRQDOH GX 4XpEHF
%LEOLRKqTXH QDWLRQDOH GX &DQDGD

(30 57
,GHQWLILFDWLRQ GHV FRHIILFLHQWV DpURG\QDPLTXHV HW FRPPDQGH GH YRO Q
SDU \$ QQLH *LUDUG
'pSDUWHPHQW GH JpQLH pOHFWULTXH
eFROH 3RO\WHFKQLTXH GH ORQWUpDO

7RXWH UHSURGXFWLRQ GH FH GRFXPHQW j GHV ILQV G pWXGH SHUVRQQHO OH RX
FRQGLWLRQOD FLWDWLRQ FL GHVVXV \ VRLW PHQWLRQQpH

7RXW DXWUH XVDJH GRLW IDLUH RQR FMUHWIS & QWHDVWHRDUQOMV SHXYHQW rWUH
DGUHVVpHV GLUHFPHQW DX[DXWHXUV FROWWSOZUZOSR @RRWWOD DVXU OH VLW
O HQWUHPLVH GH OD %LEOLRKqTXH

eFROH 3RO\WHFKQLTXH GH ORQWUpDO
%LEOLRKqTXH ± 6HUYLH GH IRXUQLWXUH GH GRFXPHQWV
&DVH SRVWDOH 6XFFXUVDOH ©&HQWUH 9LOOH^a
ORQWUpDO 4XpEHF
&DQDGD + & \$

7pOpSKRQH
7pOpFRSLH
&RXUULHU pOHFWURQLTXHEOLR VIG#SRO\PWO FD

3RXU VH SURFXUHU XQH FRSLH GMHFUH j ~~QWDESRBWWKATDXGUGH~~ O eFROH 3RO\WHFKQL

3UL[VXMHW j FKDQJHPHQW VDQV SUPDYLV
5pJOHU SDU FKqTXH RX PDQGDW SRVWH DX QRP GH O eFROH 3RO\WHFKQLTXH GH

7RXWH FRPPDQGH GRLW rWUH DFFRPSDJQpH G XQ SDLHPHQW VDXI HQ FDV G HQ
pWDEQHPHQWV G HQV BHJQMRHOpWpV HW GHV RUJDQLVPHV FDQDGLHQV

RESUME

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 contrôleur linéaires locaux pour chacun de ces points. Toutefois cette méthode demande un effort de conception important.

Un des principaux défauts 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, en angle de dérapage et en angle de roulis.

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 reconstruisable.

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 linearisé par retour de sortie, qui permet d'anticiper les changements de

trajectoires et de limiter les erreurs 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 suffisamment restreint pour être implémenté sur un processeur neuronal existant et réalise bien les objectifs de précision. La commande de vol par intégration 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 démontrent que les objectifs suivis de trajectoire sont atteints et c'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 acceptable 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 proportional+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 "real" 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

DEDICACE : :	iv
REMERCIEMENTS : :	v
RESUME : :	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 Equations newtoniennes	10
1.1.3 Equations du modèle aérodynamique	12
1.1.3.1 Modélisation linéaire	12
1.1.3.2 Modélisation non linéaire	13
1.2 Présentation du Fighting Falcon F-16	14
1.2.1 Un peu d'histoire	14
1.2.2 Le modèle aérodynamique du F-16	15

1.3 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 prémature 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 manuvrabilité	86
4.1.1 Qualités de manuvrabilité longitudinales	89
4.1.2 Qualités de manuvrabilité 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 a une par rapport à la commande	102
4.2.6 La commande en vitesse	104
4.2.7 Étude 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 $\alpha = 0.5^\circ/\text{s}^1$	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 "real"	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
 REFERENCES	180
 ANNEXES	186

LISTE DES TABLEAUX

Tableau 1.1	Définition de l'enveloppe de vol	17
Tableau 2.1	Algorithme de rétropagation	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	Sénaire 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 manuvrabilité	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 phugode	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 manuvrabilité 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	19
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'apprentissage	32
Figure 3.1	Représentation partielle des points statiques	45
Figure 3.2	Représentation 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 .	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	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 appliquée au F-16	73
Figure 3.19	Coefficients longitudinaux - Erreur d'apprentissage en %	80

Figure 3.20	Coe cients latéraux - Erreur d'entraînement en %	81
Figure 3.21	Comportement du module d'identi cation	82
Figure 3.22	Performance du reseau de la région 1 pour $q = 0$	83
Figure 3.23	Performance du reseau de la région 2 pour $q = 0$	83
Figure 3.24	Performance du reseau de la région 3 pour $q = 0$	84
Figure 3.25	Performance du reseau de la région 4 pour $q = 0$	84
Figure 4.1	Echelle de Cooper-Harper	87
Figure 4.2	Mouvement du mode phugode	90
Figure 4.3	Mouvement du roulis hollandais	93
Figure 4.4	Architecture a séparation temporelle de la dynamique	97
Figure 4.5	Architecture alternative	98
Figure 4.6	Schemas bloc comparant trois con gurations de commande par inversion dynamique	109
Figure 4.7	Lieu des racines pour la première m thode	110
Figure 4.8	Réponse en tangage suite a un doublet de $1s^{-1}$ en commande	111
Figure 4.9	Réponse suite a un doublet de 5 en commande	114
Figure 4.10	Réponse avec saturation suite a un doublet de 5 en commande	114
Figure 4.11	Architecture du contrôleur	134
Figure 5.1	Vola ailes plates - evolution des variables d'état	137
Figure 5.2	Evolution des variables d'état suite a une perturbation sur ϵ	137
Figure 5.3	Trajectoire de l'avion suite a une perturbation sur ϵ	139
Figure 5.4	Vola ailes plates - coe cients aérodynamiques	139
Figure 5.5	Virage - evolution des etats	141
Figure 5.6	Evolution des etats suite a une perturbation sur a et r	141
Figure 5.7	Trajectoire de l'avion suite a une perturbation sur a et r	142
Figure 5.8	Virage - coe cients aérodynamiques	142
Figure 5.9	Décomposition de l'autothrottle	143
Figure 5.10	Commande en P_c - Réponse a un changement de consigne	145

Figure 5.11	Commande en T - Réponse à un changement de consigne	145
Figure 5.12	Influence de W sur l'allocation	149
Figure 5.13	Allocation du contrôle - rejet de perturbation	149
Figure 5.14	Suivi d'une trajectoire - réponse en vitesse _q	153
Figure 5.15	Suivi d'une trajectoire - réponse en angles, $\dot{\theta}$ et $\dot{\phi}$	153
Figure 5.16	Suivi d'une trajectoire - réponse en vitesse de rotation q et r	154
Figure 5.17	Suivi d'une trajectoire - déactions des surfaces _e , _a , _r et _{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 _q	156
Figure 5.20	Suivi d'une trajectoire - réponse en angles, $\dot{\theta}$ et $\dot{\phi}$	157
Figure 5.21	Suivi d'une trajectoire - déaction des surfaces _e , _a , _r et _{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 prédition	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 _q	163
Figure 5.28	Suivi de trajectoire - réponse en $\dot{\theta}$ et $\dot{\phi}$	163
Figure 5.29	Suivi de trajectoire - réponse en θ_e , $\dot{\theta}_a$, $\dot{\theta}_r$ et $\dot{\theta}_{lef}$	164
Figure 5.30	Performance du système linearisé par retour de sortie sans PI	166
Figure 6.1	Commande en altitude - influence de Q	172
Figure 6.2	Commande en altitude - angle d'attaque générale	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 - $C.I.V_t = 100\text{m:s}^{-1}$ et $h = 6000\text{m}$	176
Figure 6.8	Commande en altitude - pente de 10s^{-1}	176
Figure IV.1	Linearisation entrée-état	198
Figure IV.2	Linearisation 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

:	angle d'attaque, en rad (ou , précise dans ce cas)
b:	envergure des ailes, em
:	angle de dérapage, em rad (ou , précise dans ce cas)
c:	corde aérodynamique moyenne, em
C:	coefficient aérodynamique, sans dimension
CAP:	Control Anticipation Parameter, ou paramètre d'anticipation du contrôle
α :	angle de déflexion de la surface de commande(a, e ou r), en
θ :	position de la manette des gaz, en %
g:	constante de gravitation, em:m:s ⁻²
h:	altitude, en m
h_{moteur} :	moment angulaire du moteur, erg: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, erg
p:	vitesse angulaire de roulis dans le repère avion, rad:s ⁻¹
q:	vitesse angulaire de tangage dans le repère avion, rad:s ⁻¹
\dot{q} :	pression dynamique, emPa
r:	vitesse angulaire de lacet dans le repère avion, rad:s ⁻¹
:	densité de l'air
S:	surface des ailes de l'avion, em ²

T:	poussée instantanée du moteur, en N
u:	vitesse de l'avion selon l'axe 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 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 du repère avion, en m/s ¹
x _{cg} :	position du centre de gravité, en % de overline{c}
x _{cg_{ref}} :	position de référence du centre de gravité, en % de overline{c}
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
:	assiette de roulis, en rad (ou , précise dans ce cas)
:	assiette de tangage, en rad (ou , précise dans ce cas)
:	azimut, en rad (ou , précise dans ce cas)
:	coefficient d'amortissement
! :	pulsation propre

INDICES

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

n:	du moment de lacet
O:	composante principale
p:	dû au roulis
ph:	du mode phugode
q:	dû au tangage
r:	dû au lacet
rh:	du roulis hollandais
x:	des forces suivant l'axe x de l'avion
y:	des forces suivant l'axe y de l'avion
z:	des forces suivant l'axe z de l'avion

LISTE DES ANNEXES

ANNEXE I	RAPPELS DE DYNAMIQUE DU VOL	186
I.1	Equations cinématiques et de navigation	186
I.2	Equations 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	COMMANDE PAR LINÉARISATION EXACTE	197
IV.1	Linéarisation entrée-état	197
IV.2	Linéarisation entrée-sortie	199
ANNEXE V	COMMANDE 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 identique	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

Des le milieu du 20^eme 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 "y-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 manuvrabilité et une augmentation de la sécurité (notamment vis-à-vis du décrochage). En outre les configurations aérodynamiques instables ne sont plus exclues 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 dans 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 amené à se trouver dans des conditions de vol trop éloignées de la région linearisée, la stabilité du système peut être compromise ; il faut donc discréteriser l'espace d'état avec nesse.

Un des principaux des de la commande de vol moderne est de se dégager de ces contraintes grâce à la conception de lois de commande non linéaires et/ 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.

Etat de l'art

Des les années 1950, les chercheurs portent leur intérêt sur le développement de la commande adaptive en vue de concevoir des autopilotes pour des aéronefs à haute performance. En effet, de tels avions fonctionnent sur une large plage de vitesses d'altitudes. Il est mis en évidence que les lois de contrôle linéaires usuelles, soit 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 adaptive 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 linéaire 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 spéciales du système (Lane & Stengel, 1998). Mais la robustesse de ce contrôleur laisse à désirer. Dans les applications suivantes 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 signifiants) en utilisant les états rapides (les taux de rotation signifiants) comme entrée et une boucle interne contrôle les états rapides à partir des états réelles du système (élevateur, ailerons et gouvernail). Toutefois la stabilité en boucle fermée n'est pas pour autant assurée, du fait que l'hypothèse d'une telle séparation régime lent et rapide n'est pas toujours valide. La solution proposée par Eade (1997) pour le problème de benchmark GARTEUR utilise une telle inversion, et ajoute au contrôleur une inversion dynamique non linéaire un contrôleur de 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 & Partha Sarathy, 1990), à 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 avantage naturel, que ce soit pour un modèle linéaire d'aéronef (Nho & Agarwal, 1999) ou pour un processus industriel non linéaire (Zamarreno & 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, Soly et Gold (1999) a prouvé que celle-ci convenait également des systèmes constants de temps rapides tels que les aéronefs grâce à l'apprentissage en ligne.

A 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 au précédent (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 linearisé (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 contre-direkt adaptatif à modèle de référence (Kamalasadan & Ghandakly, 2005), le contre-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 recon gurable 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 recon gurable (Bodson & Groszkiewicz, 1995).

La première approche, qui découle du programme ci-dessus, est la plus répandue. Les contrôleurs de vols recon gurable 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 modélisation du comportement de l'aéronef est trop important pour être exhaustivement pertorié, 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 Bodson (1997) établissent le lien existant entre ce type de commande de vol recon gurable et la commande adaptative : la commande adaptative permet non seulement de s'adapter aux conditions différentes de vol mais aussi une large gamme de pannes sans que celles-ci soient prédictées.

Les réseaux de neurones traitent cette seconde approche en incorporant directement le caractère recon gurable 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 recon gurable 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 (Pashikar, Sundararajan & Saratchandran, 2006) ou encore un contrôleur reconstruisable adaptatif directement 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 reconstruction 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 sécurité 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 reconstruisables 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, 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 verra à garder une flexibilité suffisante pour une évolution du contrôleur vers un contrôleur reconstruisable.

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

PR ESENTATION DU SYST EME

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 recueillies au cours d'expériences sur le système. Cette dernière 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 appliquée 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 actions des surfaces de contrôle, vitesses angulaires de l'avion, la vitesse de l'avion ou encore son orientation (K & 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 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 concider 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 gravitationnel. C'est cette configuration (north, east, down) qui lui a donné son acronyme NED.

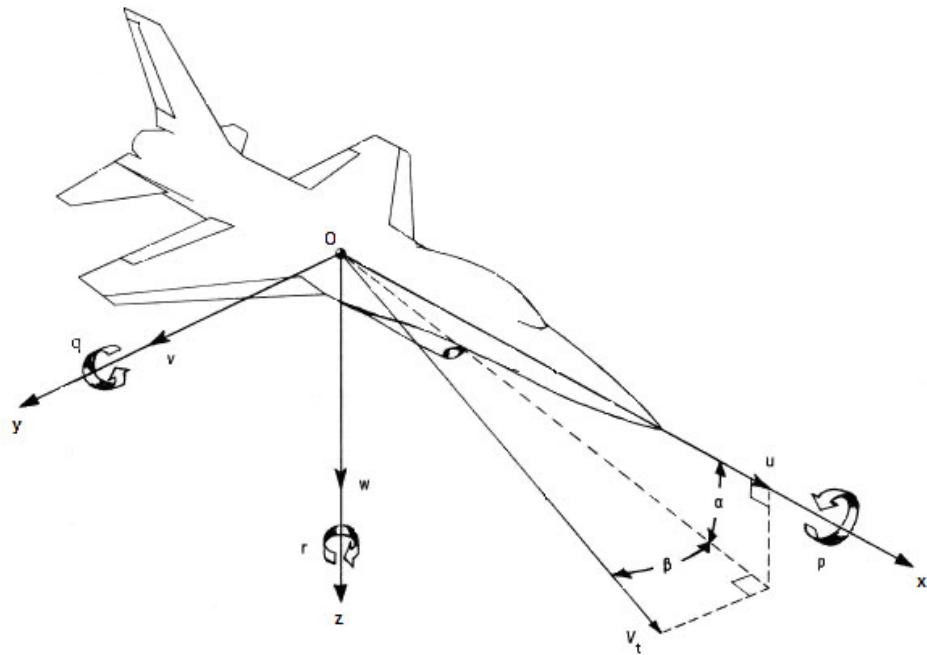


Figure 1.1 Système d'axes de l'avion et angles d'incidence

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 concorde 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 est l'angle formé dans le plan x_0z entre l'axe des x et la projection du vecteur vitesse parallèlement à l'axe y sur ce plan.

l'angle de dérapage est l'angle formé dans le plan V_0y entre la projection de l'axe des x parallèlement à l'axe z sur ce plan et le vecteur vitesse.

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, l'angle d'attaque et l'angle de dérapage :

$$\begin{aligned} V_t &= \sqrt{u^2 + v^2 + w^2} \\ &= \tan^{-1} \frac{w}{u} \\ &= \sin^{-1} \frac{v}{V_t} \end{aligned} \tag{1.1}$$

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

$$\begin{aligned} u &= V_t \cos \alpha \cos \gamma \\ v &= V_t \sin \alpha \cos \gamma \\ w &= V_t \sin \alpha \sin \gamma \end{aligned} \tag{1.2}$$

1.1.2 Equations 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'inertiel 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'e et 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 gravité 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 :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}) \quad (1.3)$$

où \mathbf{x} est un vecteur des variables d'état $V_t, \dots, p, q, r, \dots, x_E, y_E, h$; et \mathbf{u} est un vecteur des variables de contrôle composé de la manette des gaz et des défections des surfaces de contrôle. Les variables du vecteur \mathbf{u} n'apparaissent pas explicitement dans l'ensemble des équations précédentes, mais y figurent 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 $, \dots, p, q, r, \dots, x_E, y_E, h, a_x, a_y, a_z, p, q$ et r .

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

$$\mathbf{y} = \mathbf{H}(\mathbf{x}; \mathbf{x}; \mathbf{u}) \quad (1.4)$$

1.1.3 Equations 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 actions 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 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 :

$$\begin{aligned} C_a &= C_{a_0} + C_{av} \frac{V}{V_0} + C_a + C_{aq} \frac{qc}{2V_0} + C_a && \text{pour } a = x, z \text{ ou } m \\ C_a &= C_{a_0} + C_a + C_{ap} \frac{pb}{2V_0} + C_{ar} \frac{rb}{2V_0} + C_a && \text{pour } a = y, l \text{ ou } n \end{aligned} \quad (1.5)$$

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 recherche 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 déuctions des surfaces de contrôle. De plus il est usuel de considérer que, pour des régimes de vol subsatellites 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):

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

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 vols 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 appliquée 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 soumis 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 Etats-Unis. En 1971, le programme visait à concevoir un avion de chasse léger ("Lightweight Fighter") avec des spécifications précises 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 était mis sur une manuvrabilité accrue, notamment en accélération et vitesses de rotation, au détriment de la vitesse. À 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. Ce premier avion a vu doter d'un système "fly-by-wire" : il n'y a aucun lien mécanique entre les commandes du pilote et les surfaces de contrôles qui sont levées, 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 type 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 manuvrabilité. L'accent étant mis sur la manuvrabilité 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 actionneuses, 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 16% du F-16 et sont fournies sous forme de tables pour un large spectre d'angles d'attaque et d'angles de dérapage:

$$20^\circ < \alpha < 45^\circ \text{ et } 30^\circ < \beta < 30^\circ.$$

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.

$$\frac{1}{\alpha_{ref}} = 1 : 38 \frac{2s + 7.25}{s + 7.25} - \frac{180}{9.05 \frac{q}{P_s} + 1.45} \quad (1.7)$$

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

$$\begin{aligned}
 C_a = & C_{a_0}(\alpha; \beta; \gamma_e) + q \frac{\bar{C}}{2V_t} C_{a_q}(\alpha) + C_{a_{lef_q}}(\alpha) 1 - \frac{lef}{25} \\
 & + C_{a_{lef}}(\alpha) 1 - \frac{lef}{25} \quad \text{pour } a = x \text{ ou } z \\
 C_m = & C_{m_0}(\alpha; \beta; \gamma_e) + q \frac{\bar{C}}{2V_t} C_{m_q}(\alpha) + C_{m_{lef_q}}(\alpha) 1 - \frac{lef}{25} \\
 & + C_{m_{lef}}(\alpha) 1 - \frac{lef}{25} + C_m(\alpha) + x_{cg_{ref}} x_{cg} C_z
 \end{aligned} \tag{1.8}$$

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

$$\begin{aligned}
 C_a = & C_{a_0}(\alpha; \beta; \gamma_e) + C_{a_{lef}}(\alpha) 1 - \frac{lef}{25} + C_{a; r=30^\circ}(\alpha) \frac{r}{30} \\
 & + C_{a; a=20^\circ}(\alpha) + C_{a_{lef}; a=20^\circ}(\alpha) 1 - \frac{lef}{25} \frac{a}{20} \\
 & + p \frac{b}{2V_t} C_{a_p}(\alpha) + C_{a_{lef_p}}(\alpha) 1 - \frac{lef}{25} \\
 & + r \frac{b}{2V_t} C_{a_r}(\alpha) + C_{a_{lef_r}}(\alpha) 1 - \frac{lef}{25} + C_{a_{beta}}(\alpha) \\
 & \frac{\bar{C}}{b} x_{cg_{ref}} x_{cg} C_y \quad \text{si } a = n
 \end{aligned} \tag{1.9}$$

ou pour tout a :

$$\begin{aligned}
 C_{a_lef}(\cdot) &= C_{a_lef}(\cdot) - C_{a_0}(\cdot; e=0) \\
 C_{a; r=30^\circ}(\cdot) &= C_{a; r=30^\circ}(\cdot) - C_{a_0}(\cdot; e=0) \\
 C_{a; a=20^\circ}(\cdot) &= C_{a; a=20^\circ}(\cdot) - C_{a_0}(\cdot; e=0) \\
 C_{a_lef; a=20^\circ}(\cdot) &= C_{a_lef; a=20^\circ}(\cdot) - C_{a_lef}(\cdot) - C_{a; a=20^\circ}(\cdot)
 \end{aligned} \tag{1.10}$$

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
vitesseM	[0.1 ; 1]	nb de Mach
altitude h	[0 ; 15 500]	m
angle d'attaque	[-10 ; 45]	dégré
angle de dérapage	[-30 ; 30]	dégré
vitesse angulaire en roulis	[-595 ; 595]	:s ⁻¹
vitesse angulaire en tangage	[-205 ; 205]	:s ⁻¹
vitesse angulaire en lacet	[-135 ; 135]	:s ⁻¹
déflection de l'élevateur e	[-25 ; 25]	dégré
déflection des ailerons a	[-21 ; 21]	dégré
déflection du gouvernail r	[-30 ; 30]	dégré
déflection du bord d'attaque lef	[0 ; 25]	dégré

1.3 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 déterminent le comportement des différents avions.

L'ensemble est alors appliquée 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 arti ciel

L'élément de base d'un réseau de neurones est le neurone arti ciel 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, $w_{k1}; w_{k2}; \dots; w_{kp}$.

On définit alors le champ induit local ϕ_k comme suit :

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

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

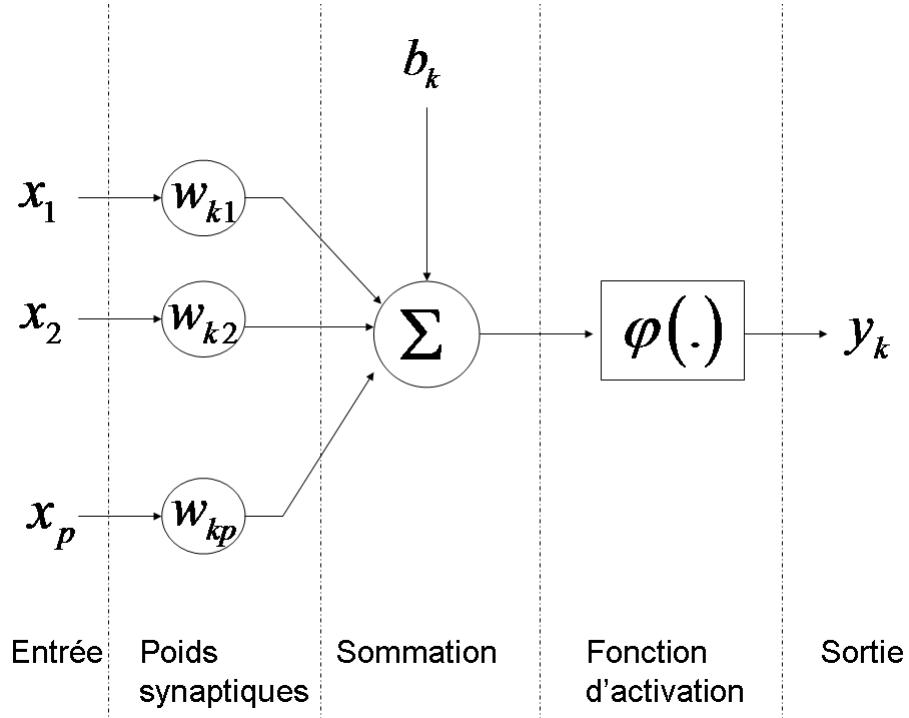


Figure 2.1 Modèle non linéaire d'un neurone

fonction d'activation ' $\varphi(\cdot)$ ' pour donner la sortie du neurone:

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

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 de réseaux de neurones, et sont les plus utilisés dans le cadre de l'identification et du contrôle des systèmes dynamiques non linéaires.

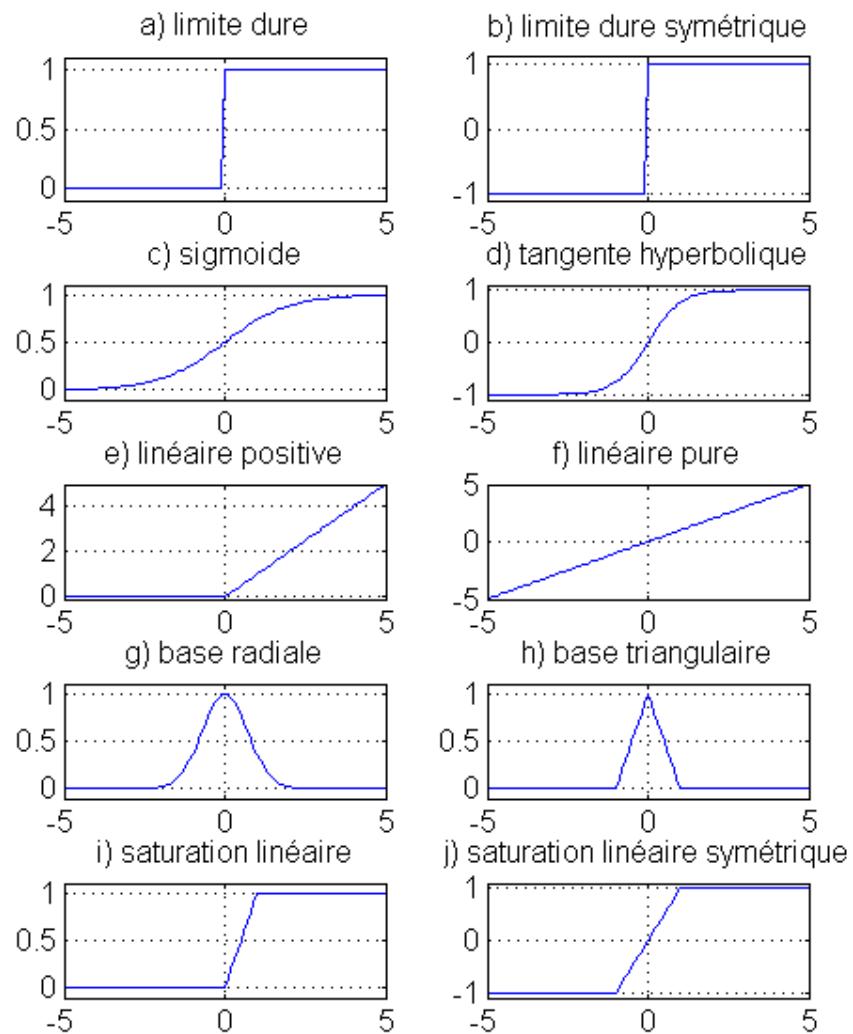


Figure 2.2 Fonctions d'activation usuelles

Un MLP est composé de plusieurs couches, elles-mêmes composées 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) ~~haut~~(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.

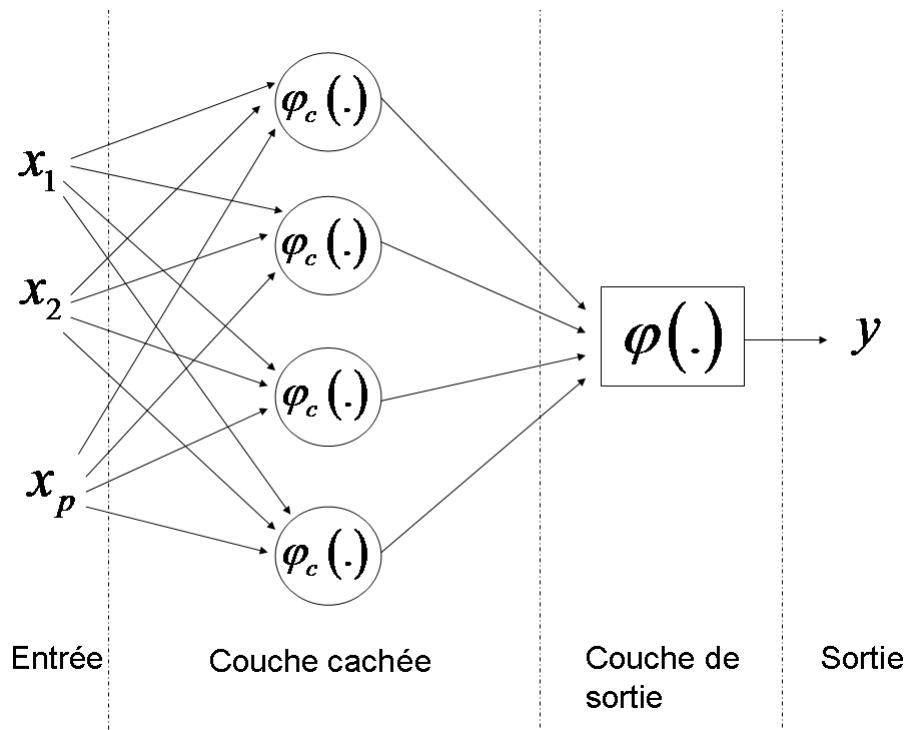


Figure 2.3 Exemple de MLP à une couche cachée

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

$$x_k^i = \sum_{j=1}^{N_i} w_{kj}^i x_j^i \quad (2.3)$$

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

Les sorties de chaque couche sont alors :

$$y_k^i = \sigma(\sum_k w_k^i + b_k) \quad (2.4)$$

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 sera 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 choisir 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 ("sequential 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 rétropager 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 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 :

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

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

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

$$\text{Energie moyenne} = \frac{1}{N} \sum_{n=1}^{N^k} \text{Energie}(n) \quad (2.7)$$

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 Energie moyenne .

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

$$\Delta w_{kj}^i = -\eta \frac{\partial \text{Energie}(n)}{\partial w_{kj}^i} \quad (2.8)$$

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

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

$$\Delta w_{kj}^i = -\eta \frac{\partial \text{Energie}(n)}{\partial w_{kj}^i} = -\eta \frac{\partial \text{Energie}(n)}{\partial y_k^i} y_k^i(n) \quad (2.9)$$

où $\frac{\partial \text{Energie}}{\partial y_k^i}$ est le gradient local de la couche i et est défini par :

$$\frac{\partial \text{Energie}(n)}{\partial y_k^i} = \frac{\partial \text{Energie}(n)}{\partial e_k^i} = e_k^i(n) \cdot \frac{\partial e_k^i(n)}{\partial y_k^i} \quad (2.10)$$

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 :

$$e_k^i(n) = \sum_j e_j^{i+1}(n) !_{kj}^{i+1}(n) \quad (2.11)$$

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 sigmoidales.

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

3. La propagation :

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

$$P_k^i(n) = \prod_{j=0}^{m_k} !_{kj}^i(n) y_j^{i-1}(n)$$

4. La rétro-propagation :

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

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

$$= ' P_k^i(n) \prod_{j=0}^{m_k} !_{kj}^i(n) y_j^{i-1}(n) \text{ sinon}$$

puis mise à jour des poids synaptiques :

$$!_{kj}^i(n+1) = !_{kj}^i(n) + \alpha_i(n) y_j^{i-1}(n)$$

α est le taux d'apprentissage.

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

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

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

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

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 :

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

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 compliqué 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 :

$$\begin{aligned} g_n &= J_n^T(\hat{x})_n(\hat{x}) \\ H_n &= J_n^T(\hat{x})J_n(\hat{x}) + I \end{aligned} \tag{2.15}$$

où $J_n(\cdot)$ est la Jacobienne à l'instant n , \hat{x}_n est la valeur de la fonction coût à l'instant n et I un scalaire.

On remarque que si λ est grand alors c'est équivalent à la descente du gradient avec un taux d'apprentissage faible et si λ 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 : λ est donc diminué si l'étape a conduit à une baisse de la valeur de la fonction coût, et λ est augmenté si l'inverse de 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 $x(n)$ au réseau, calculer les sorties correspondantes et le vecteur d'erreur $e(n)$. Calculer la fonction de coût:

$$\mathcal{J}_n(\beta) = \sum_{k=1}^N e_k^{(n)}(\beta)$$

2. Calculer la matrice jacobienne $J_n(\beta)$

3. Mettre à jour les poids :

$$\beta_{n+1} = \beta_n - J_n^T(\beta_n) J_n(\beta_n) + \alpha I \quad J_n^T(\beta_n) e_n(\beta_n)$$

4. Calculer $\mathcal{J}_{n+1}(\beta_{n+1}) = \sum_{k=1}^N e_k^{(n+1)}(\beta_{n+1})$

Si $\mathcal{J}_{n+1}(\beta_{n+1}) < \mathcal{J}_n(\beta_n)$ alors $\beta_{n+1} = \beta_{n+1}$, où α est une constante, puis retournera l'étape 1.

Si $\mathcal{J}_{n+1}(\beta_{n+1}) > \mathcal{J}_n(\beta_n)$ alors $\beta_{n+1} = \beta_n + \Delta\beta$, et retournera l'étape 3 pour recalculer β_{n+1} avec la nouvelle valeur de $\Delta\beta$.

5. Itération des étapes 2 à 4 jusqu'à avoir rencontré un critère d'arrêt (par exemple fonction coût 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 ~~isole~~ 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 prémature de l'apprentissage.

2.4.1 Données d'entraînement

Les réseaux de neurones sont non linéaires. Cela impose une excitation sur ~~le~~ 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 ~~le~~ "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 (Nørregaard, 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 filtre, 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 identifiée, 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 à 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 prétraiter 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 courant 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 prématuré de l'apprentissage

Un paramètre à vérifier 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 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 prématuement. 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 prématu 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.

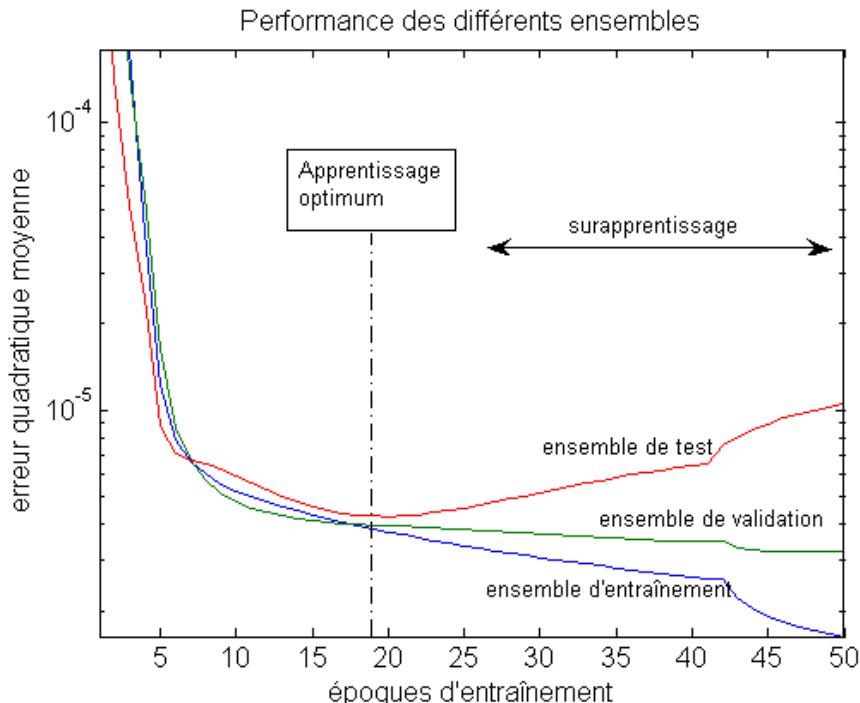


Figure 2.4 Arrêt prématu de l'entraînement

CHAPITRE 3

IDENTIFICATION DU MODÈLE À ERODYNAMIQUE

L'objectif du processus d'identification est d'obtenir un modèle aérodynamique qui représente de manière précise et détaillée 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 menée 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 apprises 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 cartographier les valeurs des coefficients aérodynamiques sur l'ensemble de l'enveloppe de vol. Cependant avec les méthodes classiques (algorithme des moindres carrés d'ajustage 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 reconnaissable. 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 additionnelles 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.

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 recueillies à 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 ~~couches~~, la valeur de ses paramètres et le choix de ses entrées et sorties. Tous ~~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 a une couche cachée est capable d'approximer toute fonction non linéaire complexe avec une précision aussi élevée 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 assez élevé d'obtenir un réseau optimal.

Tableau 3.1 Dépendances des coefficients aérodynamiques

Modèle	Coefficients	Variables
Longitudinal	$C_x; C_m; C_z$; ; $\alpha; q; V_t; (\alpha_{ref})$
Latéral	$C_l; C_y; C_n$; ; $\alpha; r; p; r; V_t; (\alpha_{ref}; \alpha_e)$

En ce qui concerne le choix des entrées, le modèle aérodynamique établi en 1.B.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 déformations 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. Cet 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 utilisés 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 trop 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 en 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. Ainsi, le 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 à dynamique 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 assignera une partition moins étendue donc plus restreint. L'application d'un processus d'interpolation ou de lissage aux frontières des partitions permet de plus de conserver le caractère continue 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

: 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 avec m_1 neurones sur la couche cachée

entraîner le réseau sur p époques

sauvegarder le réseau et ses performances

n

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.

Reitérer 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 cachée 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 zones 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 redondante, le lien étroitement établi entre l'introduction entre ces deux lois de commande suggère une possible extension de la configuration de commande 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 aisé, de plus il faut garantir que le nombre de neurones le composant n'explose pas.

Une autre méthode permettant de limiter ce phénomène d'oubli a récemment été développée dans le laboratoire d'aéronautique de l'université de Delft sur un modèle basse qualité 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. 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 C_x ne dépende pas des taux de rotation p ou r , ou des déformations des surfaces de contrôle latérales a et 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 $[; ; e; a; r; V_t; h; p; q; r]$ 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} ; C_{x_lef} et $C_{x_lef_q}$ 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 ; ; ; ; ; et . 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 linearité 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 ϕ	350; 175; 0; 175; 350	:s ⁻¹
vitesse angulaire de tangage θ	100; 50; 0; 50; 100	:s ⁻¹
vitesse angulaire de lacet ψ	50; 25; 0; 25; 50	:s ⁻¹

Pour chaque point statique, l'espace d'entrée est partitionné. De Weerdt (2005) souligne que certaines lignes directrices doivent être respectées : les hypercubes soit les partitions à point statique fixe, doivent être choisis les plus grands possibles afin que leur nombre soit minimal, mais suffisamment petits pour que l'effacement d'un 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	[-10;1] [-1;11] [9;21] [19;31] [29;45]	degré
angle de dérapage	[-30;-19] [-21;-9] [-11;1] [-1;11] [9;21] [19;30]	degré
action de l'élevateur	[-25;-10] [-14;2] [-2;14] [10;25]	degré
action d'aileron	[-21.5;-10] [-14;2] [-2;14] [10;21.5]	degré
action de gouvernail	[-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 x_3 , 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 (x_1 et x_2). Le chevauchement entre les différentes régions n'a pas été illustré pour des questions de visibilité.

Seuls les hypercubes dont la région associée comprend la condition de vol $\dot{a} \leq 0$ ($x_1; x_2; x_3; x_4; x_5$) sont actives. 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 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 suit pour passer d'une région à une autre au milieu de la zone de chevauchement.

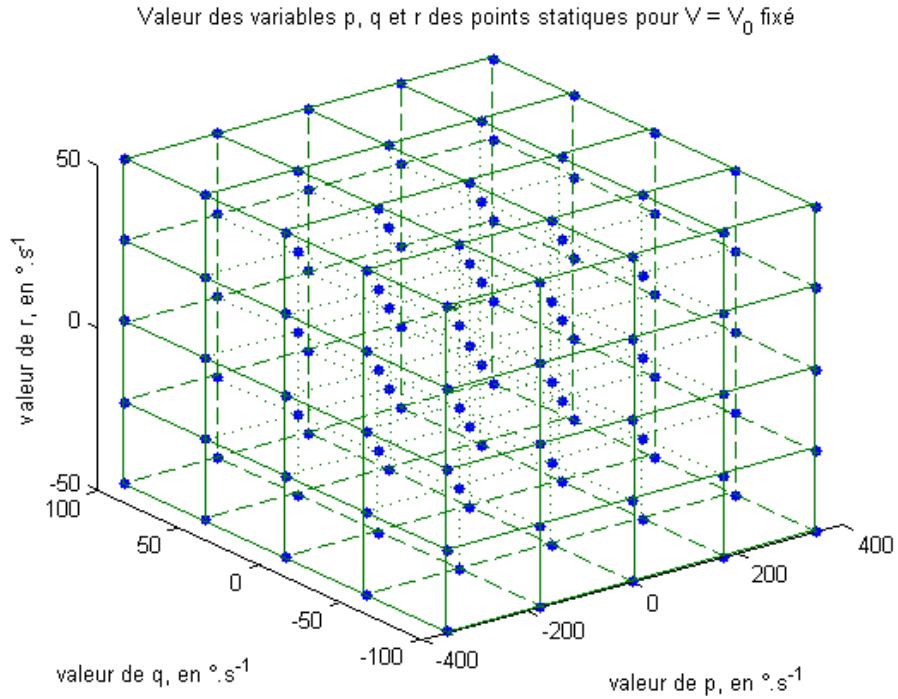


Figure 3.1 Representation partielle des points statiques

$$V = V_0, p = p_0, q = q_0, r = r_0$$

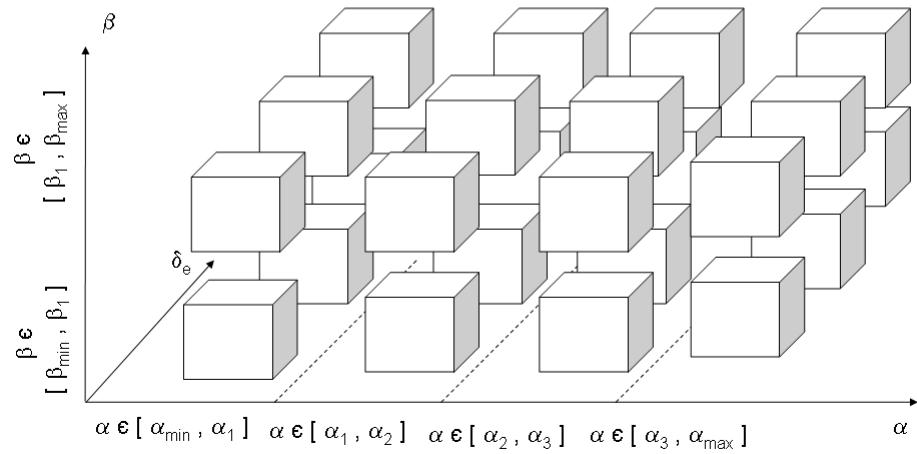


Figure 3.2 Representation partielle des hypercubes

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 donnée 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 $\mathbf{d}_t(p; q; r)$ de la condition de vol.

En effet, si la condition de vol se trouve être un point statique, les valeurs de 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 correspondantes celles d'un point statique) à $2^3 = 16$ (si aucune variable n'a la valeur correspondante 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 réconfigurable, 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.

Ainsi de ne pas développer toute la structure en hypercubes un cas simple est étudié. Seule l'identification du coefficient C_x au point statique $V_t = 150 \text{ m/s}^1$ et $q = 0 \text{ :s}^{-1}$ est effectuée. De plus un modèle simple 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 déection de l'élevateur 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 $\beta = 0$ et à déection de l'élevateur $\delta_e = 0$. On obtient un module, dit à une région unique, qui approxime la loi entrée-sortie sur l'ensemble de la plage d'angle d'attaque, appelé enveloppe de vol dans cette section, et un module, dit à plusieurs régions, qui est une application de la technique des hypercubes de De Weerdt.

Pour le module d'identification à plusieurs régions, les réseaux associés à chaque hypercube (ou chaque partition d'angle d'attaque) contiennent un faible nombre de neurones (entre 4 et 6) et l'erreur quadratique moyenne d'entraînement, d'une valeur de $2.5 \cdot 10^{-5}$, y est inférieure à la valeur de référence $2.5 \cdot 10^{-4}$. Pour obtenir les mêmes performances en termes d'erreur, le module à une 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 $2 \cdot 10^{-5}$.

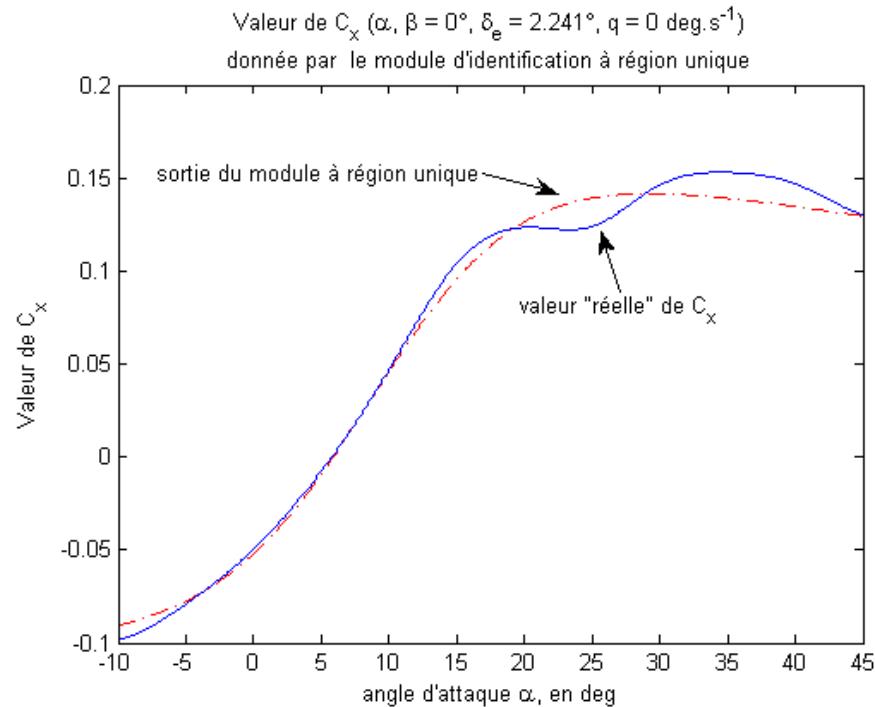


Figure 3.3 Performance du module à région unique

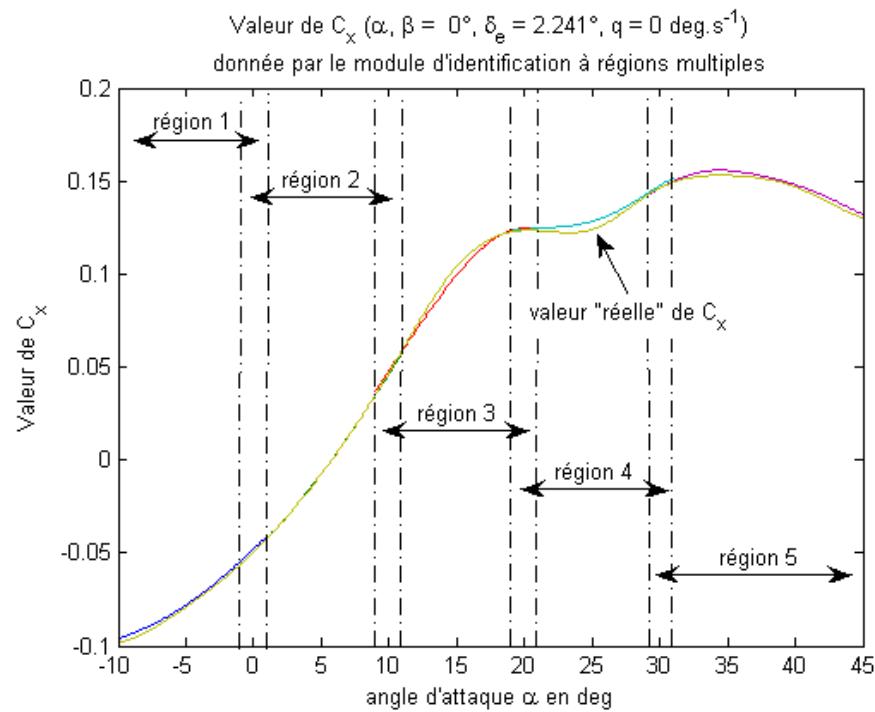


Figure 3.4 Performance du module à régions multiples

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 avec l'angle de dérapage et la défection de l'élevateur x_{es} : $\alpha = 0$ et $x_{es} = 2;241$. 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 plusieurs régions approche très précisément le coefficient C_x sur toute l'étendue de l'angle d'attaque. Le module à une 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 proportionnel à sa taille, ce dernier module est le moins rapide.

Le temps d'apprentissage hors ligne est coûteux. Chaque réseau du module à plusieurs régions est certes plus rapide et plus simple à concevoir que le module à une région unique, toutefois on dénombre 1 440 000 réseaux dans la structure hypercubes de De Weerdt (2005). Il y a donc un compromis à faire entre nécessité 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 configuration 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 est nul, la défection de l'élevateur est égale à 2;241°, et la vitesse angulaire de tangage est nulle.

Tableau 3.5 S<ø>nario g<ø>ral pour l'apprentissage en ligne

-
1. Choisir un module d'identi cation du mod<ø>le aérodynamique, dont les performances suite a l'entraînement hors ligne sont satisfaisantes
 2. G<ø>erer l'ensemble des couples (\dot{v} ; $\dot{\alpha}$; $\dot{\theta}$; \dot{q}) caract<ø>risant la trajectoire :
 $0 < t < t_1$: vol horizontal stabilisé a altitude constante ($h = 5715m$) et a 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'identi cation : un vecteur d'ent<ø>e a la fois est simulé. Le vecteur d'ent<ø>e est ajouté a une FIFO dédiée et la valeur de C_x mesurée a une autre.
 Entraînement : le module est entraîné avec les données se trouvant dans les FIFO dédiées d'ent<ø>ees et de cibles.
-

Les figures 3.5 et 3.6 sont le résultat d'une augmentation linéaire d'angle d'attaque entre 4; 503a $t_1 = 50$ et 6; 503a $t_2 = 100$.

Le phénomène d'oubli affecte le module d'identi cation a région unique :

avant que l'angle d'attaque ne varie, le module s'est adapté a la condition de vol.

lorsque l'angle d'attaque commence a augmenter, le module a région unique présente une erreur supérieure a $27:10^{-4}$ (voir figure 3.6), donc supérieure a l'erreur maximale suite a l'apprentissage hors ligne. Le module a donc perdu de l'information qu'il contenait a priori.

pour des angles d'attaques supérieurs a 15 (voir figure 3.5), la performance du module s'est nettement dégradée : c'est le phénomène d'oubli.

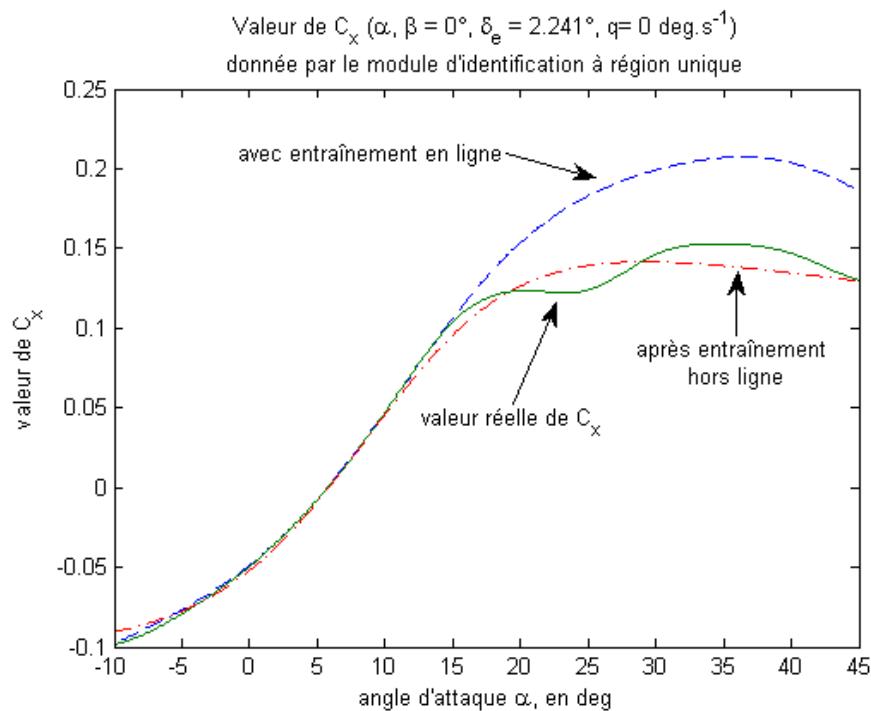


Figure 3.5 Phénomène d'oubli

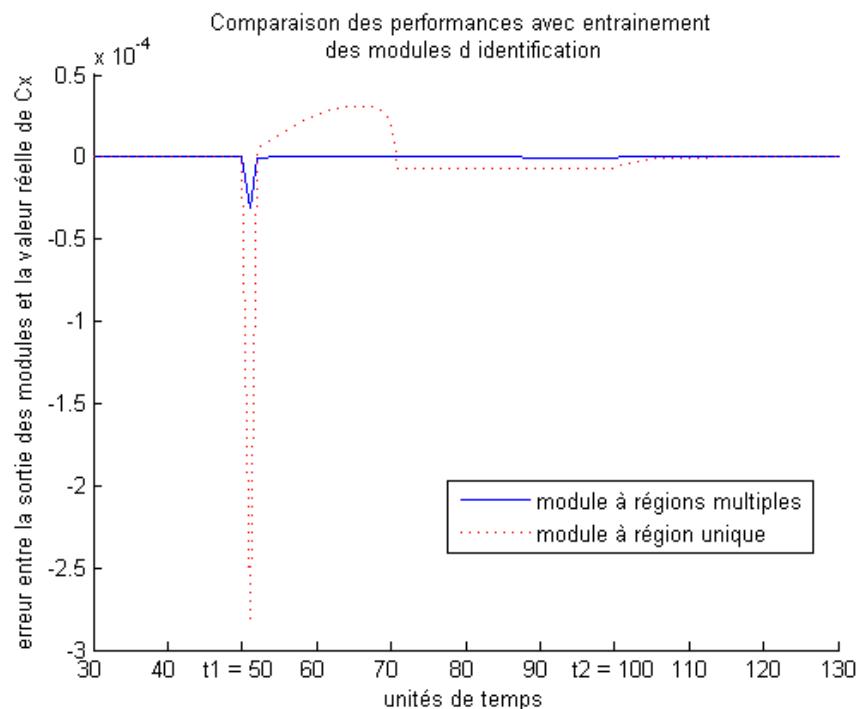


Figure 3.6 Performance suite à une augmentation d'angle d'attaque

Le module a *regions multiples* quanta lui n'est quasiment pas affecté par le changement d'angle d'attaque. La partition a permis de préserver du phénomène d'oubli, car les *regions* 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 *raisons d'être* de la structure en hypercubes.

Les scénarios suivants permettent d'évaluer qualitativement la performance des modules d'identification dans le cas où une panne ou un dégât survient. La reconfiguration 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 nous 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. At = $t_1 = 30$, l'angle d'attaque passe de 1;400 à 4;530. 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 *region unique* le phénomène d'oubli est très marqué, avec une erreur de l'ordre de 40³ 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).

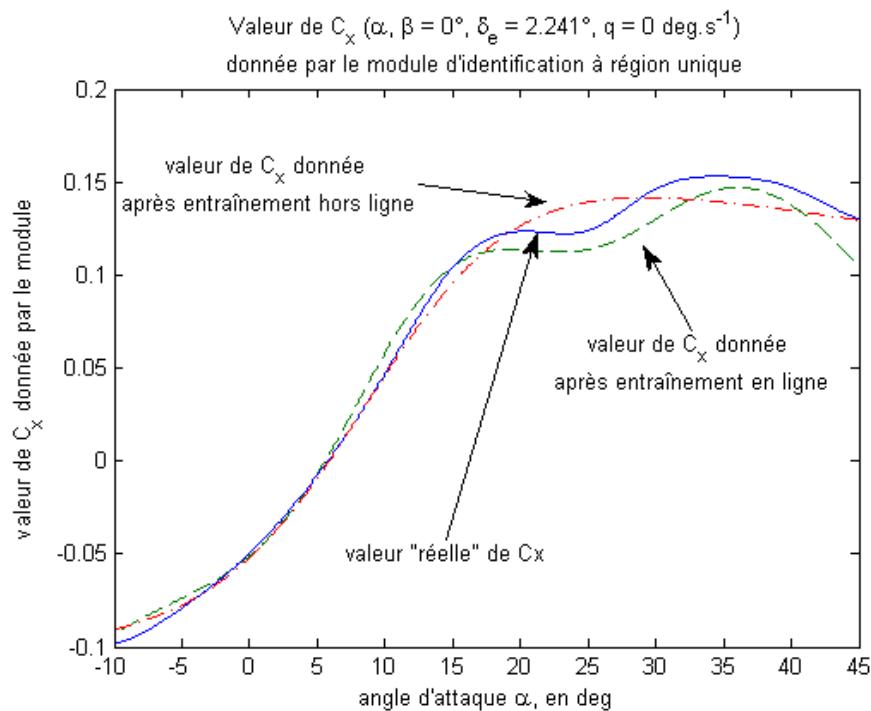


Figure 3.7 Scénario 1 - Module d'identification à région unique

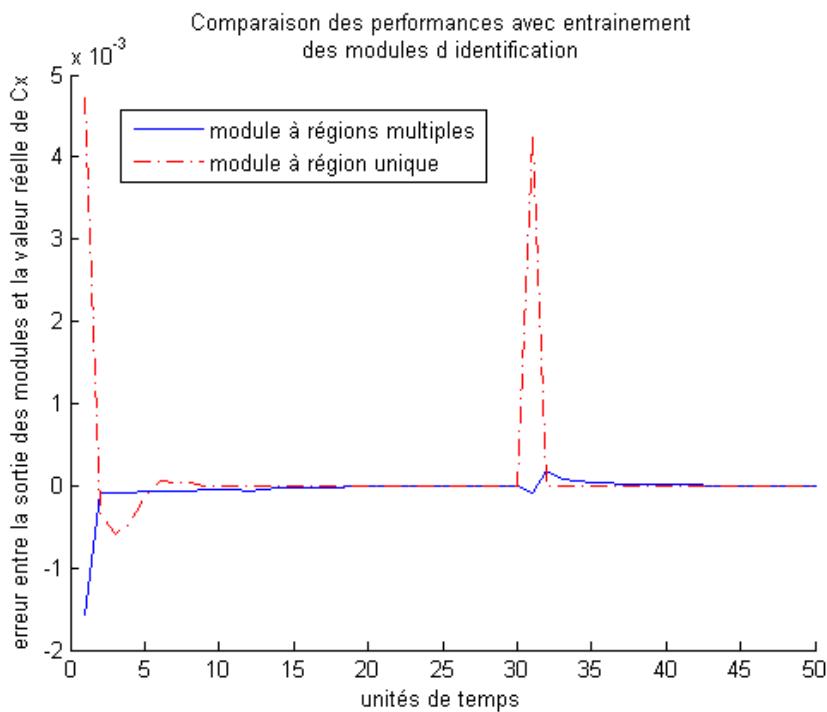


Figure 3.8 Scénario 1 - Performance lors d'un changement brutal d'angle d'attaque

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 remplacé 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 = 1;4. Cette configuration est appelée "point de fonctionnement".

En l'absence d'excitation supplémentaire sur l'ensemble de l'espace d'entrée, aucun des deux modules d'identification n'est même capable 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.

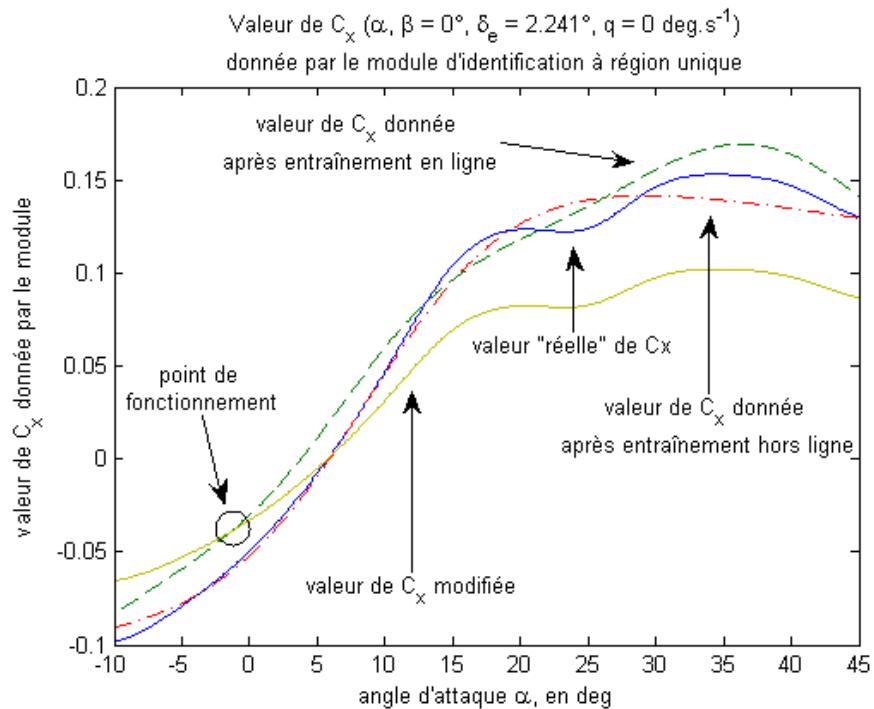


Figure 3.9 Scénario 2 - Module d'identification à région unique

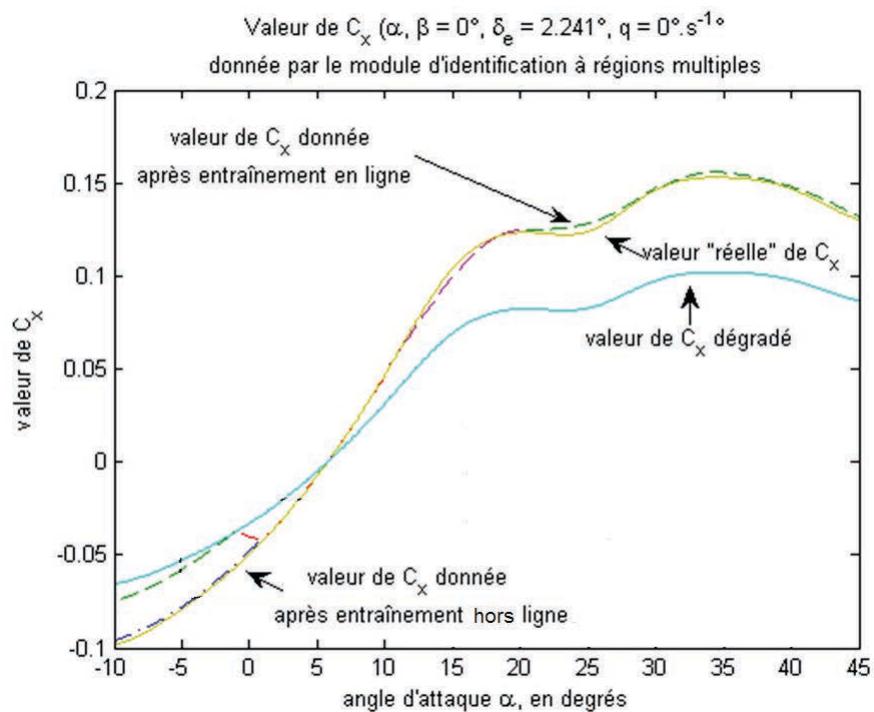


Figure 3.10 Scénario 2 - Module d'identification à régions multiples

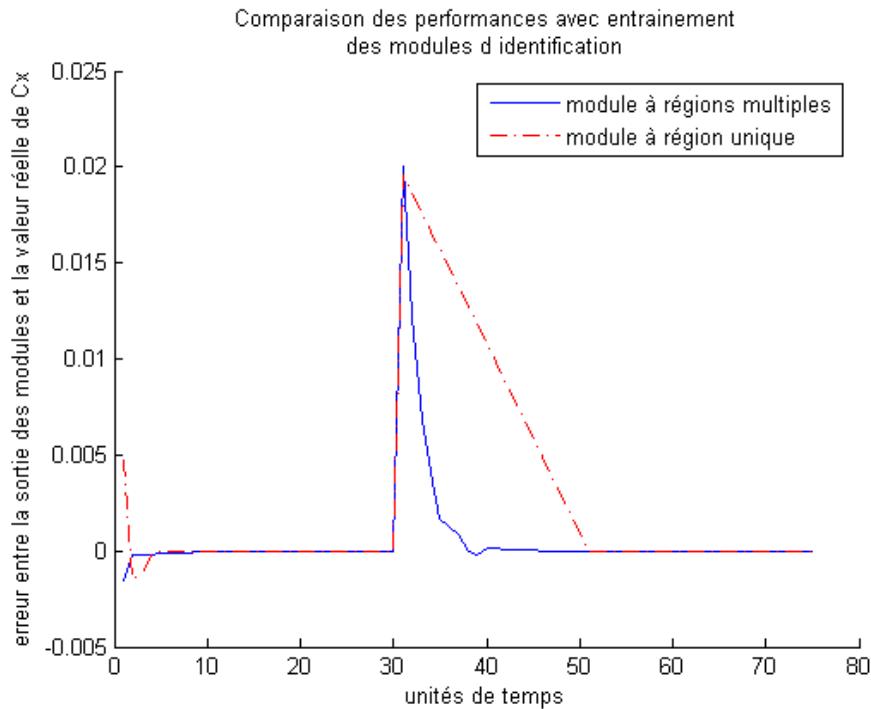


Figure 3.11 Scénario 2 - Performance lors d'un changement brutal

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 pragmatique à être entraîné en ligne (voir 2.4).

Tableau 3.6 Caractéristiques des instruments de mesure

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

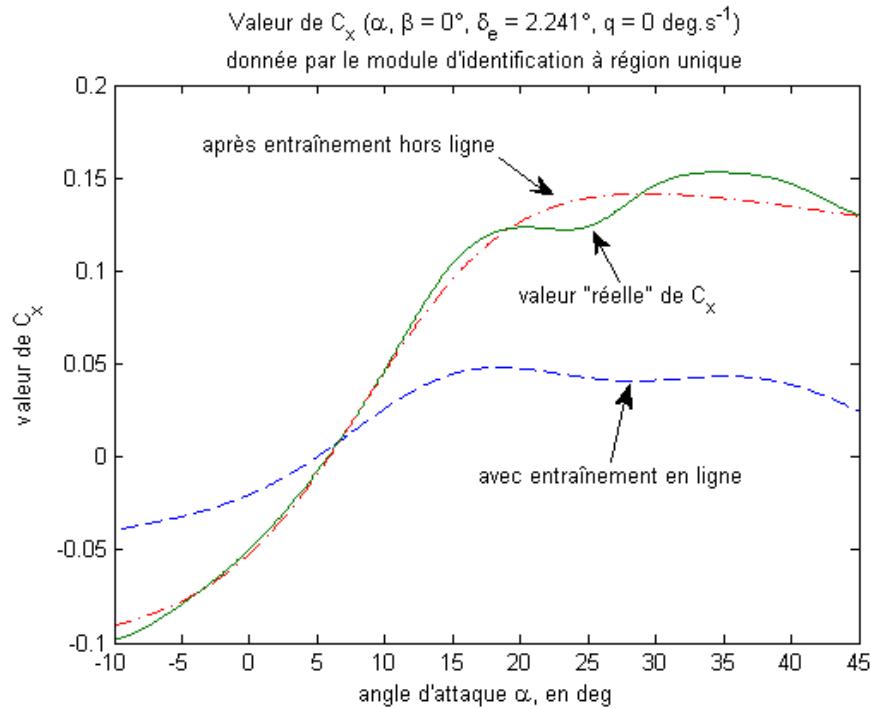


Figure 3.12 Scénario 3 - Module d'identification à région unique

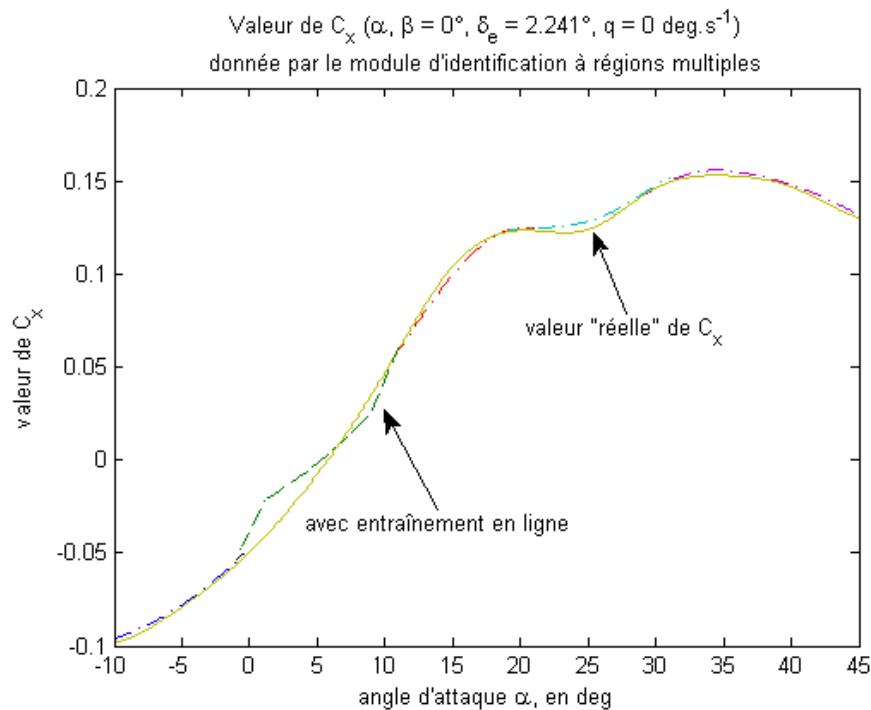


Figure 3.13 Scénario 3 - Module d'identification à régions multiples

Reprendons la simulation d'apprentissage en ligne initiale, soit une augmentation linéaire d'angle d'attaque entre $4503a$ $t_1 = 50$ et $6503a$ $t_2 = 100$, 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 $\alpha = 6503$, 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 arrêté et reste présent au sein de chaque région.

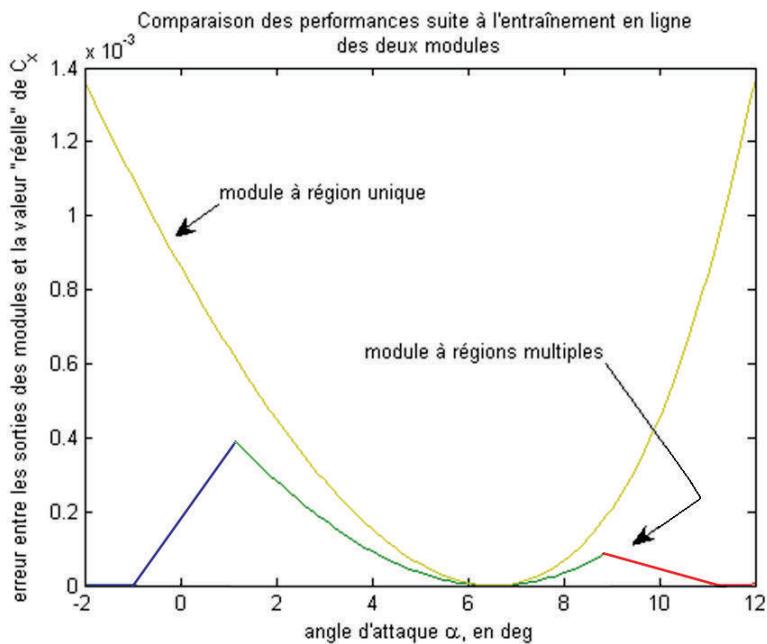


Figure 3.14 Scénario 3 - Comparaison de la performance des deux modules

3.3.3 Conclusion de l'analyse de la méthode des hypercubes

Le partitionnement de l'espace d'entrée présente des avantages notables comme 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 identification 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 reconstruction. 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 délitie 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 pour 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 reconstruction n'est pas traitée exhaustivement dans ce projet, ces allégements 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-

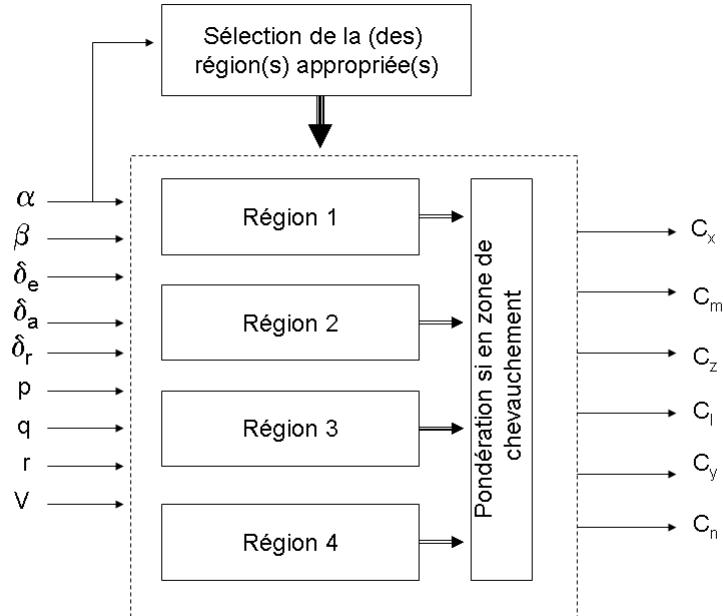


Figure 3.15 Architecture générale

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 .

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 présente, 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 nales 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 ($< 15^\circ$), 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 démonté.

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

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 $[l_{\min} ; l_{lin}]$ 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

: critère de performance

m_{\max} : nombre maximum de neurones sur la couche cachée des réseaux

: largeur des zones de chevauchement

2. Initialisation

Soit $[\alpha_{\min}; \alpha_{\max}]$ la plage d'angles d'attaque considérée

$n = 1$, rang de l'itération

Développer le réseau identifiant le modèle aérodynamique avec le critère de performance

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

Soit m le nombre de neurones de sa couche cachée

n

4. Itération

Reprendre les étapes 2 et 3 avec la plage $[\alpha_{\min}; \alpha_{\max}]$ restante

5. Chevauchement des régions

Les partitions $[\alpha_{\min}; \alpha_1]; [\alpha_1; \alpha_2]; \dots; [\alpha_{n-1}; \alpha_{\max}]$ sont obtenues

Etendre ces plages de sorte à ce qu'elles se chevauchent :

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

Afin de trouver plus rapidement un partitionnement, l'algorithme peut être appliquée 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 ϕ , 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, moins la V_0 .

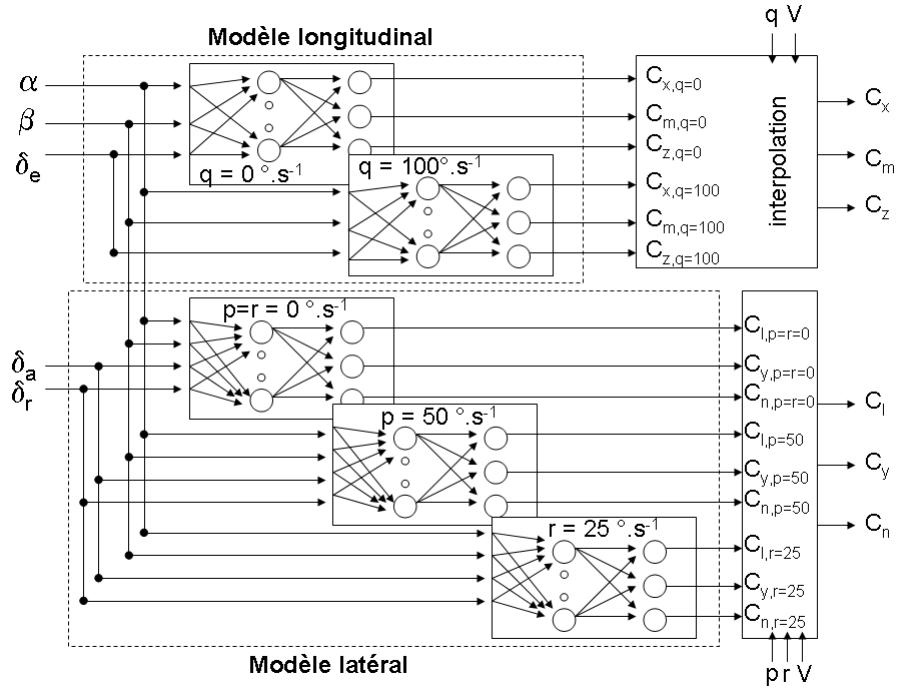


Figure 3.16 Architecture d'une région

Quant aux vitesses angulaires, on peut se limiter à deux valeurs de référence : on simplifie 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 :

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

A chaque point statique de chaque région correspond un réseau de neurones.

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 utilisées 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 doivent 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 attendue des coefficients aérodynamiques à la condition de vol désirée selon les équations générales suivantes :

$$\begin{aligned} C_{a_{\text{total}}} &= C_{a_0} (; ; e) + \frac{qc}{2V_t} C_{a_q} (; ; e) && \text{pour les longitudinaux} \\ C_{a_{\text{total}}} &= C_{a_0} (; ; a; r) + \frac{pb}{2V_t} C_{a_p} (; ; a; r) \\ &+ \frac{rb}{2V_t} C_{a_r} (; ; a; r) && \text{pour les latéraux} \end{aligned} \quad (3.2)$$

Ces équations générales 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 attendues 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 $C_{x_{q=0}}$, $C_{m_{q=0}}$, $C_{z_{q=0}}$ g et f $C_{x_{q=q_0}}$, $C_{m_{q=q_0}}$, $C_{z_{q=q_0}}$ g, et 3 réseaux pour le modèle latéral, qui

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

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:

$$C_a = C_a(; ; e)_{q=0} + \frac{qc}{2V_t} C_{a_q}(; ; e) \quad (3.3)$$

ou $C_{a_q} = \frac{2V_0}{q_0 c} C_a(; ; e)_{q=q_0} - C_a(; ; e)_{q=0}$

et les coefficients latéraux C_l , C_y et C_n par :

$$C_a = C_a(; ; a; r)_{p=0;r=0} + \frac{pb}{2V_t} C_{a_p}(; ; a; r) + \frac{rb}{2V_t} C_{a_r}(; ; a; r) \quad (3.4)$$

ou $C_{a_p} = \frac{2V_0}{p_0 b} C_a(; ; a; r)_{p=p_0;r=0} - C_a(; ; a; r)_{p=0;r=0}$

et $C_{a_r} = \frac{2V_0}{r_0 b} C_a(; ; a; r)_{p=0;r=0} - C_a(; ; a; r)_{p=0;r=r_0}$

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 n'ont pas fourni aux réseaux de ces régions. Ces réseaux ne fournissent donc pas de sortie dans ce cas. Afin de ne pas avoir une sortie vide pour ces régions, la dernière valeur active est maintenue.

A 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 activee, les valeurs des coefficients aérodynamiques sont directement celles fournies par cette région.

Si deux régions sont actives 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 valeur de l'angle d'attaque 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 :

$$C_a = \frac{1}{2} C_{a_2} + \frac{1}{2} C_{a_1} \quad (3.5)$$

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

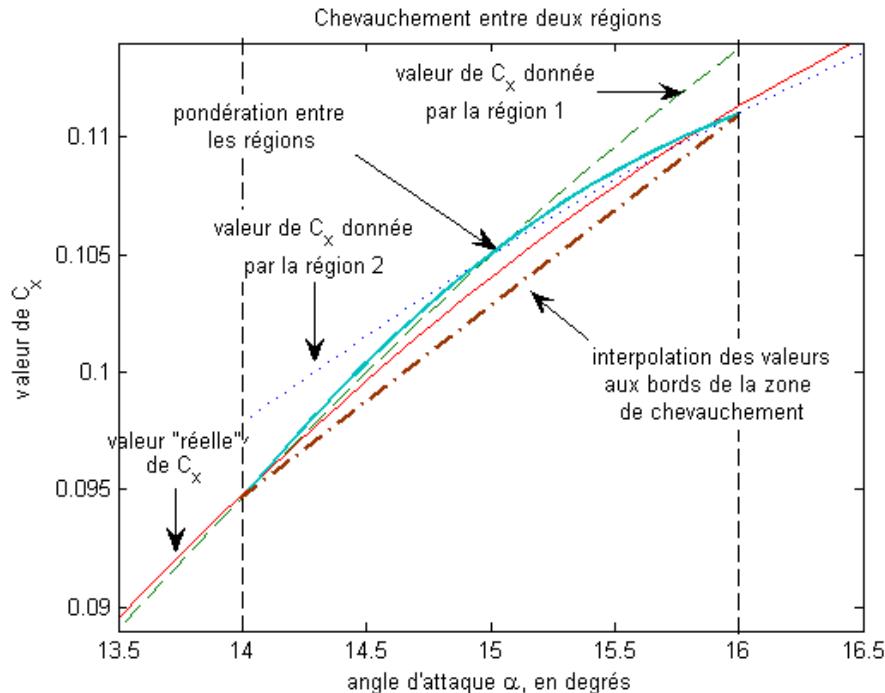


Figure 3.17 Changement de région continu

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

$$(C_a - C_{a_{\text{réel}}})^2 = \frac{1}{2} (C_{a_2} - C_{a_{\text{réel}}}) + \frac{2}{2} (C_{a_1} - C_{a_{\text{réel}}})^2 < \quad (3.6)$$

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 proposée 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, les grandes lignes (DeWeerdt et al., 2005) sont exposées ici.

Dans un premier temps, les données nécessaires à l'apprentissage en lignes 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.

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

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

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 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 connexion 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 inférieur 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 ms (à 33 MHz). Un cycle d'apprentissage pour ce même ensemble dure quant à lui 0,246 ms (à 33 MHz). Sachant que les accéléromètres fournissent une donnée toutes les 0,5 à 1,2 ms, et les gyroscope toutes les 3 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 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 développe 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 étendu pour prendre en compte l'angle du bord d'attaque. L'élévateur doit gérer parmi les entrées "globales" du module, car son angle de déflection influence l'ensemble des coefficients aérodynamiques, au même titre que les angles d'attaque et de dérapage.

La première étape du développement est le choix des régions, conformément à la procédure 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 qualité" du F-16, a mené à choisir $\Delta_{lin} = 15^\circ$. L'algorithme de la section 3.4.1 est appliqué au coefficient C_{x_0} pour $q = 0 \text{ :s}^{-1}$, avec un critère de performance de $14:10^{-6}$, 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 été 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 également 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 raisons 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 $[34^\circ, 45^\circ]$.

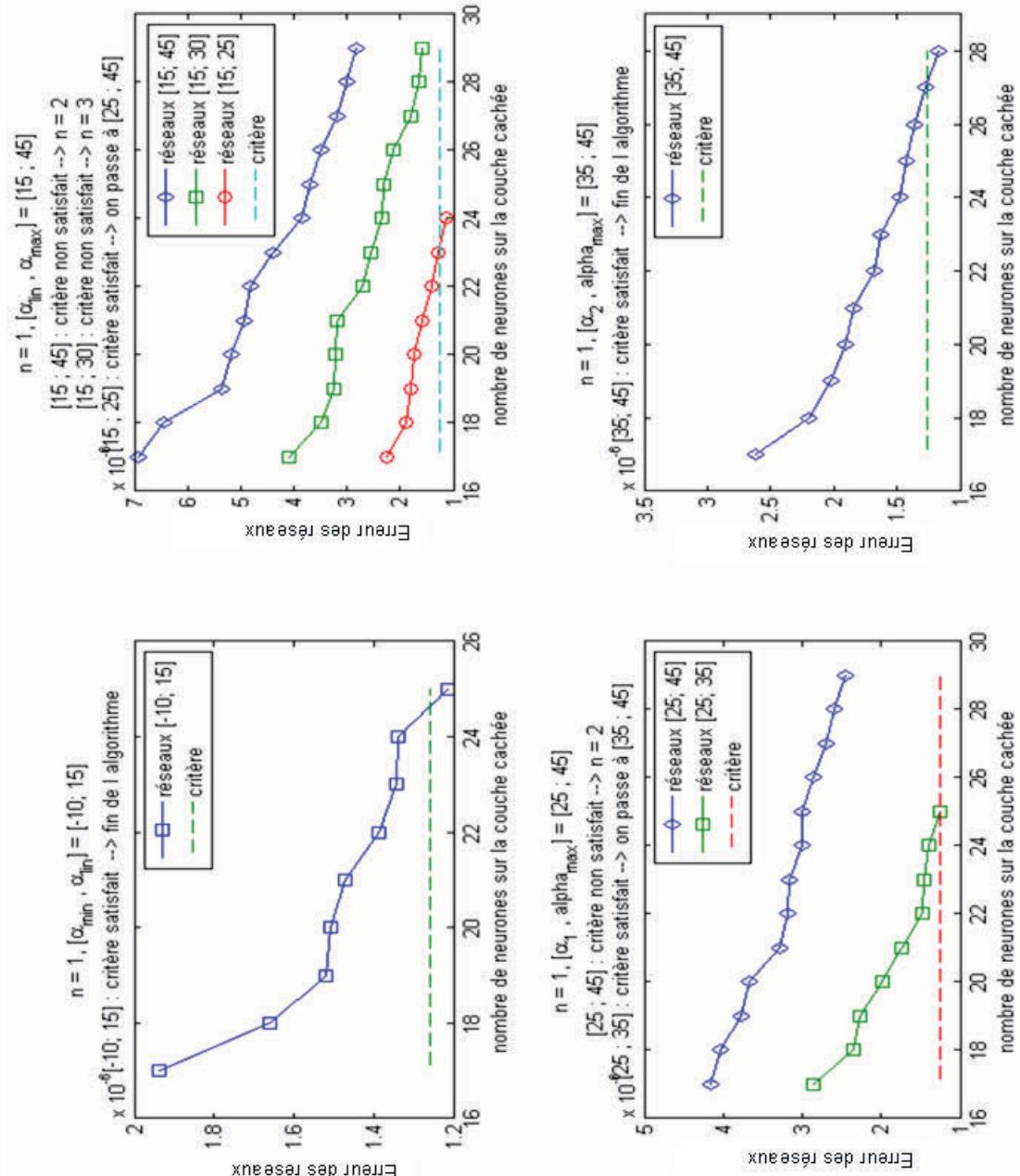


Figure 3.18 Illustration de l'algorithme de choix des régions appliquée au F-16

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	:s ⁻¹
Taux de tangage q	[-205 ; 205]	0 ; 100	:s ⁻¹
Taux de lacet r	[-135 ; 135]	0 ; 25	:s ⁻¹
Angle d'attaque	[-10 ; 45]	[-10 ; 16] [14 ; 26] [24 ; 36] [34 ; 45]	degré
Angle de dérapage	[-30 ; 30]	[-30 ; 30]	degré
Élevateur e	[-25 ; 25]	[-25 ; 25]	degré
Ailerons a	[-21 ; 21]	[-21 ; 21]	degré
Gouvernail r	[-30 ; 30]	[-30 ; 30]	degré
Bord d'attaque lef	[0 ; 25]	[0 ; 25]	degré

Les points statiques doivent être xés. 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 que $\frac{C_D}{2V_0}$, $\frac{p_0 b}{2V_0}$ et $\frac{r_0 b}{2V_0}$ soient proches de l'unité : $V_0 = 150 \text{ m:s}^{-1}$, $p_0 = 50 \text{ :s}^{-1}$, $q_0 = 100 \text{ :s}^{-1}$ et $r_0 = 25 \text{ :s}^{-1}$. 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 ((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 :

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

Tableau 3.9 Compilation par région des coefficients longitudinaux

Region	1			
valeur de q	0		100	
	E		E	
valeur de C_x	0; 0322	00424	0; 0055	00533
valeur de C_m	0; 0220	01427	0; 1832	01495
valeur de C_z	0; 2075	04961	0; 7796	05704
Region	2			
valeur de q	0		100	
	E		E	
valeur de C_x	0; 0465	00411	00804	00498
valeur de C_m	0; 0727	01375	0; 2657	01366
valeur de C_z	1; 2496	02323	1; 8522	02158
Region	3			
valeur de q	0		100	
	E		E	
valeur de C_x	0; 0536	00494	00760	00535
valeur de C_m	0; 1236	01271	0; 3277	01251
valeur de C_z	1; 7087	02260	2; 3128	02369
Region	4			
valeur de q	0		100	
	E		E	
valeur de C_x	0; 0615	00525	00838	00557
valeur de C_m	0; 1612	01001	0; 3746	00983
valeur de C_z	1; 9724	02054	2; 6987	02162

Tableau 3.10 Compilation par région des coefficients latéraux

Région		1		
valeur de f p; rg		f 0; 0g	f 50; 0g	f 0; 25g
	E		E	E
valeur de C _l	1; 3:10 ⁴	0; 040	1; 0:10 ²	0; 040
valeur de C _y	5; 3:10 ³	0; 362	4; 1:10 ³	0; 362
valeur de C _n	8; 0:10 ⁴	0; 077	9; 5:10 ⁴	0; 077

Région		2		
valeur de f p; rg		f 0; 0g	f 50; 0g	f 0; 25g
	E		E	E
valeur de C _l	3; 2:10 ⁴	0; 050	8; 2:10 ³	0; 050
valeur de C _y	2; 8:10 ³	0; 306	6; 1:10 ³	0; 306
valeur de C _n	7; 8:10 ⁴	0; 051	2; 0:10 ³	0; 051

Région		3		
valeur de f p; rg		f 0; 0g	f 50; 0g	f 0; 25g
	E		E	E
valeur de C _l	1; 5:10 ³	0; 044	7; 4:10 ³	0; 044
valeur de C _y	1; 2:10 ²	0; 230	2; 7:10 ⁴	0; 230
valeur de C _n	1; 8:10 ³	0; 031	6; 3:10 ³	0; 032

Région		4		
valeur de f p; rg		f 0; 0g	f 50; 0g	f 0; 25g
	E		E	E
valeur de C _l	1; 9:10 ³	0; 037	2; 8:10 ³	0; 037
valeur de C _y	4; 4:10 ³	0; 182	1; 8:10 ²	0; 186
valeur de C _n	3; 2:10 ⁴	0; 040	1; 0:10 ²	0; 040

Tableau 3.11 Valeurs extrêmes des coe cients

Coe cien t		Min.	Max.	(:0;01) ²
$q = 0 \text{ :s}^{-1}$	C_x	0;1712	01838	03550 $126:10^5$
	C_m	0;4319	03396	07715 $595:10^5$
	C_z	2;4471	09573	34044 $116:10^3$
$q = 100 \text{ :s}^{-1}$	C_x	0;1527	02175	03702 $137:10^5$
	C_m	0;6183	01915	08038 $646:10^5$
	C_z	3;2204	07574	39778 $158:10^3$
$p = 0 \text{ :s}^{-1}, r = 0 \text{ :s}^{-1}$	C_l	0;1155	01801	02956 $874:10^6$
	C_y	0;8096	08775	16871 $284:10^4$
	C_n	0;1966	01816	03782 $143:10^5$
$p = 50 \text{ :s}^{-1}, r = 0 \text{ :s}^{-1}$	C_l	0;1272	01639	02911 $847:10^6$
	C_y	0;7974	08928	16902 $286:10^4$
	C_n	0;1955	01837	03792 $144:10^5$
$p = 0 \text{ :s}^{-1}, r = 25 \text{ :s}^{-1}$	C_l	0;1091	01736	02927 $857:10^6$
	C_y	0;7971	09593	17564 $308:10^4$
	C_n	0;2024	01759	03783 $143:10^5$

Les valeurs extrêmes des coe cients (voir tableau 3.11) permettent de déterminer le critère d'arrêt pour l'apprentissage des réseaux. On souhaite identifier les coe cients 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 coe cients 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

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

La moyenne de ces deux valeurs est bien égale au critère usuel, soit une erreur moyenne quadratique d'entraînement inférieure à $25:10^{-4}$.

Les réseaux sont alors développés selon la procédure mise en évidence 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 (nombre de 1400000). L'architecture élaborée ici en compte entre 25 et 61, malgré le fait qu'il 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

Region d'angle d'attaque	1	2	3	4
	$p = 0 :s^{-1}, r = 0 :s^{-1}$			
nb neurones	40	32	45	59
performanceC _l	1; 54:10 ⁶	3; 45:10 ⁶	4; 38:10 ⁶	4; 02:10 ⁶
performanceC _y	5; 07:10 ⁵	5; 02:10 ⁵	3; 49:10 ⁵	6; 24:10 ⁵
performanceC _n	4; 88:10 ⁶	4; 23:10 ⁶	3; 08:10 ⁶	6; 61:10 ⁶
performance globale	,190:10 ⁵	1; 93:10 ⁵	1; 41:10 ⁵	2; 44:10 ⁵
	$p = 50 :s^{-1}, r = 0 :s^{-1}$			
nb neurones	40	34	47	61
performanceC _l	1; 49:10 ⁶	3; 37:10 ⁶	4; 18:10 ⁶	3; 83:10 ⁶
performanceC _y	4; 38:10 ⁵	3; 60:10 ⁵	3; 79:10 ⁵	5; 43:10 ⁵
performanceC _n	4; 83:10 ⁶	3; 84:10 ⁶	3; 25:10 ⁶	6; 59:10 ⁶
performance globale	,167:10 ⁵	1; 44:10 ⁵	1; 51:10 ⁵	2; 16:10 ⁵
	$p = 0 :s^{-1}, r = 25 :s^{-1}$			
nb neurones	41	32	46	60
performanceC _l	1; 47:10 ⁶	3; 43:10 ⁶	4; 05:10 ⁶	4; 47:10 ⁶
performanceC _y	3; 89:10 ⁵	4; 17:10 ⁵	4; 12:10 ⁵	5; 41:10 ⁵
performanceC _n	4; 11:10 ⁶	4; 21:10 ⁶	3; 20:10 ⁴	6; 18:10 ⁶
performance globale	,148:10 ⁵	1; 64:10 ⁵	1; 62:10 ⁵	2; 16:10 ⁵

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

L'erreur quadratique moyenne d'entraînement est inférieure à $25:10^{-4}$, valeur de référence, pour l'ensemble des réseaux. En outre, pour chacun des cinq cas 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 montre que les réseaux obtenus sont satisfaisants.

L'erreur quadratique moyenne d'entraînement a pu être abaissée à 10^{-5} pour les réseaux du modèle longitudinal (à l'exception de la région 4) et à $2 \cdot 10^{-5}$ pour ceux du modèle latéral, sans pour autant entraîner une explosion du nombre de neurone sur la couche cachée des réseaux. Soit une amélioration de facteur 10 !

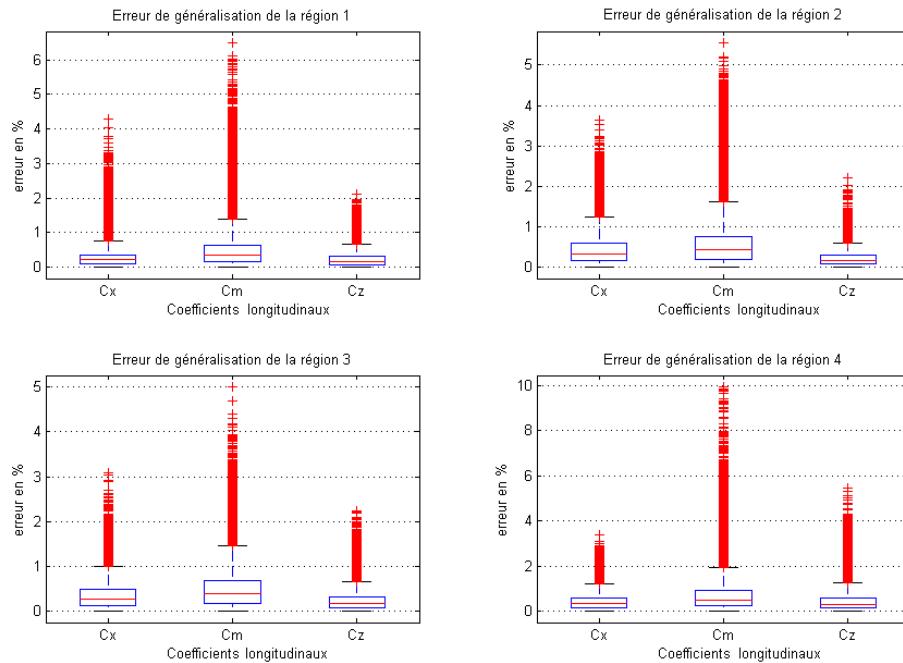


Figure 3.19 Coefficients longitudinaux - Erreur d'entraînement en %

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. Pres de 100% des données se trouvent entre les moustaches des boîtes, les bords des rectangles mettent en évidence le 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.

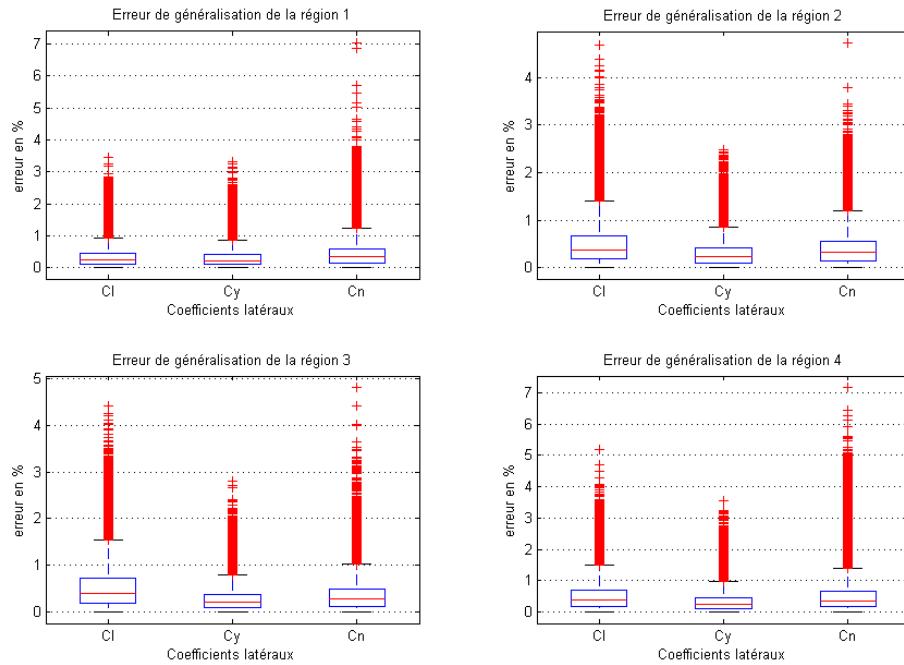


Figure 3.20 Coefficients latéraux - Erreur d'entraînement en %

Les coefficients de force C_x , C_y et C_z sont identiques à environ 1% près, tandis que les coefficients de moment C_l , C_m et C_n sont identiques à 1,5% près. Bien que ces performances soient légèrement supérieures aux attentes, elles ne sont pas satisfaisantes. Les performances observées ici sont d'ailleurs comparables à la performance des méthodes classiques, qui est de 25% pour le filtrage de Kalman étendu par exemple (Jategaonkar, 2006).

Il faut comparer les résultats obtenus avec ceux de la structure en hypercubes, la sortie du module d'identification pour $q = 0$: s^{-1} , $\alpha = 0$, $\beta_e = 2;2441$, et $l_{ref} = 25$ 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 simple de l'analyse critique du paragraphe 3.3.2 où seul l'angle d'attaque était amené à varier.

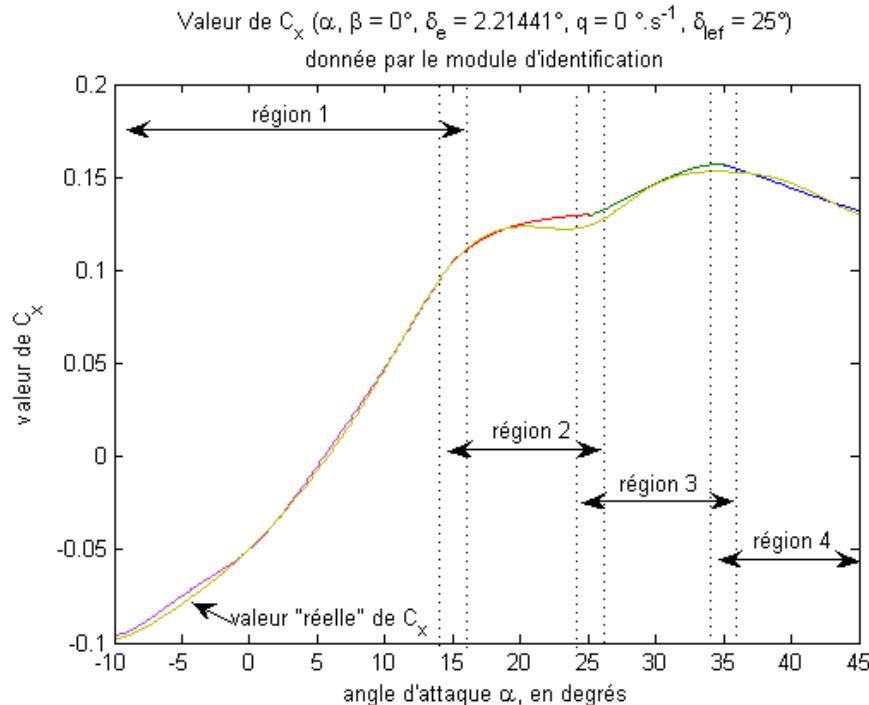


Figure 3.21 Comportement du module d'identification

Les figures 3.22 à 3.25 permettent de visualiser le comportement des réseaux 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.

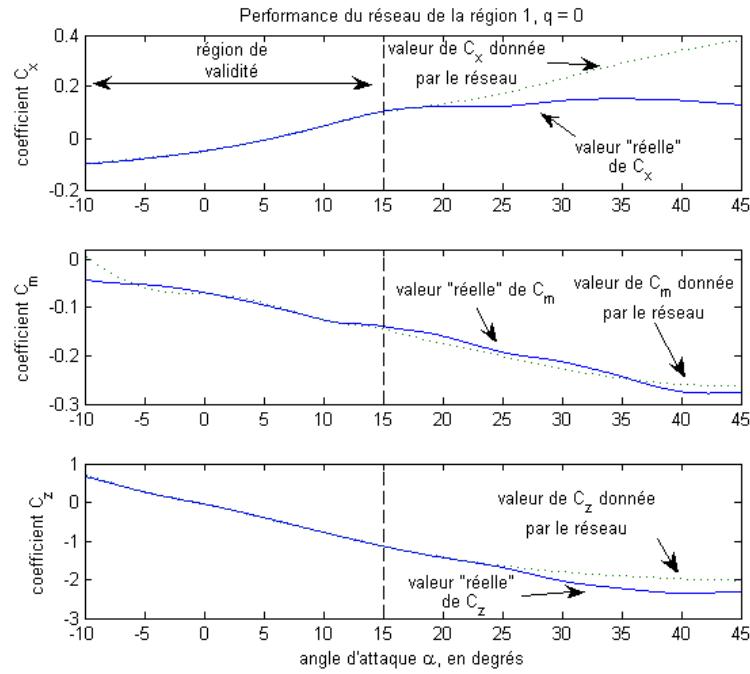


Figure 3.22 Performance du réseau de la région 1 pour $q = 0$

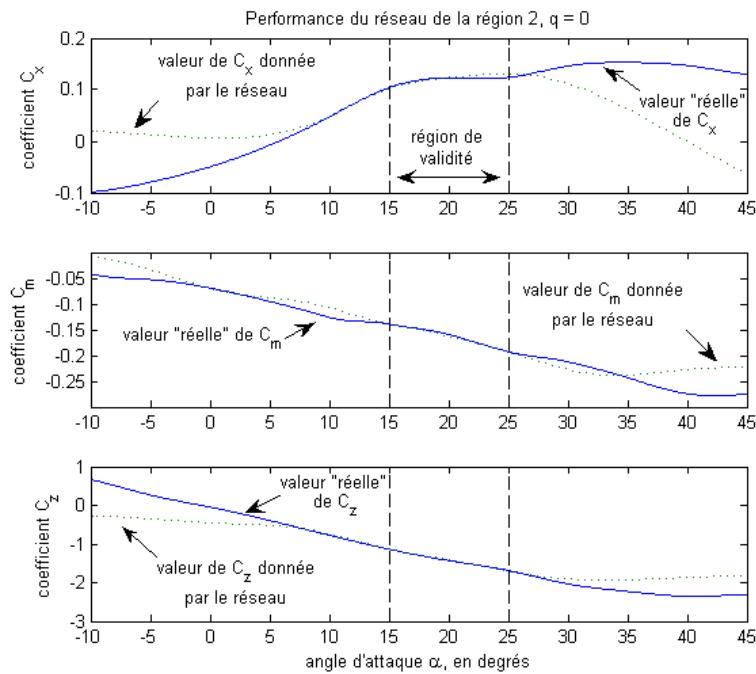


Figure 3.23 Performance du réseau de la région 2 pour $q = 0$

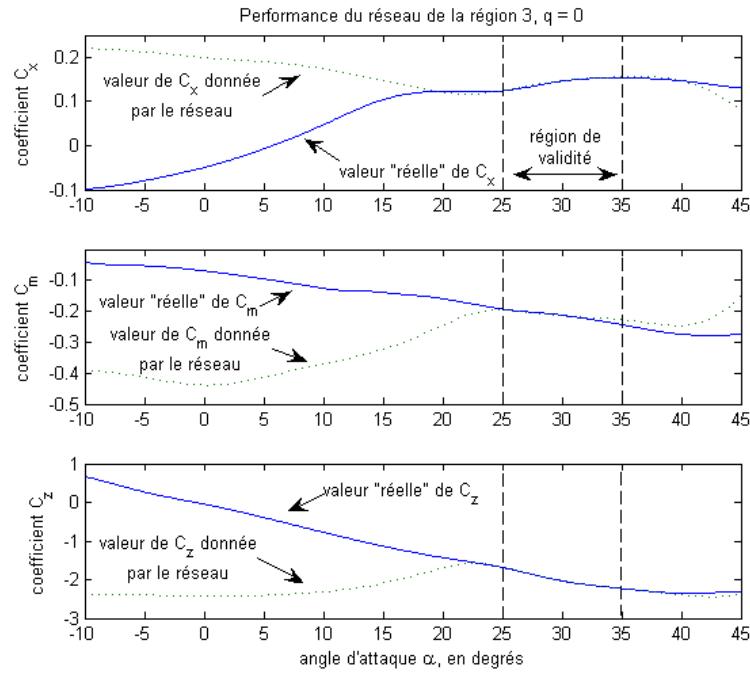


Figure 3.24 Performance du réseau de la région 3 pour $q = 0$

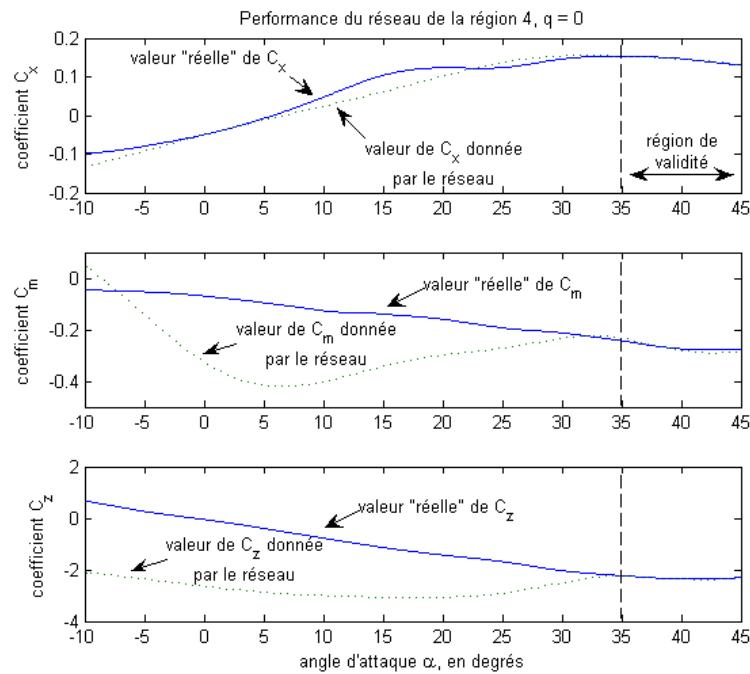


Figure 3.25 Performance du réseau de la région 4 pour $q = 0$

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 reconnaissables, 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'enranger 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%. 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 manuvrabilité, sont d'abord mises en évidence. La structure globale du contrôleur est ensuite explicitée. Après avoir rapidement exposé la théorie du contrôle utilisée, les différentes composantes du contrôleur sont traitées en détail.

4.1 Qualités de manuvrabilité

Les spécifications de qualités de manuvrabilité, ou "ying 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 influencée 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 manuvrabilité, 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.

A chaque niveau correspondent 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

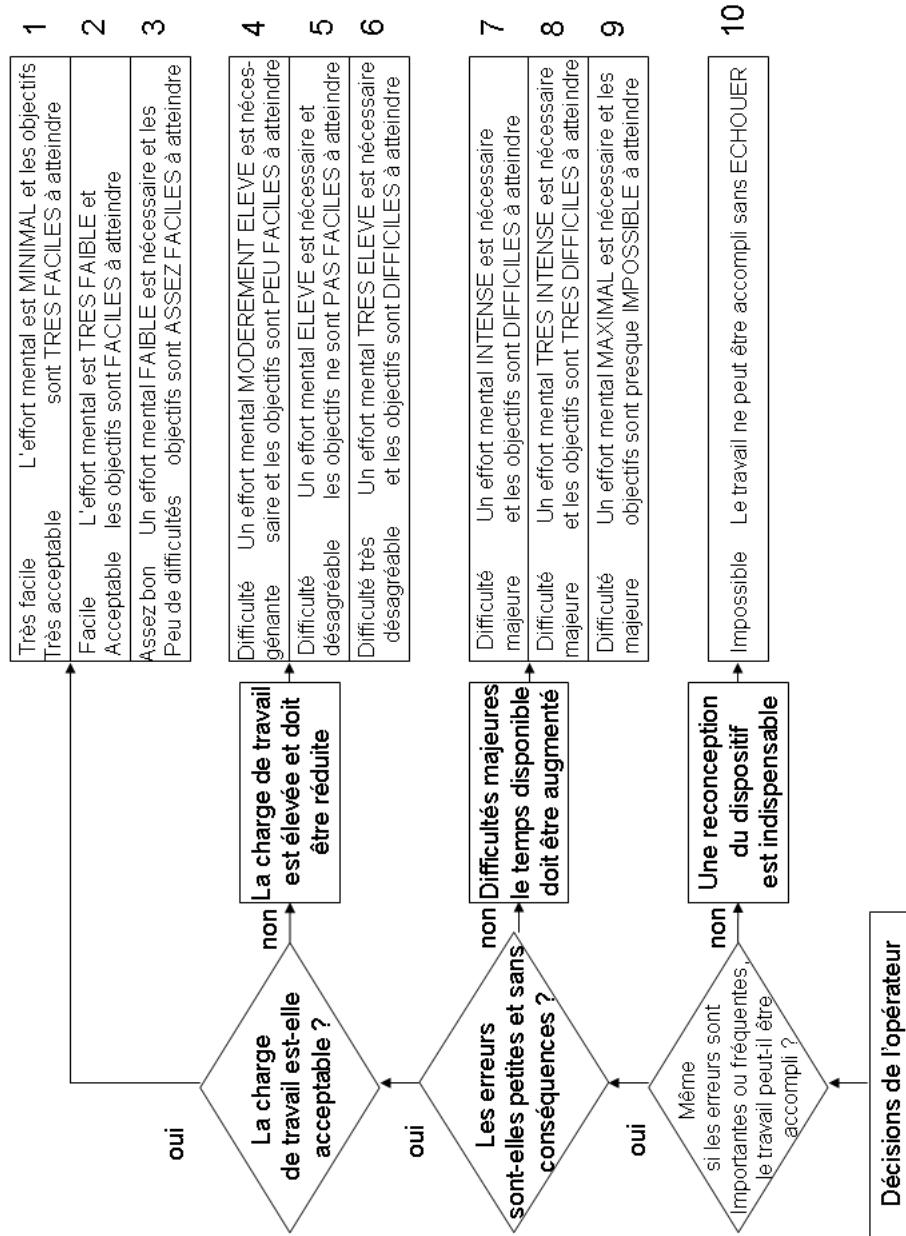


Figure 4.1 Echelle de Cooper-Harper

Tableau 4.1 Niveaux de man uvrabilité

Niveau	Description
I	Qualités de man uvrabilité totalement adéquates pour la phase de la mission considérée.
II	Qualités de man uvrabilité adéquates pour accomplir la phase de la mission considérée lorsque l'effort du pilote est quelque peu augmenté ou que l'exactitude de la mission se dégrade, ou les deux.
III	Qualités de man uvrabilité telles que la mission peut être contrôlée de manière sécuritaire, mais l'effort du pilote est excessif ou l'exactitude 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 man uvrabilité 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 man uvrabilité faible à modérée, tels que les avions cargo ou de transport lourds, les bombardiers lourds, etc.
IV	Avions hautement manoeuvrants, 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-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 graduées 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 déclenchements 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 phugode et rapide, ainsi que de la pulsation propre de ce dernier.

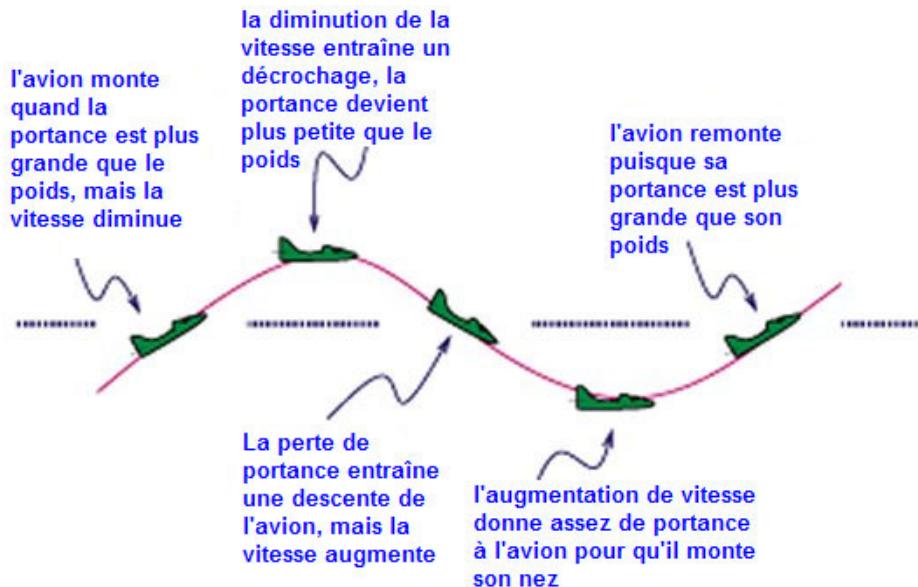


Figure 4.2 Mouvement du mode phugode

Tableau 4.4 Coefficient d'amortissement pour le mode phugode

Niveau I	ρ_h	0; 04
Niveau II	ρ_h	0; 00
Niveau III	T_2	55s

Le mode phugode 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 et de la pente = . La figure 4.2 illustre le mouvement de ce mode.

Les qualités de manuvrabilité requièrent un coefficient d'amortissement ρ_h conforme aux spécifications du tableau 4.4. En niveau III, le mode phugode est supposé instable, soit $\rho_h < 0$; 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 phugode.

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

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 , le taux de tangage et le facteur de charge n_z varient rapidement tandis que la vitesse longitudinale 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 ζ_{mr} , elles le sont sur un critère additionnel nommé "Control Anticipation Parameter" (CAP). Le CAP est défini comme le ratio suivant:

$$CAP = \frac{\zeta_{mr}^2}{n_z} \quad (4.2)$$

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. Il affecte principalement le taux de roulis p , tandis que l'angle de dérapage , l'azimut et la vitesse angulaire de lacetr restent quasiment constants.

Tableau 4.6 CAP pour le mode rapide

cat.	A	B	C
Niv. I	$0;28 < \text{CAP} < 3;6$ $\lambda_r = 1;0$	$0;085 < \text{CAP} < 3;6$	$0;16 < \text{CAP} < 3;6$ $\lambda_r = 0;7$
Niv. II	$0;16 < \text{CAP} < 10;0$ $\lambda_r = 0;6$	$0;038 < \text{CAP} < 10;0$	$0;096 < \text{CAP} < 10;0$ $\lambda_r = 0;4$
Niv. III	$0;16 < \text{CAP}$	$0;038 < \text{CAP}$	$0;096 < \text{CAP}$

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 décrit principalement l'angle de dérapage et le taux de lacet, 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 manuvrabilité latérales s'expriment en terme de constante de temps maximale du mode de roulis (voir tableau 4.7), de pulsation propre ω_h et de coefficient d'amortissement ζ_h 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).

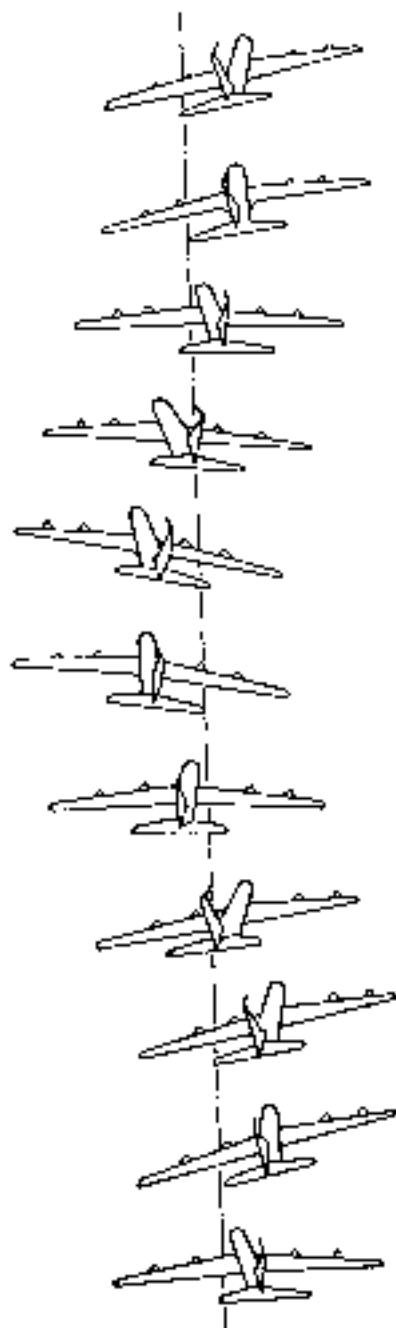


Figure 4.3 Mouvement du roulis hollandais

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 manuvrabilité du roulis hollandais

niveau	cat.	classe	min r_h	min $r_h ! r_h$	min $! r_h$
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 linearisation 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 :

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

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 inversible, $\text{sig}(x)$ 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 à ne en la commande, or le modèle général de la dynamique d'un aéronef est :

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

où f est une fonction non linéaire.

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

$$C_a \approx C_{a_0}(\cdot) + C_{a_e}(\cdot) e$$

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)):

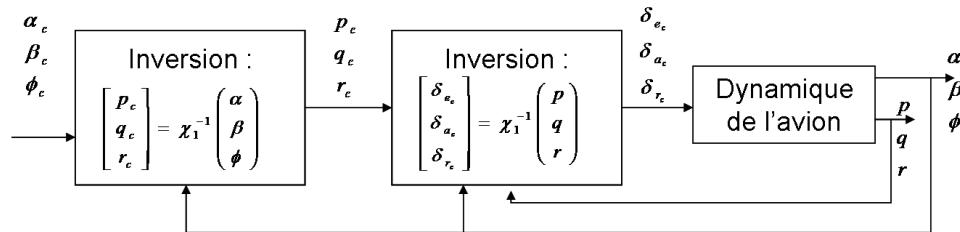


Figure 4.4 Architecture à séparation temporelle de la dynamique

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.

la dynamique des angles (α, β) est plus rapide que celle de la vitesse \dot{q} , de l'assiette de tangage et de l'azimut

Ces hypothèses permettent de séparer la conception du contrôleur en 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 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 :

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

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

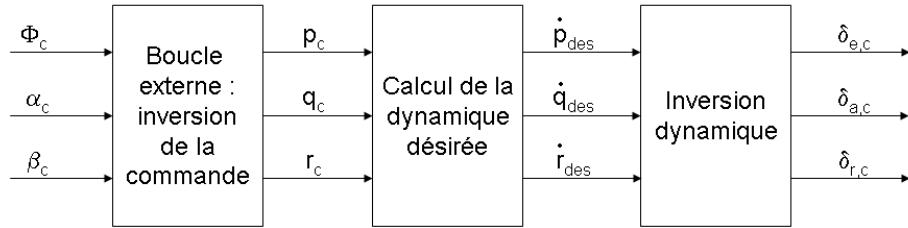


Figure 4.5 Architecture alternative

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 $[\Phi_c; \alpha_c; \beta_c]^T$.

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 γ_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 fournit 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 découpé pour chaque variable d'état x_i :

$$\underline{x}_{i\text{des}} = K_{i_p} (x_{i_c} - x_i) \quad (4.7)$$

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

Le gain K_p xe 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 $\underline{x}_i = x_{i\text{des}}$. 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 manuvrabilité, alors un système d'ordre plus élevé doit être utilisé. Parmi ces solutions, un retour d'état découpé proportionnel intégral est largement répandu :

$$\underline{x}_{i\text{des}} = K_{i_p} (x_{i_c} - x_i) + K_{i_i} \int (x_{i_c} - x) dt \quad (4.9)$$

Le choix des gains K_{i_p} et K_{i_i} permet de régler l'amortissement ζ et la pulsation propre ω_n de la variable d'état x_i du système.

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

$$x_{i\text{des}} = K_{i_B} - \frac{1}{2}x_{i_c} - x_i + \frac{K_{i_B}^2}{4} Z (x_{i_c} - x_i) dt \quad (4.10)$$

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

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

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

Le gain peut ainsi être vu 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 $\omega_n = \frac{1}{2}K_{i_B}$ et l'amortissement est $\zeta = 1$.

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 contrôle 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 manuvrabilité 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 et δ_a pour l'inversion lente et V_t pour l'autothrottle. 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 $\dot{\gamma}$ et l'angle d'attaque α , tandis que le mode phugode agit sur la vitesse V_t . Le mode de roulis amorti influence sur le taux de roulis \dot{p} et le mode spiral sur l'assiette de roulis \dot{q} . Quant au roulis hollandais, il affecte à la fois l'angle de dérapage $\dot{\beta}$ 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 γ_c .

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

$$\begin{matrix} 2 & 3 & 2 \\ p & q & r \\ \hline 6 & 6 & 6 \\ 6 & 6 & 6 \\ 4 & 4 & 5 \end{matrix} = \begin{matrix} 1 & 0 & \sin \gamma \\ 0 & \cos \gamma & \cos \alpha \\ \sin \beta & \cos \beta & \cos \alpha \end{matrix} \begin{matrix} 3 & 2 & 3 \\ \hline 7 & 6 & 7 \\ 7 & 6 & 7 \\ 5 & 4 & 5 \end{matrix} \quad (4.12)$$

On peut alors remplacer p , q et r par les valeurs de commande recherchées et γ et β par leurs valeurs commandées :

$$\dot{\alpha}_c = \frac{d}{dt} \alpha_c, \quad \dot{\beta}_c = \frac{d}{dt} \beta_c, \quad \dot{\gamma}_c = \frac{d}{dt} \gamma_c$$

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

$$\begin{matrix} 2 & 3 & 2 \\ p_c & q_c & r_c \\ \hline 6 & 6 & 6 \\ 6 & 6 & 6 \\ 4 & 4 & 5 \end{matrix} = \begin{matrix} 1 & 0 & \sin \gamma \\ 0 & \cos \gamma & \cos \alpha \\ \sin \beta & \cos \beta & \cos \alpha \end{matrix} \begin{matrix} 3 & 2 & 3 \\ \hline 7 & 4 & 5 \\ 7 & 4 & 5 \\ \tau & \tau & \tau \end{matrix} \quad (4.13)$$

4.2.5 Un système à ne pas rapporter la commande

L'inversion dynamique requiert un système à ne pas rapporter 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 a une couche cachée ayant pour entrée le vecteur $[; ; e]$ 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 :

$$\begin{matrix} & & & 2 & & 3 \\ & & & \text{---} & & \text{---} \\ & & & 6 & & 1 & 2 & 3 & z \\ & & & 6 & & 6 & 1 & 7 & z \\ & & & 6 & & 6 & 7 & 7 & z \\ & & & 4 & & 6 & 7 & 7 & z \\ & & & C_x & & 6 & 7 & 7 & z \\ & & & 6 & & 6 & 7 & 7 & z \\ & & & 6 & & 6 & 7 & 7 & z \\ & & & 4 & & 6 & 7 & 7 & z \\ & & & C_m & & 6 & 7 & 7 & z \\ & & & 6 & & 6 & 7 & 7 & z \\ & & & 4 & & 6 & 7 & 7 & z \\ & & & C_z & & 6 & 7 & 7 & z \\ & & & 4 & & 6 & 7 & 7 & z \\ & & & & & 4 & 5 & 5 & z \\ & & & & & & & & e \end{matrix} \quad (4.14)$$

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 σ est la fonction sigmoïdale tangente hyperbolique.

De manière analogue un réseau dont le vecteur d'entrée est $[; ; a; r]$ et celui de sortie $[C_l; C_y; C_n]$ est représenté par une fonction de même type aW_{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 :

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

et sa dérivée est donc :

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

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 -ème sortie par rapport à la i -ème entrée est alors :

$$\frac{\partial y_j}{\partial x_i} = w_{out,j} \cdot w_{in,i} : \begin{matrix} & 2 & & 2 & 3 & 3 \\ & \vdots & & \vdots & \vdots & \vdots \\ \partial' & w_{out,j} & w_{in,i} : & \partial' & b_n & W_{in} \\ & \vdots & & \vdots & \vdots & \vdots \\ & 4 & & 4 & 5 & 5 \\ & & & & x_m & \end{matrix} \quad (4.17)$$

où $w_{out,j}$ est la j -ème ligne de W_{out} , $w_{in,i}$ est la i -ème colonne de W_{in} et : représente la multiplication matricielle terme à terme aussi appelée "dot-product".

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

Si la linearisation des coefficients est choisie, alors un coefficient $C_a(\alpha; \beta; e)$ est approximé par son développement de Taylor au premier ordre autour d'un point d'équilibre $[0; 0; e_0]$:

$$C_a \approx C_a(0; 0; e_0) + \frac{\partial C_a}{\partial \alpha} \alpha + \frac{\partial C_a}{\partial \beta} \beta + \frac{\partial C_a}{\partial e} e \quad (4.18)$$

$$\alpha = 0, \beta = 0 \text{ et } e = e - e_0.$$

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

méthode n'est donc pas retenue. Toutefois, une linearisation instantanée ~~à 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 :

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

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 à ne en la commande, il n'y a donc ni approximation à faire, ni calcul fastidieux des dérivées aérodynamiques. Les réseaux de neurones fournissent une ~~en~~fonction analytique des coefficients aérodynamiques qui est alors utilisée par un algorithme de programmation linéaire pour calculer la valeur des actions 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 V_t par celle désirée $V_{t_{des}}$:

$$T = \frac{m}{\cos \gamma \cos \phi} f V_{t_{des}} + \frac{qS}{m} \begin{pmatrix} C_x \cos \gamma \cos \phi & C_y \sin \gamma & C_z \sin \gamma \cos \phi \\ g(\cos \gamma \cos \phi \sin \psi + \sin \gamma \cos \phi \sin \psi) & \sin \gamma \cos \psi & \sin \gamma \cos \phi \cos \psi \end{pmatrix} g \quad (4.20)$$

En choisissant la pulsation propre ν et l'amortissement ζ conformes aux spécifications du mode phugode, la dynamique suivante permet d'assurer la qualité de manuvrabilité requise :

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

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 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 appliquée à un modèle longitudinal simplifié : celui-ci est linéaire et invariant dans le temps.

Soit le modèle longitudinal suivant :

$$\begin{aligned}\dot{q} &= Z + q \\ \ddot{q} &= M_e + M_q q + M_{qe}\end{aligned}\quad (4.22)$$

$x = [; q]^T$ est le vecteur d'état du système et $u = e$ la commande.

Application à la boucle interne : l'inversion rapide

Le système est déjà à ne en la commande, il n'y a donc pas lieu d'étudier la linearisation du coefficient aerodynamique en moment de tangage C_m .

On suppose que M_e est non nul. On a alors :

$$e = \frac{1}{M_e} [q \quad M \quad M_q q] \quad (4.23)$$

On remplace ensuite par la dynamique désirée dans l'équation 4.23 :

$$e = \frac{1}{M_e} [q_{des} \quad M \quad M_q q] \quad (4.24)$$

La dynamique désirée est choisie de forme proportionnelle :

$$q_{des} = K_q (q_c - q) \quad (4.25)$$

La dynamique de q doit répondre aux spécifications de qualité de manuvrabilité 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 réglé par un premier ordre de constante de temps T_e :

$$\dot{e} = \frac{1}{T_e} (e_{com} - e) \quad (4.26)$$

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

variable suivant :

$$\begin{array}{ccccccccc}
 2 & 3 & 2 & & 3 & 2 & 3 & 2 & 3 \\
 6 & -7 & 6 & Z & 1 & 0 & 7 & 6 & 7 \\
 6 & 6 & 6 & M & M_q & M_e & 6 & 6 & 6 \\
 4 & q & 5 & = & 4 & 5 & 4 & 5 & 4 \\
 \hline
 e & & & & \frac{M}{T_e M_e} & \frac{K_q + M_q}{T_e M_e} & \frac{1}{T_e} & e & \frac{K_q}{T_e M_e}
 \end{array} \quad (4.27)$$

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

$$\frac{K_q(s - Z)}{T_e s^3 + (1 - M_q T_e - Z T_e) s^2 + (K_q - Z (1 - M_q T_e) - M T_e) s + Z (K_q + M_q)} \quad (4.28)$$

Le lieu des racines permet alors de chercher 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 dépassements 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 :

$$q_{des} = 2 \omega_q! q (q_c - q) + !^2_q (q_c - q) \quad (4.29)$$

où $!_q$ est la pulsation propre du mode rapide et ω_q son amortissement, tous dans les plages autorisées par les qualités de manuvrabilité.

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 :

$$\begin{matrix} 2 & 3 & 2 \\ 6 & -7 & 6 \\ 6 & 7 & 6 \\ \hline 4 & 5 & 4 \end{matrix} Z + \begin{matrix} 1 & 0 \\ 7 & 6 \\ 7 & 6 \\ \hline 5 & 4 \end{matrix} q \begin{matrix} 3 & 2 & 3 \\ 6 & 7 & 6 \\ 6 & 7 & 6 \\ \hline 4 & 5 & 4 \end{matrix} + \begin{matrix} 2 & 3 \\ 6 & 0 \\ 6 & 0 \\ \hline 4 & 5 \end{matrix} e_c = \begin{matrix} 0 & 0 & \frac{1}{T_e} \\ 0 & 0 & e \end{matrix} \quad (4.30)$$

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

$$q = M_Z + M_q q + M_{qq} \frac{M_e}{T_e} e + \frac{M_e}{T_e} e_c \quad (4.31)$$

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

$$e_c = \frac{T_e}{M_e} (q_{des} - M_Z - M_q q - M_{qq}) + e \quad (4.32)$$

En choisissant 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 "trimme" autour de la position d'équilibre suivante:

$V_t = 150 \text{ m:s}^{-1}$ et $h = 5000 \text{ m}$. La figure 4.6 présente le schéma bloc de la simulation.

On a $M = 5; 1047$, $M_q = 3; 1226$ et $M_e = 0; 38014$. De plus à l'équilibre $= 0; 0147 \text{ rad}$ et $n_z = 1$.

L'amortissement du mode rapide désiré est défini à $\zeta_{mr_{des}} = 0; 7$ (valeur choisie arbitrairement dans la plage $[0; 35; 1; 3]$) et le CAP désiré est $1; 5$ (choisi 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 :

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

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

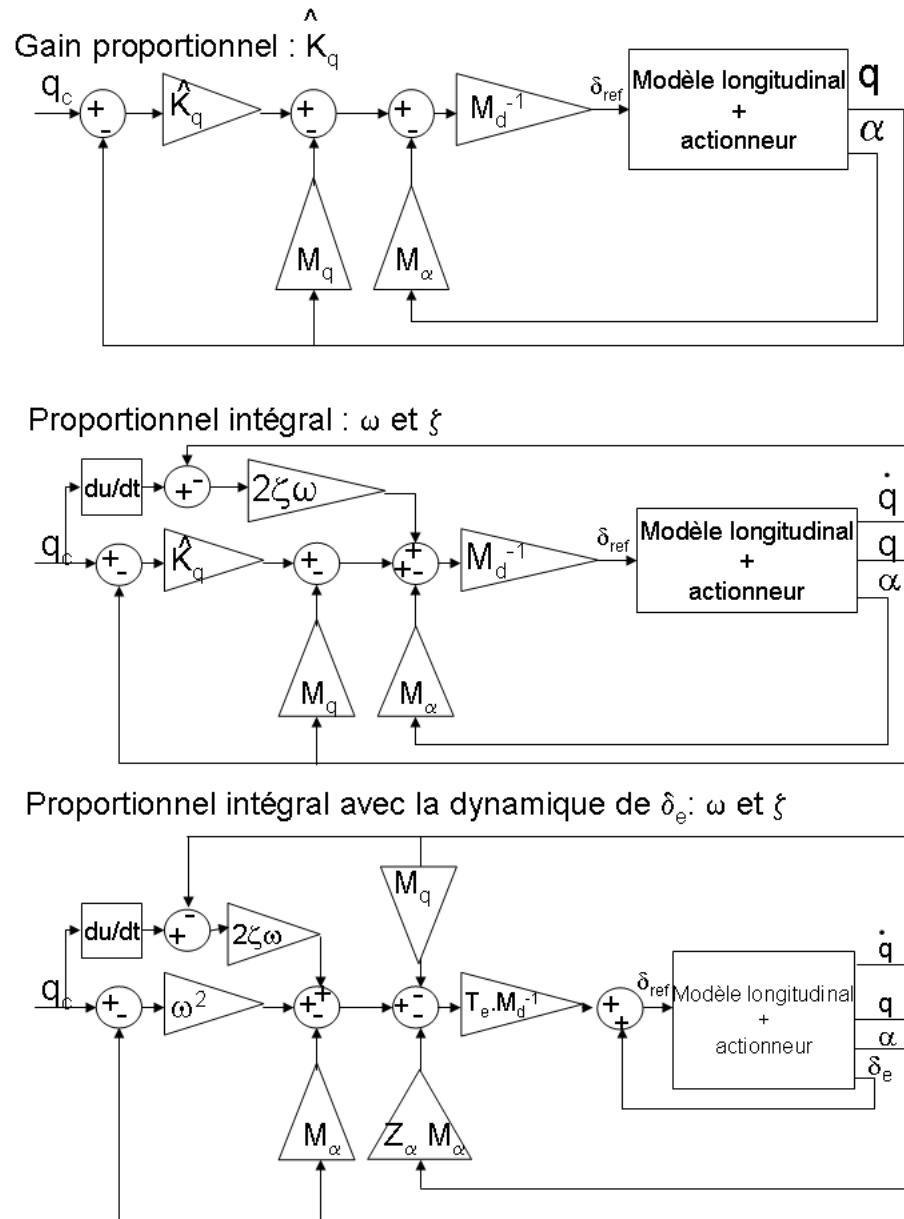


Figure 4.6 Schémas bloc comparant trois configurations de commande par inversion dynamique

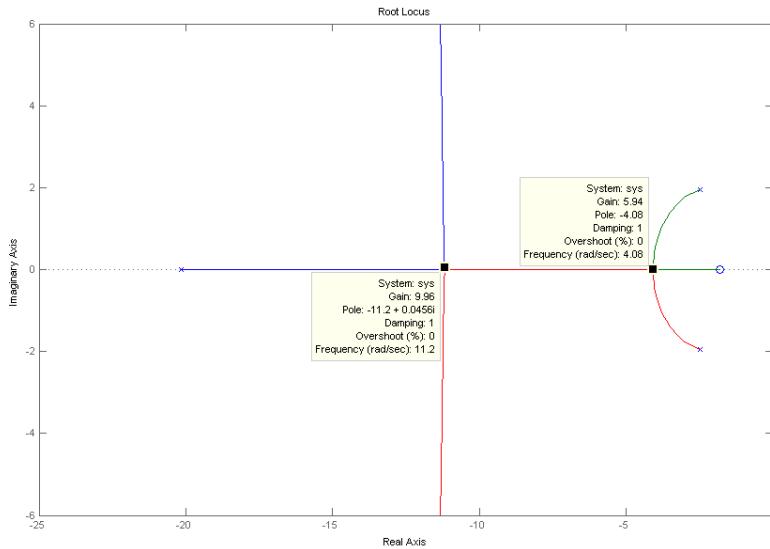


Figure 4.7 Lieu des racines pour la première méthode

La figure 4.7 met en évidence le lieu des racines. Les valeurs désirées 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_a et 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 :

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

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 $s^2 + 2\zeta_{mr}s + \zeta_{mr}^2$, on obtient

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

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 :

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

En qualité de manuvrabilité de niveau I, le CAP doit être inférieur à 3.6. La pulsation propre maximale dans la condition considérée est donc 16.66 rad:s^{-1} . Cependant le gain calculé avec l'équation 4.34 induit une pulsation propre de 16.66 rad:s^{-1} . Il est donc nécessaire de modifier le choix de l'amortissement : $\omega_{mr,des;mod} = 1$, ce qui entraîne $\omega_{mr,des;mod} = 11.67 \text{ rad:s}^{-1}$.

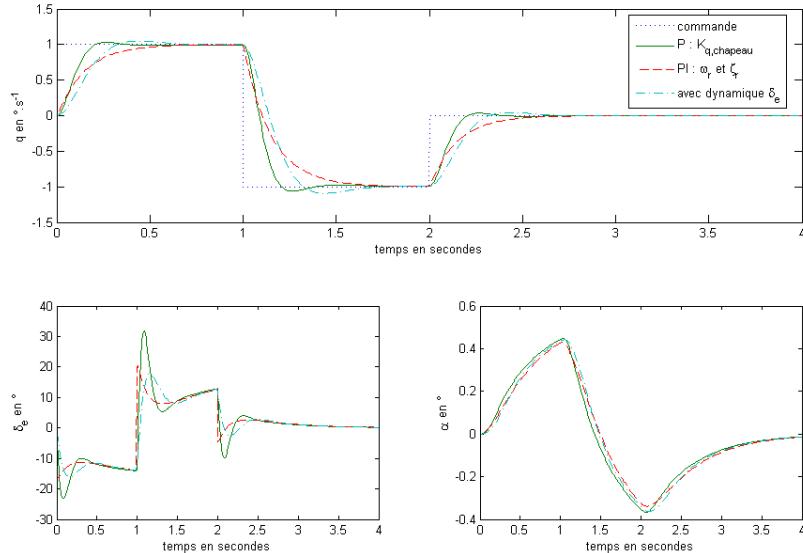


Figure 4.8 Réponse en tangage suite à un doublet de $1:s^{-1}$ en commande

On observe (voir figure 4.8) la réponse en tangage ainsi que l'effort de corde associé, suite à un doublet d'une amplitude de $1:s^{-1}$ 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 désiré.

Cependant elle présente une erreur non nulle en régime permanent

$$e_{\text{perm}} = \frac{M_q}{Z K_q + M_q} \quad (4.36)$$

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 $t = 1$ s. 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 détection 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 fournit directement la commande en assiette de tangage. La commande en taux de tangage est alors directement obtenue par l'équation suivante :

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

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 est alors :

$$\dot{\alpha} = Z - q_c \quad (4.38)$$

L'inversion de la dynamique donne :

$$q_c = \alpha_{\text{des}} - Z \quad (4.39)$$

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

$$\begin{aligned} & \stackrel{8}{\geq} \alpha_{\text{des}} = K (\alpha_c) \\ & \stackrel{>}{\geq} \alpha_{\text{des}} = K \left(\frac{1}{2} \alpha_c \right) + \frac{K^2}{4s} (\alpha_c) \end{aligned} \quad (4.40)$$

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 égale à K ($0; 5K$ respectivement). Le temps de réponse en angle d'attaque devant être plus long que celui en taux de tangage pour que l'hypothèse de séparation temporelle soit valide, on choisit ! inférieure 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 le effort sur l'actionneur.

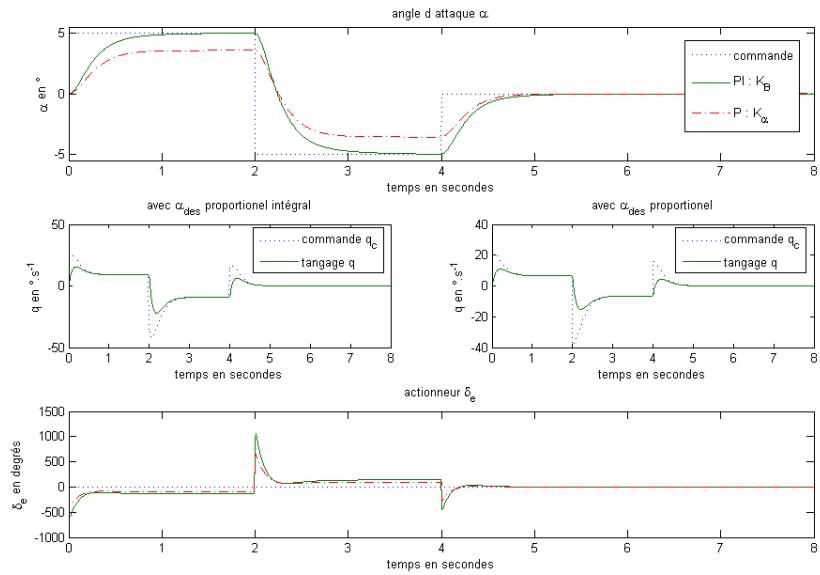


Figure 4.9 Réponse suite à un doublet de 5 en commande

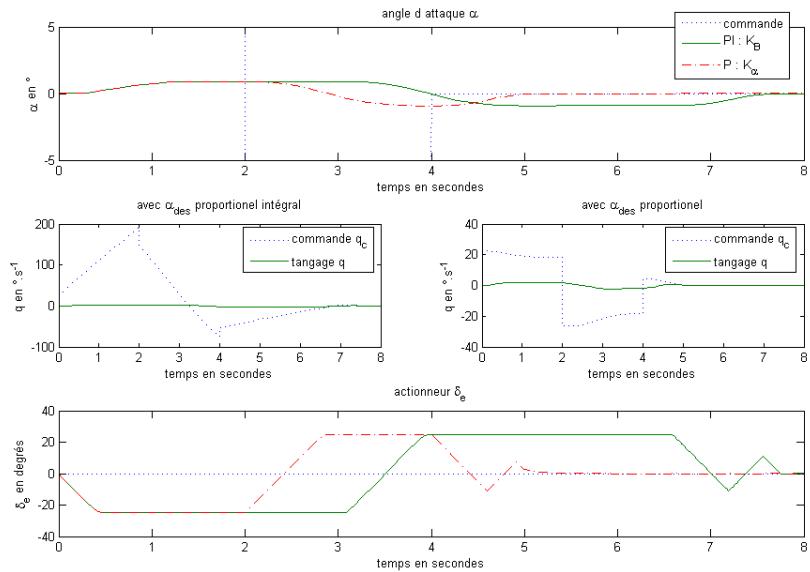


Figure 4.10 Réponse avec saturation suite à un doublet de 5 en commande

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'élevateur ne peut pas dépasser 25 sur le F-16, cette limite est cependant atteinte avec les deux types de contrôle. Le module 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 permettent d'adopter 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 concue : elle prend les vitesses angulaires de roulis $\dot{\phi}$, de tangage $\dot{\theta}$ et de lacet $\dot{\psi}$ en entrée et fournit la commande à appliquer sur les surfaces de contrôle de l'avion (δ_e pour l'élevateur, δ_a pour les ailerons et δ_r pour le gouvernail). La boucle externe est ensuite concue. Elle a pour entrée les angles d'attaque α et de dérapage γ , et l'assiette de roulis β ; les sorties sont les commandes en vitesses angulaires $\dot{\phi}$, $\dot{\theta}$ et $\dot{\psi}$.

Conception de la boucle interne

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

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

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

$$\begin{aligned} C_l &= C_{l_0}(\cdot; \cdot; a; r) + \frac{pb}{2V_t} C_{l_p}(\cdot; \cdot; a; r) + \frac{rb}{2V_t} C_{l_r}(\cdot; \cdot; a; r) \\ C_m &= C_{m_0}(\cdot; \cdot; e) + \frac{qc}{2V_t} C_{m_q}(\cdot; \cdot; e) \\ C_n &= C_{n_0}(\cdot; \cdot; a; r) + \frac{pb}{2V_t} C_{n_p}(\cdot; \cdot; a; r) + \frac{rb}{2V_t} C_{n_r}(\cdot; \cdot; a; r) \end{aligned} \quad (4.42)$$

Les qualités de manuvrabilité 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

$$\begin{aligned} p_{des} &= !_p (p_c - p) \\ q_{des} &= 2 !_q (q_c - q) + !_q^2 (q_c - q) \\ r_{des} &= !_r (r_c - r) \end{aligned} \quad (4.43)$$

où $!_p$ est conforme à la bande passante du mode de roulis amorti, à celle du roulis hollandais et ($_q; !_q$) aux caractéristiques du mode rapide longitudinal.

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

$$\begin{aligned} \bar{e} &= \frac{1}{T_e} (e_c - e) \\ \bar{a} &= \frac{1}{T_a} (a_c - a) \\ \bar{r} &= \frac{1}{T_r} (r_c - r) \end{aligned} \quad (4.44)$$

Lorsque les délections des surfaces de contrôle sont prises comme constante 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 \dot{q}_c , \dot{a}_c et \dot{r}_c apparaissent. L'inversion des équations obtenues et le remplacement de \dot{q} et \dot{r} par les dynamiques désirées donne la relation suivante entre les commandes et les dynamiques désirées :

au

$$\begin{aligned}
 p_1 &= \bar{q}Sb c_3 C_{l_a} + c_4 C_{n_a} \\
 p_2 &= \bar{q}Sb c_3 C_{l_r} + c_4 C_{n_ar} \\
 p_3 &= \bar{q}Sb c_3 \frac{@C_0}{@} + c_4 \frac{@C_0}{@} \\
 p_4 &= \bar{q}Sb c_3 \frac{@C_0}{@} + c_4 \frac{@C_0}{@} \\
 p_5 &= \frac{\bar{q}Sb^2}{2V_t} c_3 C_{l_p} + c_4 C_{n_p} \\
 p_6 &= c_4 h_{moteur} \\
 p_7 &= \frac{\bar{q}Sb^2}{2V_t} (c_3 C_{l_r} + c_4 C_{n_r})
 \end{aligned} \tag{4.46}$$

$$\begin{aligned}
p_8 = & \bar{q}Sb c_3 \frac{pb}{2V_t} \frac{@C_p}{@} - + \frac{@C_p}{@} - + \frac{rb}{2V_t} \frac{@C}{@} - + \frac{@C}{@} - \\
& + c_4 \frac{pb}{2V_t} \frac{@C_p}{@} - + \frac{@C_p}{@} - + \frac{rb}{2V_t} \frac{@C_r}{@} - + \frac{@C_r}{@} - \\
& + c_1 (rq + r q) + c_2 (pq + pq)
\end{aligned}$$

$$\begin{aligned}
q_1 = & \bar{q}ScC_7 \frac{@C_{m_0}}{@e} + \frac{qc}{2V_t} \frac{@C_{m_q}}{@e} \\
q_2 = & \bar{q}ScC_7 \frac{@C_{m_0}}{@} \\
q_3 = & \bar{q}ScC_7 \frac{@C_{m_0}}{@} \\
q_4 = & \frac{\bar{q}Sc^2C_7}{2V_t} C_{m_q} \\
q_5 = & c_7 h_{\text{moteur}} \\
q_6 = & \frac{\bar{q}Sc^2C_7}{2V_t} \frac{@C_{m_q}}{@} - + \frac{@C_{m_q}}{@} - + c_5 (pr + pr) - 2c_6 (pp - rr)
\end{aligned}$$

$$\begin{aligned}
r_1 = & \bar{q}Sb c_4 C_{l_a} + c_9 C_{n_a} \\
r_2 = & \bar{q}Sb c_4 C_{l_r} + c_9 C_{n_r} \\
r_3 = & \bar{q}Sb c_4 \frac{@C_0}{@} + c_9 \frac{@C_0}{@} \\
r_4 = & \bar{q}Sb c_4 \frac{@C_0}{@} + c_9 \frac{@C_0}{@} \\
r_5 = & \frac{\bar{q}Sb^2}{2V_t} c_4 C_{l_p} + c_9 C_{n_p} \\
r_6 = & c_9 h_{\text{moteur}} \\
r_7 = & \frac{\bar{q}Sb^2}{2V_t} (c_4 C_{l_r} + c_9 C_{n_r}) \\
r_8 = & \bar{q}Sb c_4 \frac{pb}{2V_t} \frac{@C_p}{@} - + \frac{@C_p}{@} - + \frac{rb}{2V_t} \frac{@C}{@} - + \frac{@C}{@} - \\
& + c_9 \frac{pb}{2V_t} \frac{@C_p}{@} - + \frac{@C_p}{@} - + \frac{rb}{2V_t} \frac{@C_r}{@} - + \frac{@C_r}{@} - \\
& + c_2 (rq + r q) + c_8 (pq + pq)
\end{aligned}$$

$$\begin{aligned}
 C_{n_a} &= \frac{\alpha_{\psi_0}}{\alpha_a} + \frac{pb}{2V_t} \frac{\alpha_{\psi_p}}{\alpha_a} + \frac{rb}{2V_t} \frac{\alpha_{\psi_r}}{\alpha_a} \\
 C_{n_r} &= \frac{\alpha_{\psi_0}}{\alpha_r} + \frac{pb}{2V_t} \frac{\alpha_{\psi_p}}{\alpha_r} + \frac{rb}{2V_t} \frac{\alpha_{\psi_r}}{\alpha_r} \\
 C_{l_a} &= \frac{\alpha_{\psi_0}}{\alpha_a} + \frac{pb}{2V_t} \frac{\alpha_{\psi_p}}{\alpha_a} + \frac{rb}{2V_t} \frac{\alpha_{\psi_r}}{\alpha_a} \\
 C_{l_r} &= \frac{\alpha_{\psi_0}}{\alpha_r} + \frac{pb}{2V_t} \frac{\alpha_{\psi_p}}{\alpha_r} + \frac{rb}{2V_t} \frac{\alpha_{\psi_r}}{\alpha_r}
 \end{aligned} \tag{4.47}$$

Cependant, autant les accélérations angulaires ψ et r sont mesurées, autant les dérivées en angles d'attaque et de dérapage – ne le sont pas.

Celles-ci pourraient être obtenues par l'itrage, mais cette approche ajoute 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 :

$$\begin{aligned}
 \dot{\psi}_0 &= \frac{1}{V_t \cos} (a_z \cos + a_x \sin) + q \tan (p \cos + r \sin) \\
 &\quad + \frac{g}{V_t \cos} (\cos \cos \cos + \sin \sin) \\
 \dot{\psi}_p &= \frac{1}{V_t} (a_x \cos \sin + a_y \cos - a_z \sin \sin) + p \sin - r \cos \\
 &\quad + \frac{g}{V_t} (\cos \sin \cos + \sin (\cos \sin - \sin \cos \cos))
 \end{aligned} \tag{4.48}$$

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

$$p_1 r_2 - p_2 r_1 \neq 0 \quad C_{n_a} C_{l_r} - C_{n_r} C_{l_a} \neq 0 \tag{4.49}$$

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 à 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 éliminés. Rappelons que le module d'identification a été conçu afin de pouvoir être adapté à une utilisation par une commande de vol reconnaissable. La commande développée ici n'a pas une telle envergure, de ce fait certaines hypothèses peuvent être levées. En effet, les données utilisées pour l'entraînement des réseaux de neurones sont telles que C_{ap} , C_{aq} et C_{ar} sont fonction uniquement de l'angle d'attaque, il est donc juste de faire les approximations suivantes :

$$\frac{\partial C_a}{\partial e} = \frac{\partial C_p}{\partial \alpha} = \frac{\partial C_r}{\partial \alpha} = \frac{\partial C_i}{\partial \alpha} = 0 \quad (4.50)$$

où $a = l, m$ ou n , $e = q$ ou r et $i = q, p$ ou r .

Cette commande a l'avantage de fournir directement la valeur des déflexions 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 peuvent pas être commandées directement et une allocation entre les commandes classiques (e , a , 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 entiers, l'inversion est directe et on obtient à partir de l'équation I.7:

$$\begin{matrix}
 & 2 & 3 & 2 & & 3 & 1 \\
 C_{Ic} & = & \frac{6}{6} & \frac{7}{6} & \bar{q}Sb_6 & 0 & \bar{q}Sb_7 \\
 & 6 & 7 & 6 & 7 & & 7 \\
 & 4C_{m_c} & = & \frac{6}{6} & \frac{7}{6} & 0 & \bar{q}Sc_6 & 0 & \frac{7}{5} \\
 & 4 & 5 & 4 & 5 & & & & 5 \\
 C_{n_c} & & 2 & 3 & 2 & & 3 & 2 & & 3 \\
 & & 2 & 3 & 2 & & 3 & 2 & & 3 \\
 & (& \frac{6}{6} & \frac{7}{6} & 0 & c_4 h_{moteur} & 0 & \frac{7}{6} & \frac{6}{6} & c_1 rq + c_2 pq \\
 & & 6 & 7 & 6 & & 7 & 6 & 7 & 7 \\
 & & 4 & 5 & 4 & 0 & c_7 h_{moteur} & \frac{5}{5} & \frac{6}{4} & c_5 pr \\
 & & & & & & & & 4 & c_6(p^2 - r^2) \\
 & r_{des} & & 0 & c_9 h_{moteur} & 0 & r & & & 5 \\
 & & & & & & & & & \\
 & & & & & & & & c_8 pq & c_2 rq
 \end{matrix} \quad (4.51)$$

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, 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 déuctions 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, l'angle de dérapage et l'assiette de roulis sont données en annexe I et rappelées ici:

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

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 :

$$\begin{array}{ccccccc}
 2 & 3 & 2 & & & 3 & 1 \\
 \frac{6}{6} p_c \frac{7}{7} & \frac{6}{6} \tan \cos & 1 + a_1 & \tan \sin \frac{7}{7} \\
 \frac{6}{6} q_c \frac{7}{7} = & \frac{6}{6} \sin + b_1 & b_2 & \cos + b_3 \frac{7}{7} \\
 4 & 4 & 5 & & & 5 & \\
 r_c & 1 & \tan \sin & \tan \cos & & & 3 \\
 2 & 3 & 2 & & & & \\
 \frac{6}{6} \frac{-des}{-des} \frac{7}{7} & a_2 + \frac{g}{V_t \cos} (\cos \cos \cos + \sin \sin) & & & & & \frac{7}{7} \\
 \frac{6}{6} \frac{-des}{-des} \frac{7}{7} & b_4 + \frac{g}{V_t} \cos \sin \cos + \frac{\sin}{V_t} (g \cos \sin \quad g \sin \cos \cos) \frac{7}{7} & & & & & \\
 -des & & & & 0 & & \\
 & & & & & &
 \end{array} \tag{4.53}$$

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

$$\begin{aligned}
 b_1 &= \frac{\bar{q}Sb}{2mV_t^2} C_{y_p} \cos \\
 b_2 &= \frac{\bar{q}Sc}{2mV_t^2} \quad C_{x_q} \cos \quad \sin \quad C_{z_q} \sin \quad \sin \\
 b_3 &= \frac{\bar{q}Sb}{2mV_t^2} C_{y_r} \cos \\
 b_4 &= \frac{\bar{q}S}{mV_t} \quad C_{x_0} + \frac{T}{\bar{q}S} \quad \cos \quad \sin \quad + C_{y_0} \cos \quad C_{z_0} \sin \quad \sin
 \end{aligned} \tag{4.55}$$

De même que dans la boucle interne, on met 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 des réseaux de neurones permet d'exprimer T en fonction de celui-ci et de la mesure de l'accélération a_x :

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

Les qualités de manuvrabilité 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 :

$$\begin{aligned}
 \dot{\gamma}_{des} &= 2 \quad ! \quad (\quad c \quad) + !^2 \int_0^Z (\quad c \quad) dt \\
 \dot{\alpha}_{des} &= 2 \quad ! \quad (\quad c \quad) + !^2 \quad (\quad c \quad) dt \\
 \dot{\psi}_{des} &= ! \quad (\quad c \quad)
 \end{aligned} \tag{4.57}$$

ou $(! ;)$ sont conformes aux spécifications du mode rapide avec $< !_q$,
 $(! ;)$ à celles du roulis hollandais avec $< !_r$,
et $!$ 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 exible, 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 manœuvrabilité. 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 nombres 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 manuvrabilité 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'exactitude de la commande. Les contraintes en amplitude et en taux de rotation de chaque surface de contrôle sont prises en compte dans l'allocation du contrôle :

$$\underline{i}_i \leq i_i \leq \bar{i}_i \text{ et } \underline{\dot{i}}_i \leq \dot{i}_i \leq \bar{\dot{i}}_i \quad (4.58)$$

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 :

$$+ t_i \leq \frac{i(t_i) - i(t_i - T)}{T} \leq + \bar{t}_i \quad (4.59)$$

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):

$$\begin{aligned} i_{\min} &= \max_{n_i} \frac{i}{n_i}; \quad i(t-T) + \frac{\dot{i}}{T} o \\ i_{\max} &= \max_{n_i} \frac{i}{n_i}; \quad i(t-T) + \frac{\dot{i}}{T} o \end{aligned} \quad (4.60)$$

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 :

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

où M est le vecteur des coefficients aérodynamiques en moment, une matrice appelée "control effectiveness matrix" et B est le vecteur des actions des surfaces de contrôle.

De nombreuses méthodes existent pour de calculer le vecteur donnant les coefficients commandes M_c , telles que la programmation linéaire, la programmation quadratique, le pseudo-inverse, le pseudo-inverse cascade ou encore le "designing". La méthode du pseudo-inverse cascade est présentée ici (Luo et al., 2004).

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

$$J = \frac{1}{2} (\alpha - \alpha_{\text{des}})^T W (\alpha - \alpha_{\text{des}}) \quad (4.62)$$

où W est une matrice de pondération et α_{des} un contrôle désiré.

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

$$= \mathbf{u}_{\text{des}} + \mathbf{W}^{-1} \mathbf{B}^T \mathbf{B} \mathbf{W}^{-1} \mathbf{B}^T^{-1} (\mathbf{M}_c - \mathbf{B} \mathbf{u}_{\text{des}}) \quad (4.63)$$

Le principe de la cascade est alors le suivant. Si aucune composante du vecteur de contrôle n'atteint ses limites, alors celui-ci est trouvé et les coefficients aérodynamiques en moment commandés \mathbf{M}_c sont atteints. Si une composante du vecteur de contrôle 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 \mathbf{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 limiter les coefficients.

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

Une linearisation 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 lineariser les coéfficients à chaque instant

$$C_a(t) = C_{a_0} + \sum_{i=1}^{X^n} \frac{\partial C}{\partial i} \cdot i \quad (4.64)$$

$$\omega_i = i - i_0.$$

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 coéfficients aérodynamiques C et les déuctions des surfaces de contrôle est variante dans le temps :

$$C(t) = B(t) \cdot (t) \quad (4.65)$$

La fonction suivante doit alors être minimisée à chaque instant sous la contrainte

$$C_c(t) = B(t) \cdot (t):$$

$$J = \frac{1}{2} \| (t)^T W(t) \cdot (t) \|^2 \quad (4.66)$$

où $W(t)$ est une matrice de pondération et $C_c(t)$ la variation des coéfficients aérodynamiques désirée.

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

$$W(i;i) = \frac{1}{\max(i)^2} \quad (4.67)$$

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

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

Divers auteurs ((Luo et al., 2004), (Keviczky & Balas, 2006) et (Kale & Chipperfield, 2002)) établissent les avantages d'utiliser des modèles linéaires 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 T_{ech} . Le problème de l'allocation est alors formulé à chaque instant nT_{ech} de la manière suivante :

$$\begin{aligned} \alpha(k+1) &= A \alpha(k) + B \alpha_c(k) \\ 2 \quad \alpha_c(k) &= 2 \alpha_c(k-1) + r(k) \\ 4 \quad C(k) &= 4 B(nT_{ech}) \alpha_c(k) \end{aligned} \quad (4.69)$$

$$\alpha(k=0) = 0.$$

Les contraintes sur α sont similaires à celles de l'équation 4.60 :

$$\begin{aligned} i_{\min}(k) &\leq \alpha_i(k) \leq i_{\max}(k) & 0 \\ i_{\min}(k) &= \max_n \frac{i}{n} \left(\alpha_i(k-1) + \alpha_i((n-1)T_{ech}) \right); \frac{T_{ech}}{n} & 0 \\ i_{\max}(k) &= \min_n \frac{i}{n} \left(\alpha_i(k-1) + \alpha_i((n-1)T_{ech}) \right); \frac{T_{ech}}{n} & 0 \end{aligned} \quad (4.70)$$

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

$$\begin{aligned} y &= C \\ z &= ; \quad (4.71) \\ u &= \end{aligned}$$

Les horizons de contrôle H_c et de prédiction 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 prédiction 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 du cas d'allocation continue précédent. La matrice de pondération Q peut également être choisie diagonale, avec ses éléments 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 sont identiques 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 ($\tilde{c}_r, \tilde{\gamma}_r, \tilde{\alpha}_r$) en fournissant une commande $(\tilde{c}_c, \tilde{\gamma}_c)$ qui minimise un critère quadratique.

Sous l'hypothèse que le retour de sortie effectue une linearisation exacte, la dynamique en $(\tilde{c}, \tilde{\gamma}, \tilde{\alpha})$ de la boucle fermée par inversion dynamique a la forme suivante

$$\begin{matrix} 2 & 3 & 2 & 3 & 2 & 3 \\ 6 & -7 & 6 & 0 & 0 & 7 & 6 & 7 & 6 & 1 & 0 & 0 & 7 & 6 & \sim_c & 7 \\ 6 & -7 & 6 & 0 & 0 & 7 & 6 & 7 & 6 & 0 & 1 & 0 & 7 & 6 & \sim_c & 7 \\ 4 & -5 & 4 & 0 & 0 & 5 & 4 & 5 & 4 & 0 & 1 & 5 & 4 & \sim_c & 5 \\ - & & - & 0 & 0 & 0 & & & & 0 & 0 & 1 & & \sim_c & \\ & & & & & & & & & & & & & & & \end{matrix} \quad (4.72)$$

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:

$$\begin{matrix} y = [; ;]^T \\ 10 & 45 \\ 30 & 30 \end{matrix} \quad (4.73)$$

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 prédition 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 découpé. Une stratégie hybride peut alors être envisagée. En effet :

Pour la coordination, la référence en angle de dérapage $\dot{\theta}_{ref}$ doit être prise nulle. Il est alors légitime de conserver la dynamique désirée de type proportionnel+integral et de fixer $\dot{\theta}_c = 0$. 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 linearisé par retour de sortie n'ont été étudiées dans le cadre de ce mémoire, et que ces deux points restent à démontrer. De 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 ℓ_1 -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 rétroaction LQR effectuée sur un nombre réduit de variables d'état.

Une solution plus simple a été adoptée ici. En effet, le module d'identification a base de réseaux de neurones identifiée, 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 diaboliques 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 Farrell et Polycarpou (2006). Parmi elles, on trouve la ℓ_1 -modification, la ℓ_∞ -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 présentée 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 maniabilité. 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

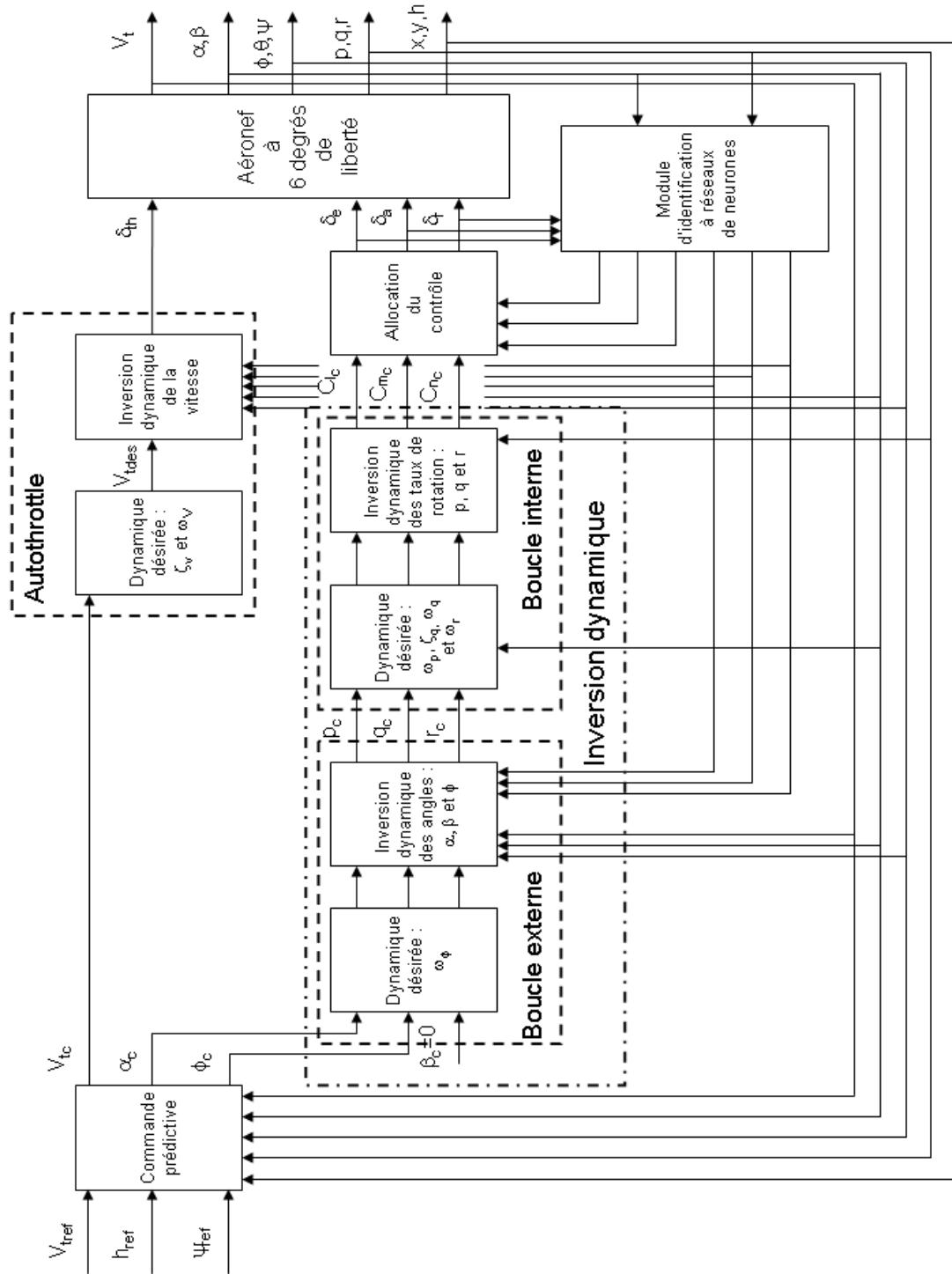


Figure 4.11 Architecture du contrôleur

réseaux de neurones développés précédemment permet notamment de garantir, 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. L'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 désirées nécessaires au calcul de l'allocation.

Une commande prédictive permet d'anticiper les changements de consigne de la trajectoire planifiée (\dot{x}_{ref} ; \ddot{x}_{ref} ; \dddot{x}_{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

RESULTATS 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 également 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 fixes :

$$V_t = 150 \text{ m:s}^{-1} \text{ et } h = 5000 \text{ m},$$

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 $x_0 = [150 \ 0,0147 \ 0 \ 0 \ 0,0147 \ 0 \ 0 \ 0 \ 0 \ 5000 \ -1,8 \ 1,38 \ 22,28 \ 0 \ 0 \ 0,84]$. Les arrondis entraînent que ces valeurs sont pas strictement identiques à la configuration de l'avion en régime permanent

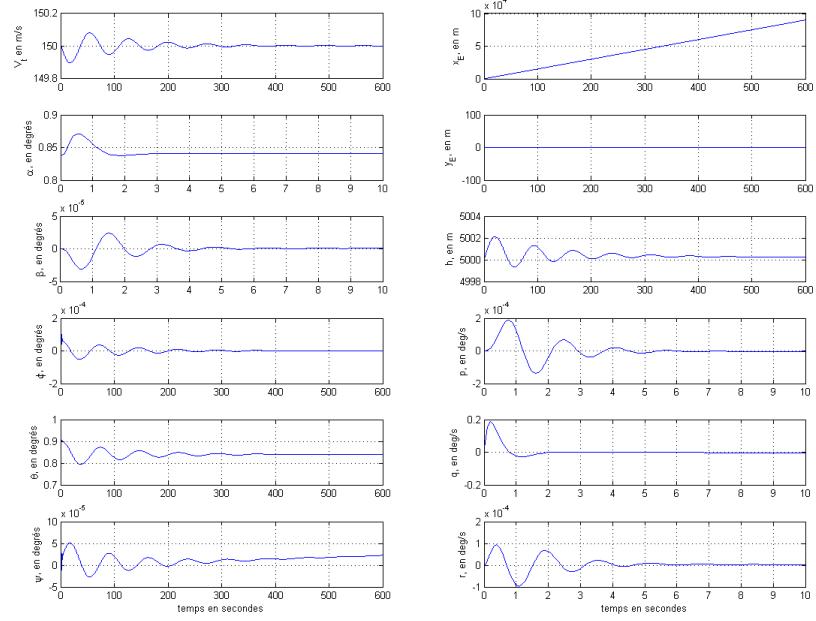


Figure 5.1 Vola ailes plates - evolution des variables détat

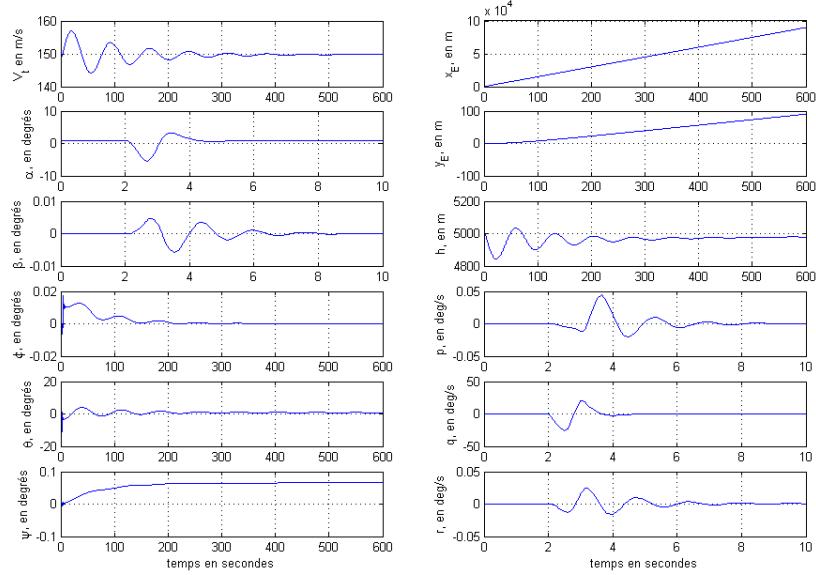


Figure 5.2 Evolution des variables détat suite à une perturbation sur α

d'au l'évolution des etats 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 et de son altitude h autour des valeurs de la condition de vol. La vitesse V oscille à la pulsation $\omega_v = 0;08 \text{ rad:s}^{-1}$ et a un dépassement maximal de 1055%. L'altitude h oscille à la pulsation $\omega_h = 0;08 \text{ rad:s}^{-1}$ et a un dépassement maximal de 1042%.

Ceci est caractéristique du mode phugode.

L'angle de dérapage est nul, tandis que l'angle d'attaque vaut $\alpha_0 = 0;0147 \text{ rad} = 0;84$ à ne pas créer une force aérodynamique qui maintienne le F-16 en équilibre.

L'angle de tangage oscille de manière analogue à la vitesse, et se stabilise à $\phi_0 = -0;84$. La pente ("path angle") $\dot{\phi}$ est ainsi nul. Les deux autres angles d'Euler, de même que les taux de rotations, sont nuls.

À ne pas vérifier la stabilité du système dans cette position d'équilibre, un doublet d'amplitude 5 et d'une durée de 1 s est appliquée à $t = 2 \text{ s}$ sur l'élévateur.

L'évolution des variables d'état est illustrée par la figure 5.2.

Le dépassement maximal en vitesse V 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 et le taux de tangage reviennent à leur point d'équilibre en 2 s ; l'angle de dérapage et les taux de roulis et de lacet reviennent en 6 s. L'angle de roulis revient à sa valeur d'équilibre en 20 s. L'angle de lacet converge vers une nouvelle valeur $= 0;8$ en 200 s, 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 x_0y 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, mais aussi sur les vitesses de roulis et de lacet.

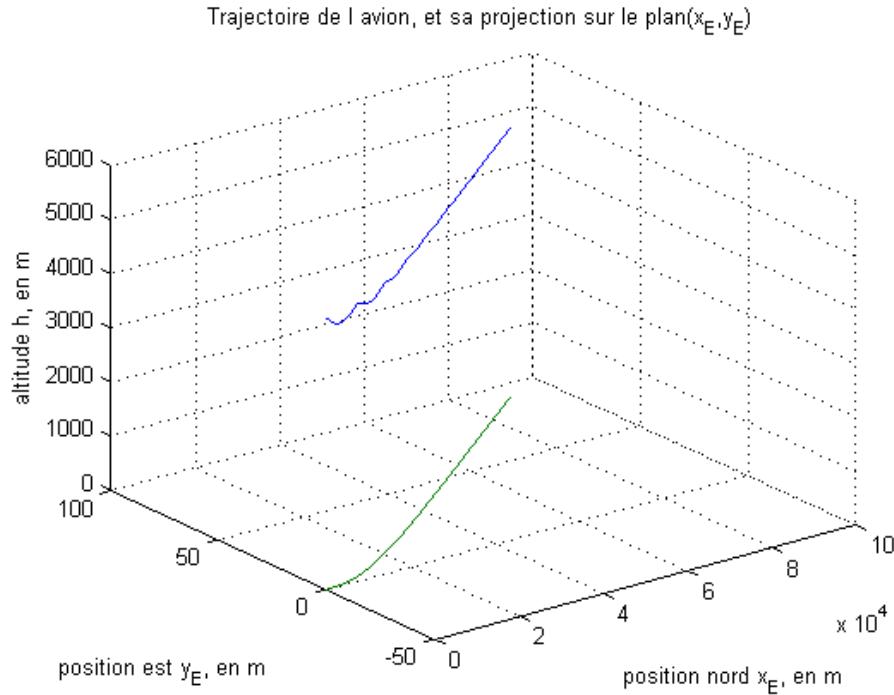


Figure 5.3 Trajectoire de l'avion suite à une perturbation sur e

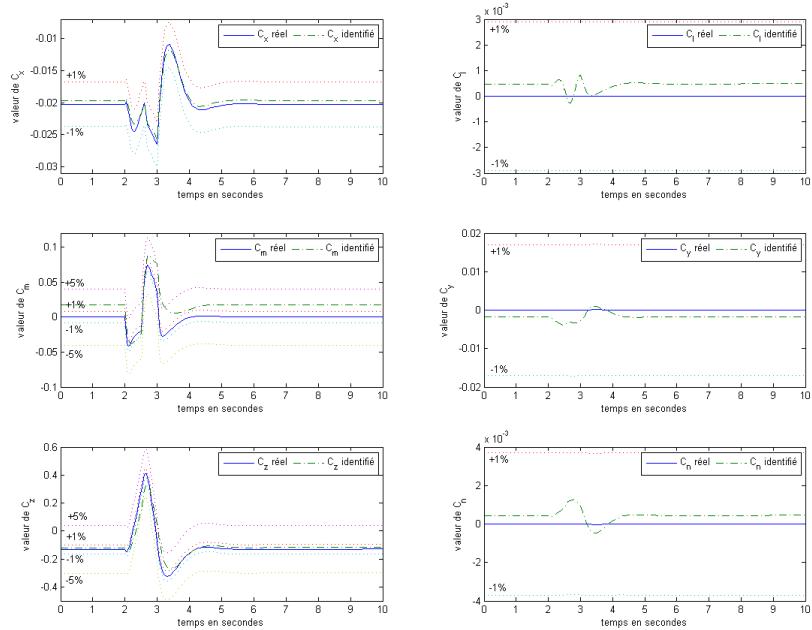


Figure 5.4 Vol a ailes plates - coéficients aérodynamiques

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 Manœuvre : virage coordonné $\alpha = 0,5 \text{ :s}^{-1}$

Pour un virage à $0,5 \text{ :s}^{-1}$ à $V_t = 150 \text{ m:s}^{-1}$ et $h = 5000 \text{ m}$, les variables d'état et de contrôle trimmées sont : $u_0 = [0,34 -1,8 -0,03 0,19]$ et $k_0 = [150 0,015 0,002 0,142 0,015 0 -0,0001 0,001 0,009 0 0 5000 -1,8 1,39 22,30 0,20 -0,02 0,86]$.

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 phugode, sont visibles et identiques à celles observées pour le vol rectiligne à ailes horizontales.

L'angle d'attaque a une valeur nulle non nulle à ne pas créer les forces aérodynamiques nécessaires au maintien en équilibre de l'aéronef. L'angle de dérapage 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 $t = 2 \text{ s}$, et un autre de mêmes caractéristiques à $t = 2,5 \text{ s}$ 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 présente 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.

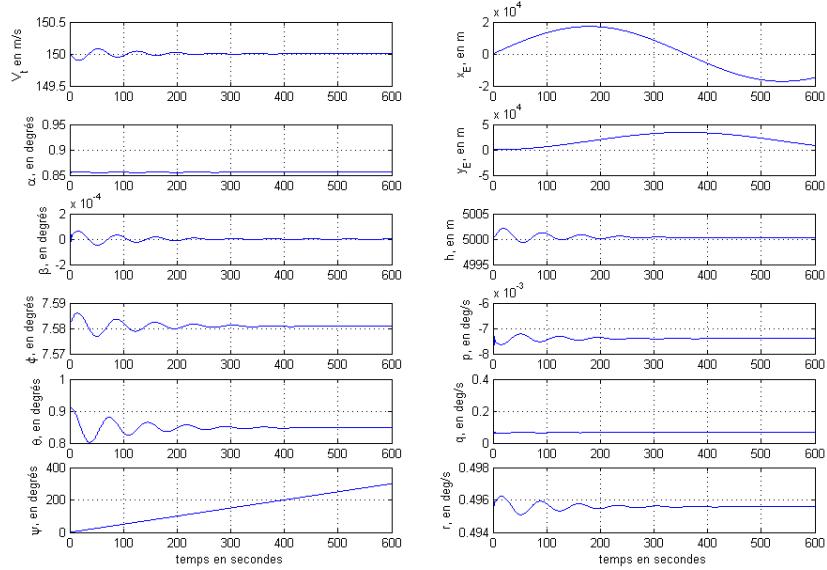
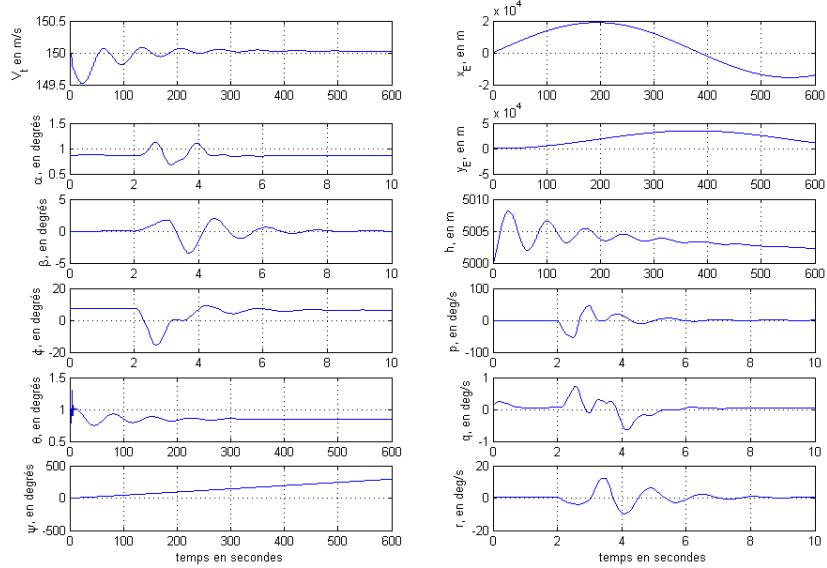


Figure 5.5 Virage - evolution des etats

Figure 5.6 Evolution des etats suite a une perturbation sur a et r

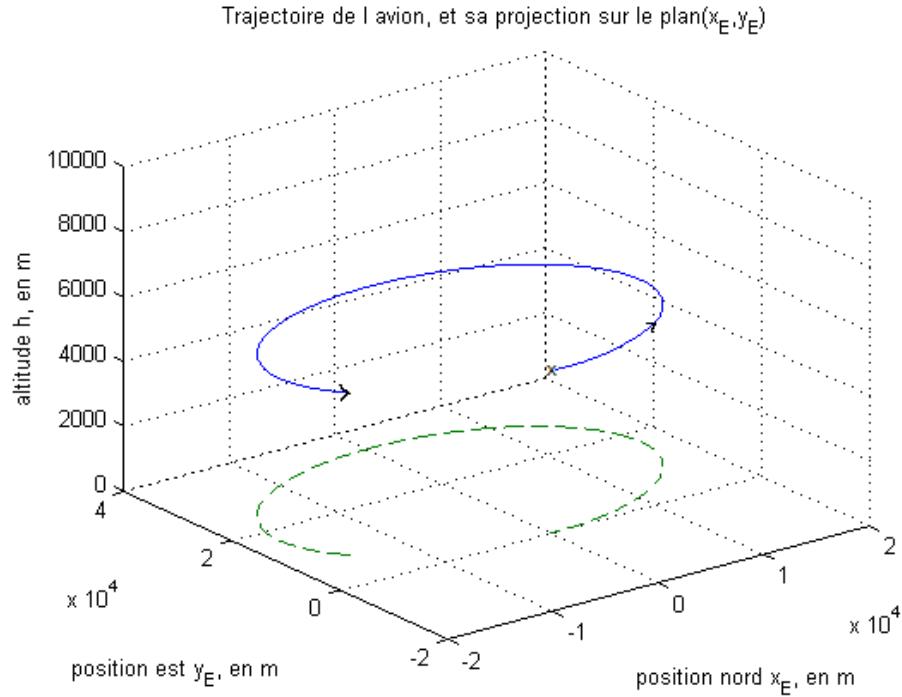


Figure 5.7 Trajectoire de l'avion suite à une perturbation sur a_x et r

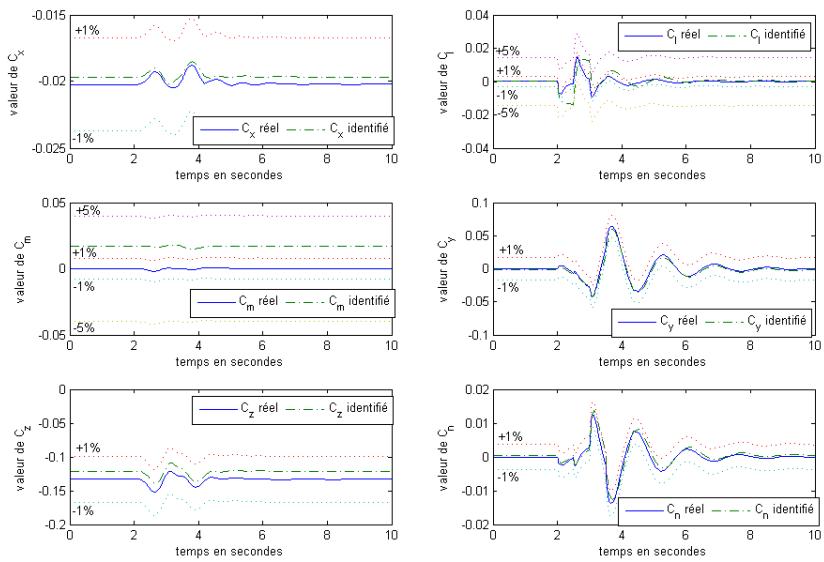


Figure 5.8 Virage - coéfficients aérodynamiques

5.2 Application de la structure du contrôleur au F-16

La structure du contrôleur développée 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.

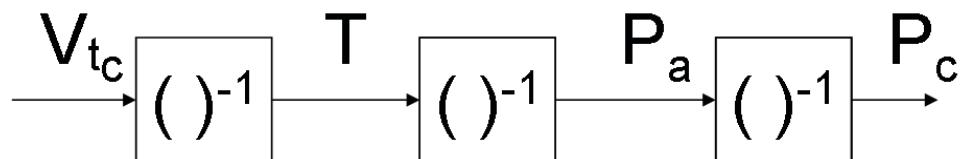


Figure 5.9 Décomposition de l'autothrottle

La première étape est l'inversion de la commande en vitesse pour obtenir la pression telle que décrite en 4.2.6. L'amortissement ζ_v et la pulsation propre ω_v doivent être choisis conformément aux qualités de manuvrabilité.

Pour la dynamique de la vitesse \dot{V}_t , $(\zeta_v; \omega_v)$ sont caractéristiques du mode phugophile. On désire fixer l'amortissement de ce mode $\zeta_{ph} = 0.7$, 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 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

Pousse identifiée	T_{idle}	T_{mil}	T_{max}
nb de neurones	9	2	2
erreur d'entraînement	$9; 65:10^{-4}$	$4; 10:10^{-4}$	$3; 18:10^{-4}$

est cependant vérifié par la simulation, car le temps de réponse du moteur n'est pas négligeable en général. Si nécessaire, on pourra considérer plusieurs 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.

$$P_a = \begin{cases} \frac{8}{\approx 50} \frac{T}{T_{idle}} \frac{T_{idle}}{T_{idle}} & \text{si } T < T_{mil} \\ \frac{8}{\geq 50} \frac{T}{T_{max}} \frac{T_{idle}}{T_{idle}} + 50 & \text{sinon} \end{cases} \quad (5.1)$$

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 due 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 : dépassement et le temps de réponse ayant été fixés par le choix de l'amortissement ν et de la pulsation propre ν .

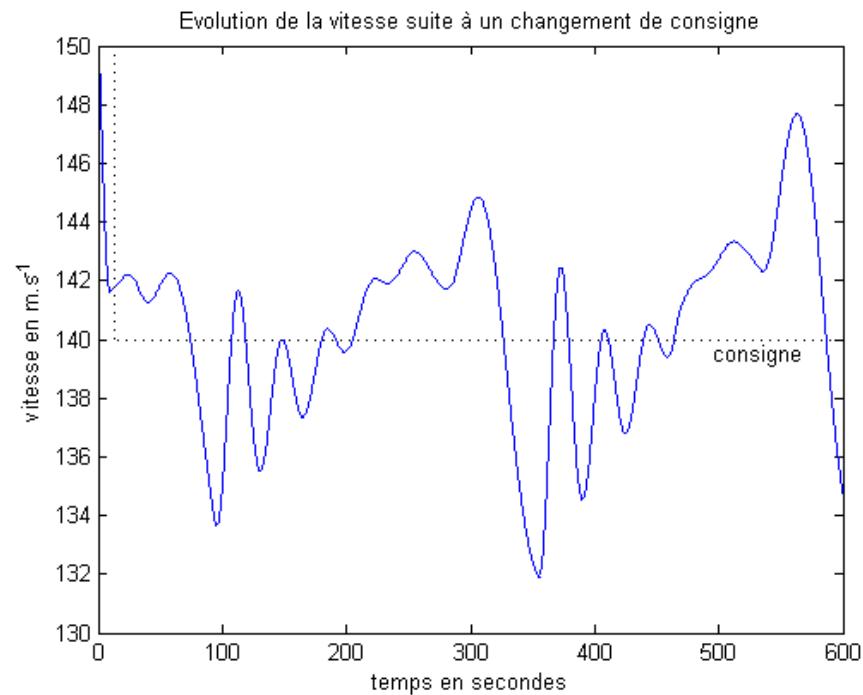


Figure 5.10 Commande e^{rP_c} - Réponse à un changement de consigne

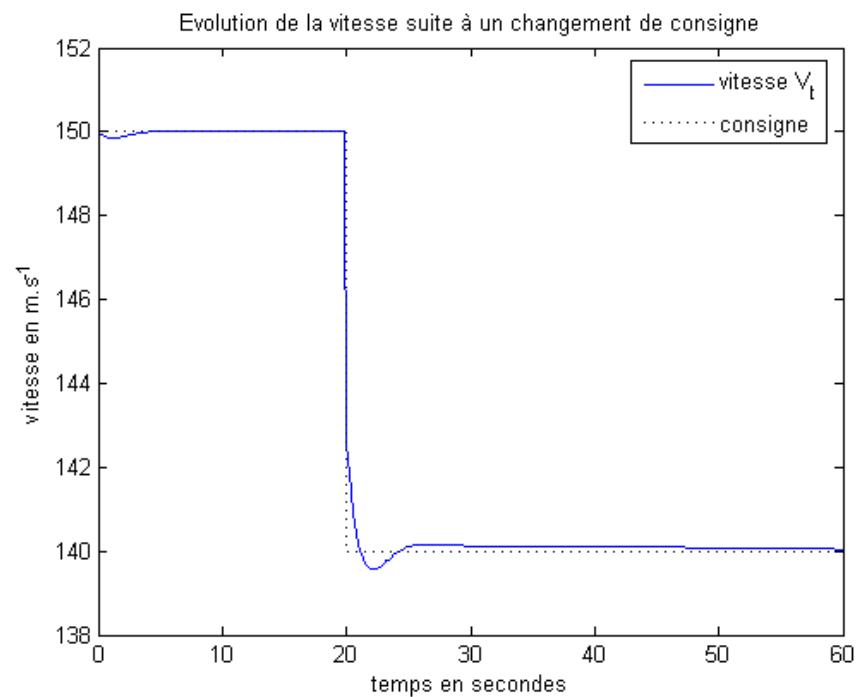


Figure 5.11 Commande e^{rT} - Réponse à un changement de consigne

5.2.2 L'allocation du contrôle

Differentes 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 actions des surfaces de contrôle, et de choisir une matrice de pondération W .

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

$$B(t) = \begin{pmatrix} 2 & & 3 \\ 6 \frac{\partial C_l}{\partial e} & \frac{\partial C_l}{\partial a} & \frac{\partial C_l}{\partial r} \\ 8 \frac{\partial C_m}{\partial e} & 0 & 0 \\ 4 \frac{\partial C_n}{\partial e} & \frac{\partial C_n}{\partial a} & \frac{\partial C_n}{\partial r} \end{pmatrix} \quad (5.2)$$

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

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

$$c(t) = c_{mes}(t) + W^{-1} B(t)^T W^{-1} B(t)^{-1} C_c(t) - C_{mes}(t^{-1}) \quad (5.3)$$

$\alpha_{mes}(t)$ est la position mesurée
et $C_c(t)$ la commande en coefficients aérodynamiques.

On remarque que, comme $B(t)$ est carree, lorsqu'elle est inversible, la matrice de pondération W 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 :

$$W = \begin{matrix} & 2 & 3 \\ & \frac{1}{25^2} & 0 & 0 \\ & 0 & \frac{1}{20^2} & 0 \\ & 0 & 0 & \frac{1}{30^2} \\ \end{matrix} \quad (5.4)$$

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'égalité : 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 met l'hypothèse que les manœuvres étudiées n'entrent pas en saturation, les contraintes ne sont donc pas explicitement prises en compte.

Le F-16 possède, outre les trois surfaces de contrôle usuelles, une surface supplémentaire : 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 :

$$B(t) = \begin{matrix} & 2 & & 3 \\ & \frac{\partial C}{\partial e} & \frac{\partial C}{\partial a} & \frac{\partial C}{\partial r} & \frac{\partial C}{\partial lef} \\ \begin{matrix} 6 \\ 6 \\ 4 \end{matrix} & 0 & 0 & \frac{\partial C}{\partial lef} \\ & \frac{\partial C}{\partial e} & \frac{\partial C}{\partial a} & \frac{\partial C}{\partial r} & \frac{\partial C}{\partial lef} \end{matrix} \quad (5.5)$$

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 .

$$W_1 = \begin{matrix} & 2 & & 3 \\ & \frac{1}{25^2} & 0 & 0 \\ 6 & 0 & \frac{1}{20^2} & 0 \\ 6 & 0 & 0 & \frac{1}{30^2} \\ 4 & 0 & 0 & 0 \end{matrix}, \quad W_2 = \begin{matrix} & 2 & & 3 \\ & 1 & 0 & 0 \\ 6 & 0 & 1 & 0 \\ 6 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 \end{matrix}$$

$$W_3 = \begin{matrix} & 2 & & 3 \\ & 0:1 & 0 & 0 \\ 6 & 0 & 0:1 & 0 \\ 6 & 0 & 0 & 0:1 \\ 4 & 0 & 0 & 1 \end{matrix}, \quad W_4 = \begin{matrix} & 2 & & 3 \\ & 0:001 & 0 & 0 \\ 6 & 0 & 0:001 & 0 \\ 6 & 0 & 0 & 0:001 \\ 4 & 0 & 0 & 100 \end{matrix}$$

Pour l'allocation sur l'ensemble des quatre surfaces de contrôle, le choix $= W_4$ 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.

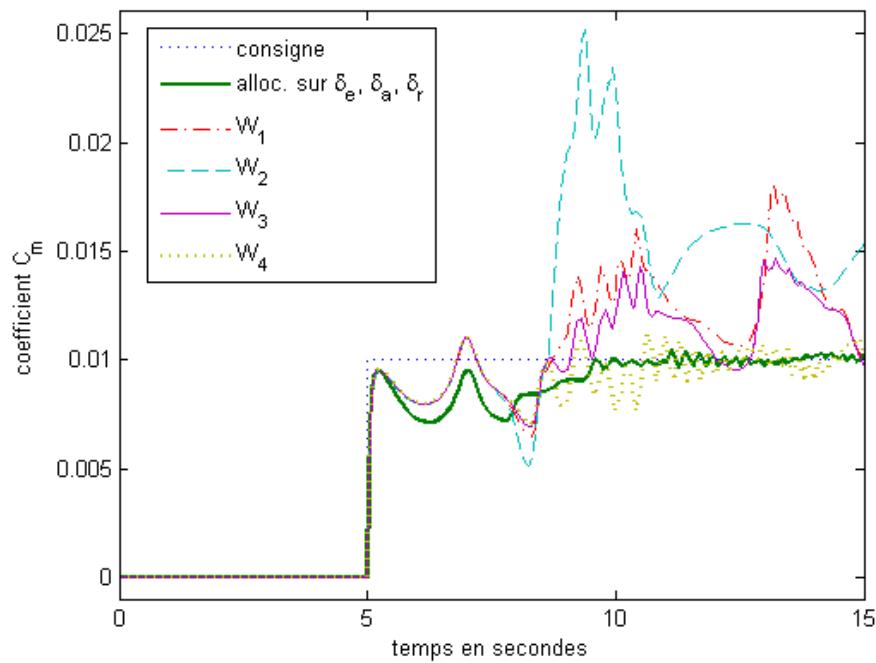
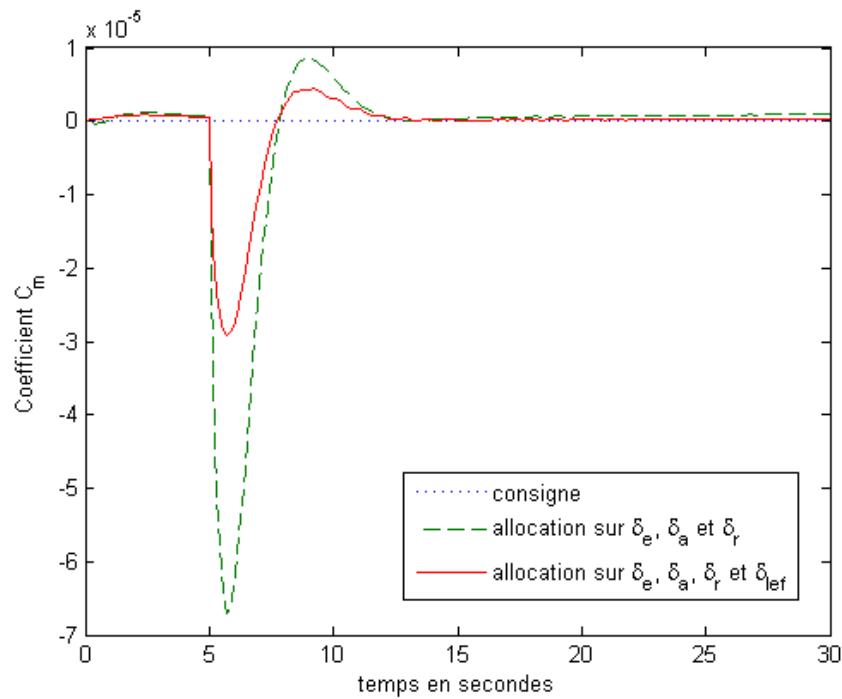
Figure 5.12 Influence de W sur l'allocation

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 $t = 5\text{ s}$ la faisant passer de 150 m:s^{-1} à 140 m: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 autres de manœuvrabilité doivent donc être de niveau I.

Pour la dynamique du taux de tangage, $(!_q; q)$ 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

$$!_q = \zeta_{mr} = 1 \text{ et } !_q = \max \left(\frac{r}{CAP_{des}}, \frac{n_z}{1} \right)$$

Pour la dynamique du taux de roulis, $_p$ est caractéristique du mode roulis amorti.

Pour les phases de vol de catégorie B, il faut $r_{ra} = \frac{1}{1.4}$; pour celles de catégorie A, $r_{ra} = 1$. Alors $_p$ est choisie égale à r_{ra} .

Pour la dynamique du taux de lacet, λ_r est caractéristique du roulis hollandais. On désire xer la pulsation propre du roulis hollandais $\omega_{rh} = 3$, l'amortissement etant $\zeta_a = \omega_{rh} = 1$ avec le type de dynamique désirée choisi. D'où $\lambda_r = \omega_{rh}$.

Pour la dynamique de l'angle d'attaque, $(\dot{\alpha}; \dot{\beta})$ sont caractéristiques du mode rapide longitudinal. Les caractéristiques ont été xées pour la dynamique du taux de tangage, cependant l'hypothèse de séparation temporelle de la dynamique doit être respectée. On choisit alors

$$\lambda_m = 0; 7 \text{ et } \lambda_r = 2$$

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

$$\lambda_d = \omega_{rh} \text{ et } \lambda_r = \omega_{rh}$$

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

Tout en étant conformes aux qualités de manuvrabilité, les valeurs des amortissements et des pulsations propres ont été choisies arbitrairement ici. Ces dernières 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 Sauret 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 augmenté de retour proportionnel + intégral qui a été développé dans cette étude.

La trajectoire à suivre est la suivante :

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

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 $\dot{\gamma}$, 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 $\dot{\gamma}$ et l'angle de roulis ϕ , soit par l'étape précédente du contrôle pour les vitesses en roulis, en tangage et en lacet, les coefficients aérodynamiques de moment C_l , C_m et C_n et les actions des surfaces de contrôle e , a , r et lef .

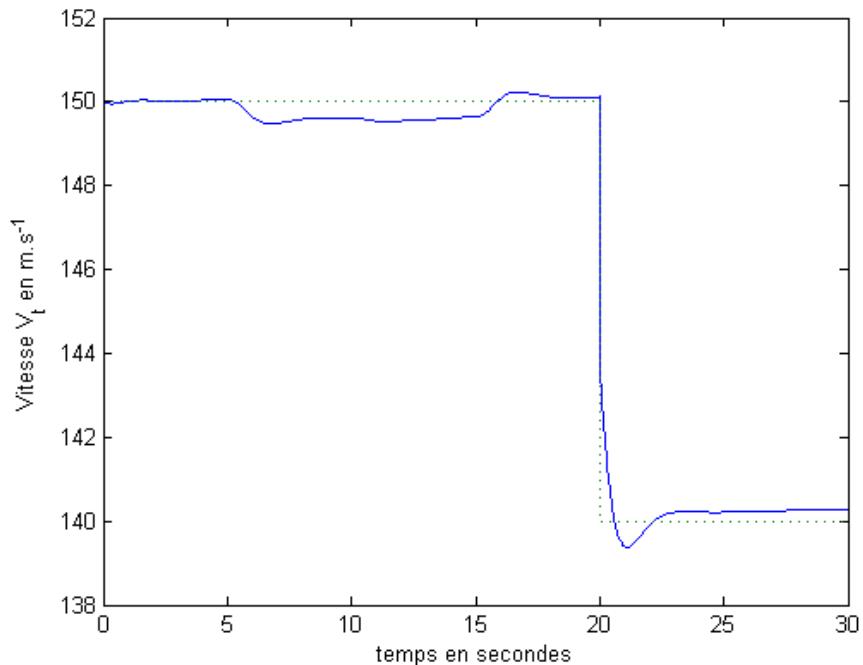


Figure 5.14 Suivi d'une trajectoire - réponse en vitesse

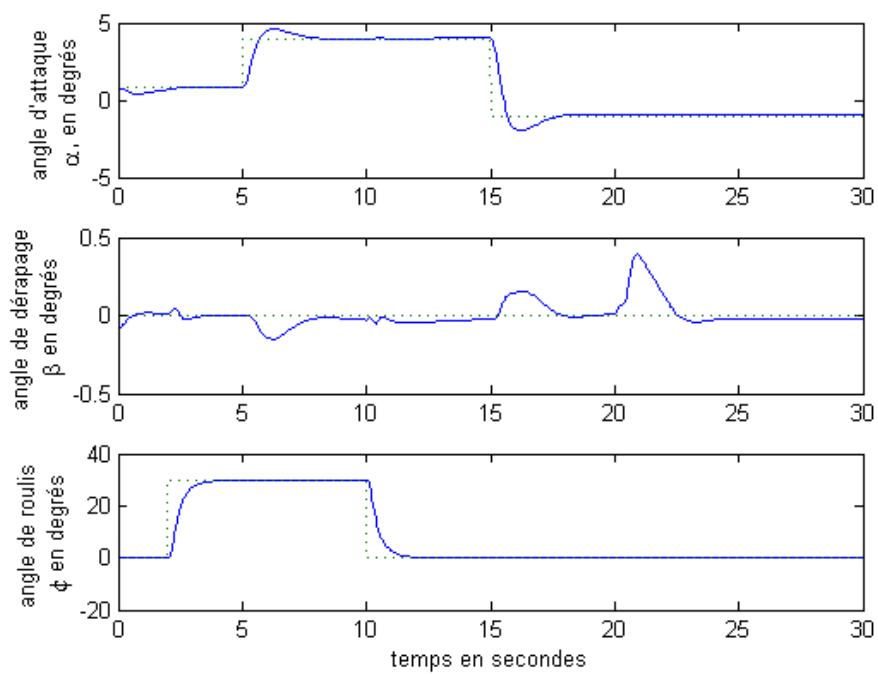


Figure 5.15 Suivi d'une trajectoire - réponse en angles, et

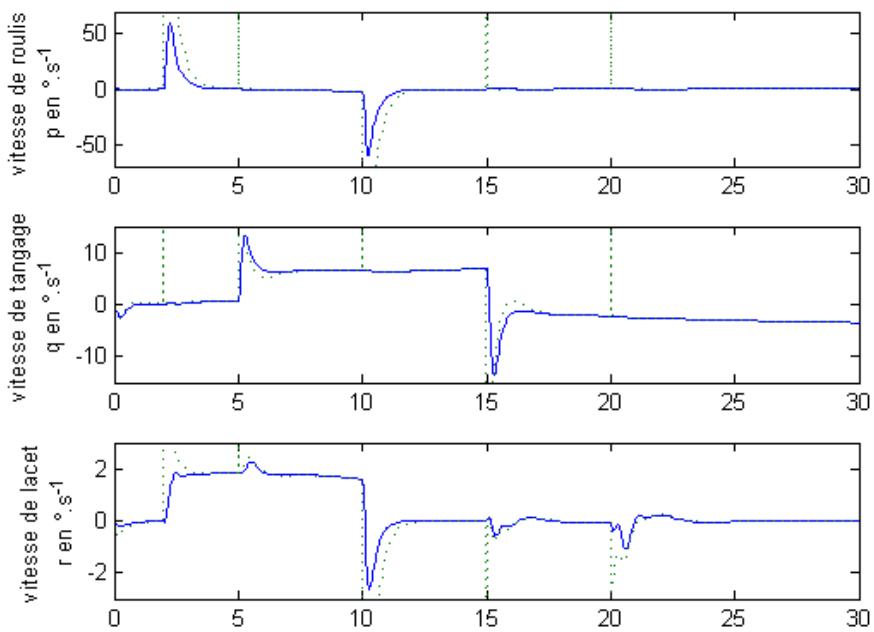


Figure 5.16 Suivi d'une trajectoire - réponse en vitesse de rotation p , q et r

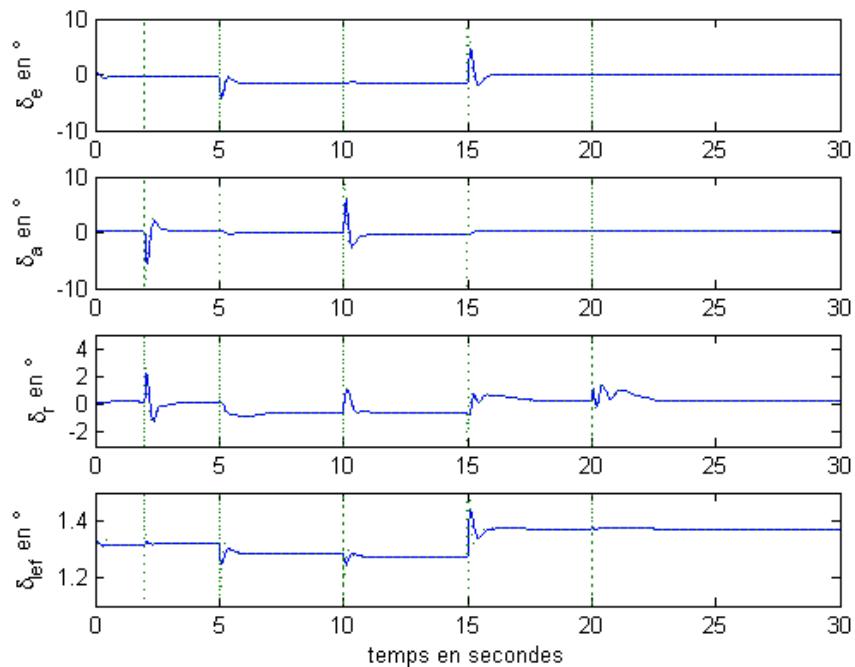


Figure 5.17 Suivi d'une trajectoire - défections des surfaces e , a , r et lef

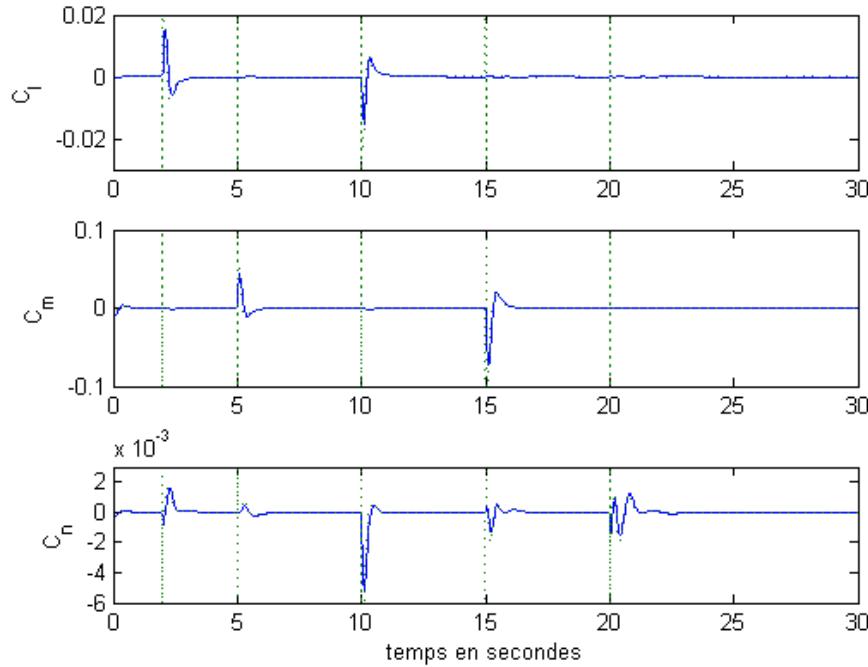


Figure 5.18 Suivi d'une trajectoire - coe cients aérodynamiques C_l , C_m et C_n

La vitesse V_t présente de beres dierences avec sa consigne. Cela est dû au fait que l'autothrottle n'est pas totalement de-couplé du reste du contrôleur. Mais cette perturbation reste très faible : inférieure a 0; 005%. L'angle de derapage reste dans un voisinage de zero (< 0; 5), sa performance est donc jugée satisfaisante. Ainsi le contrôleur assure un bon suivi de trajectoire lorsque le modle est ~~parfait~~.

5.3.2 Cas d'un modle "neural"

Observons desormais le comportement du système dans le cas où le modle ~~parfait~~ dynamique n'est pas parfait. L'identi cation des coe cients aérodynamiques est faite par le module a reseaux de neurones, tandis que le modle "neural" des coe cients utilis dans le simulateur du F-16 est obtenu par interpolation cubique des donrées 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'élevateur 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 prenant un modèle non parfait.

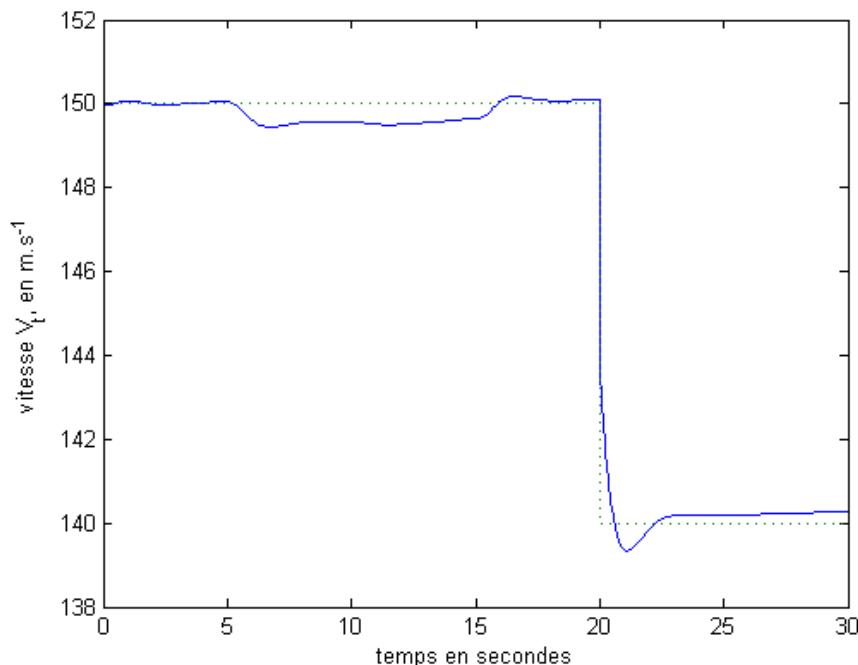


Figure 5.19 Suivi d'une trajectoire - réponse en vitesse

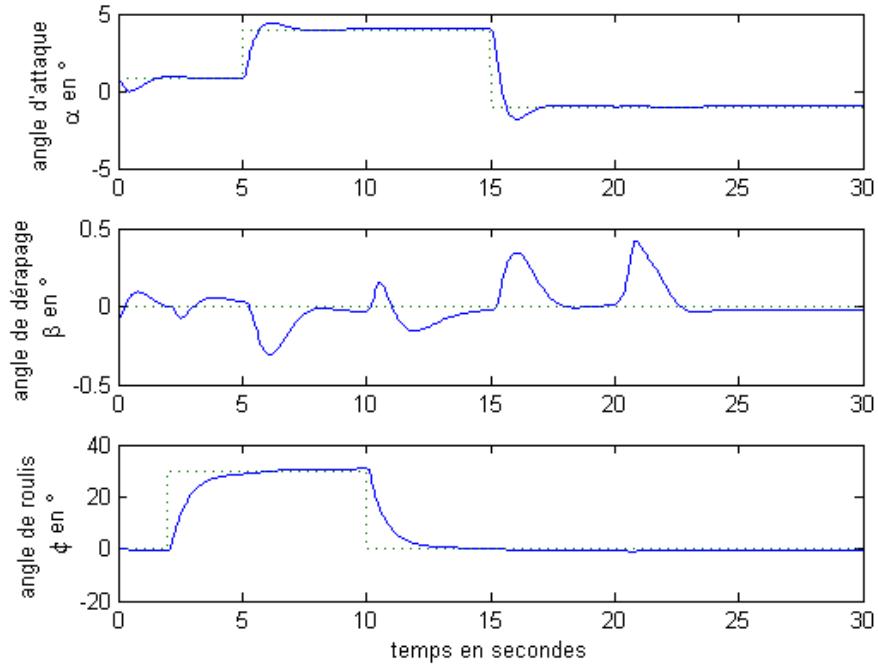


Figure 5.20 Suivi d'une trajectoire - réponse en angles, et

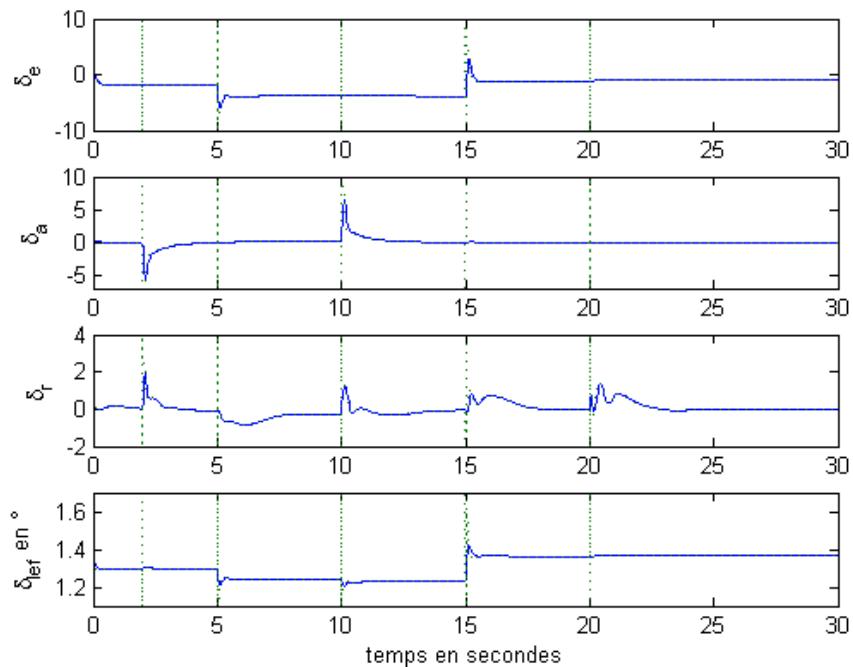


Figure 5.21 Suivi d'une trajectoire - déflection des surfaces e , a , r et lef

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 mise en évidence. La commande prédictive est ensuite appliquée au F-16, dont la boucle interne linearise le modèle par retour de sortie.

5.4.1 Influence des différents paramètres

Diverses 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 associé avec l'horizon de contrôle. Plus R est grand, plus l'horizon 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 prédition.

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 prédition pour un temps d'échantillonage Δt . Plus l'horizon est grand, plus le changement de consigne est anticipé. Cette anticipation permet d'ailleurs de réduire l'horizon 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

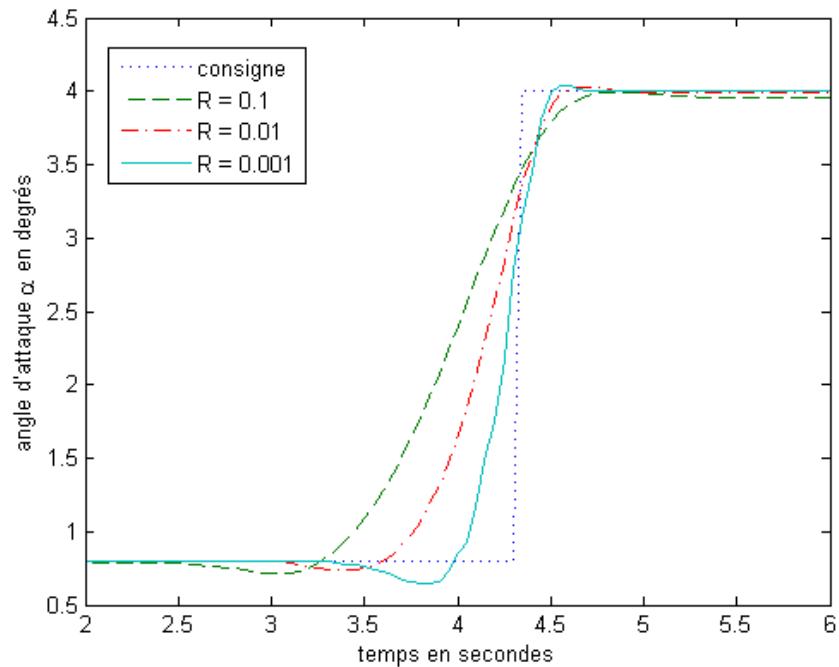


Figure 5.22 Influence de la matrice de pondération R

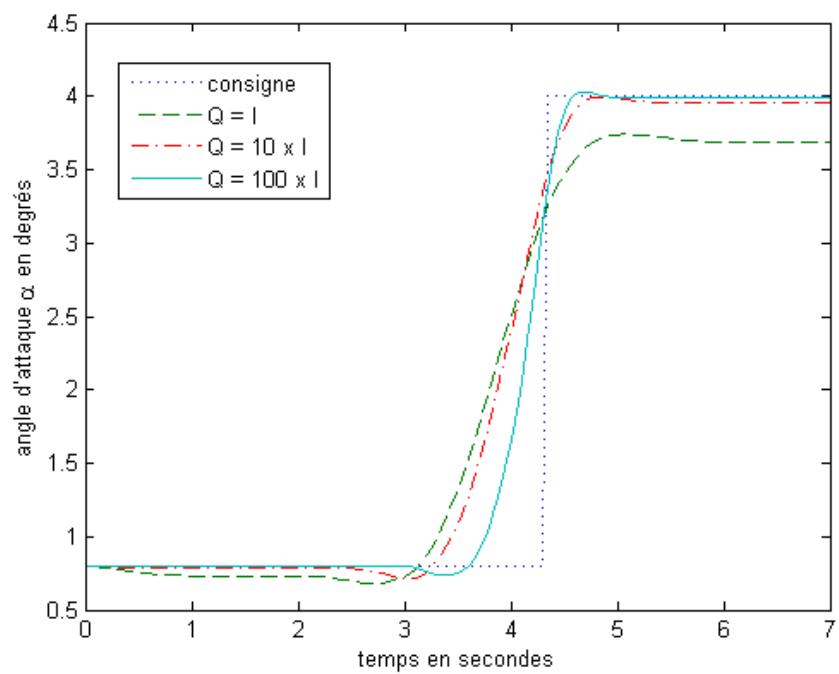


Figure 5.23 Influence de la matrice de pondération Q

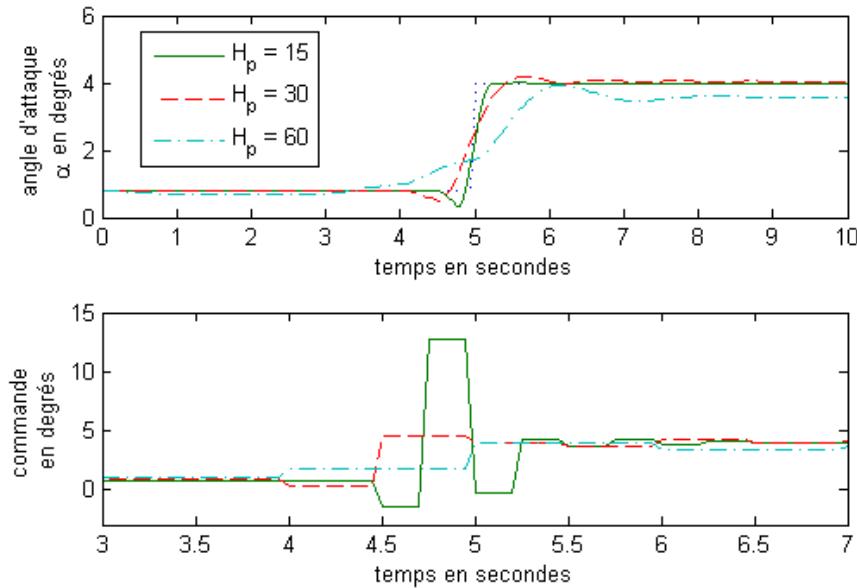


Figure 5.24 Influence de l'horizon de prédiction

Bbkbar, dont la dimension est modifiée. La nouvelle matrice est obtenue quasiment en sommant les colonnes originelles et en augmentant les puissances 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 H_p sur laquelle la consigne est maintenue constante. On remarque sur la figure 5.25 que plus H_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 $t = 4\text{s}$. La commande prédictive l'enraye rapidement comme le montre la figure 5.26.

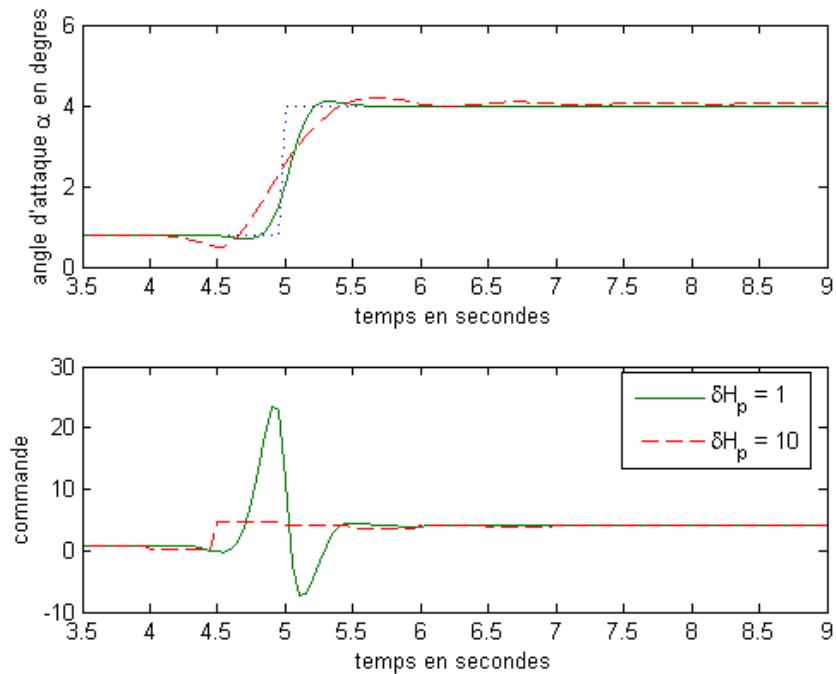


Figure 5.25 Influence du temps de maintien de la commande

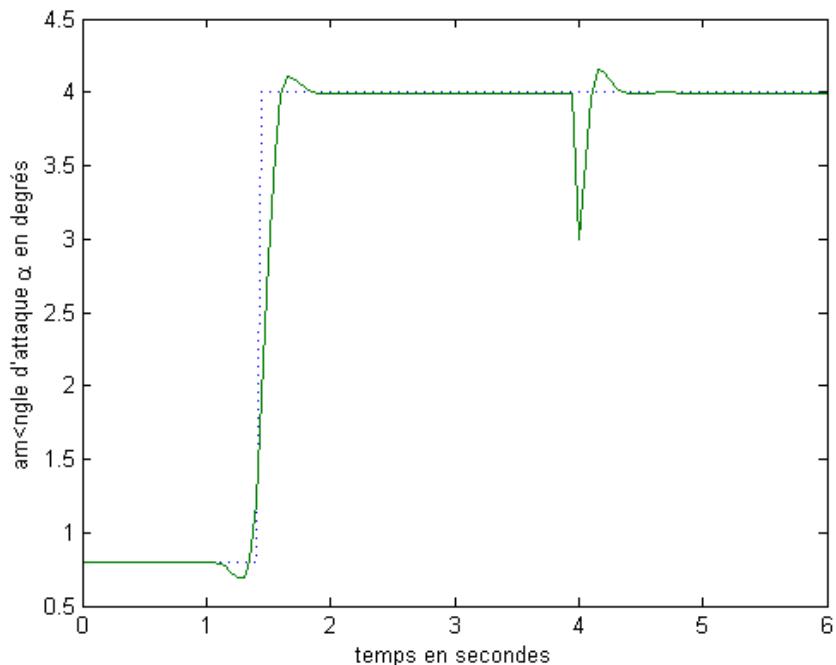


Figure 5.26 Rejet de perturbation constante

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 γ .

L'étude de l'influence des paramètres et leur application au modèle obtenu si l'inversion était parfaite mène aux choix suivants :

$$T_{ech} = 0,05s$$

$$H_p = 1,5s$$

$$H_p = 0,5s$$

$$Q = 10 \quad I_3$$

$$R = 0;01 \quad I_3$$

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 linearisation par retour de sortie n'est pas exactement vérifiée. En effet en l'absence du boulage de la dynamique désirée de type proportionnel + intégral, la dynamique de l'angle d'attaque linearisée par le retour de sortie devrait être la suivante :

$$\dot{\alpha} = \alpha_{des} \tag{5.6}$$

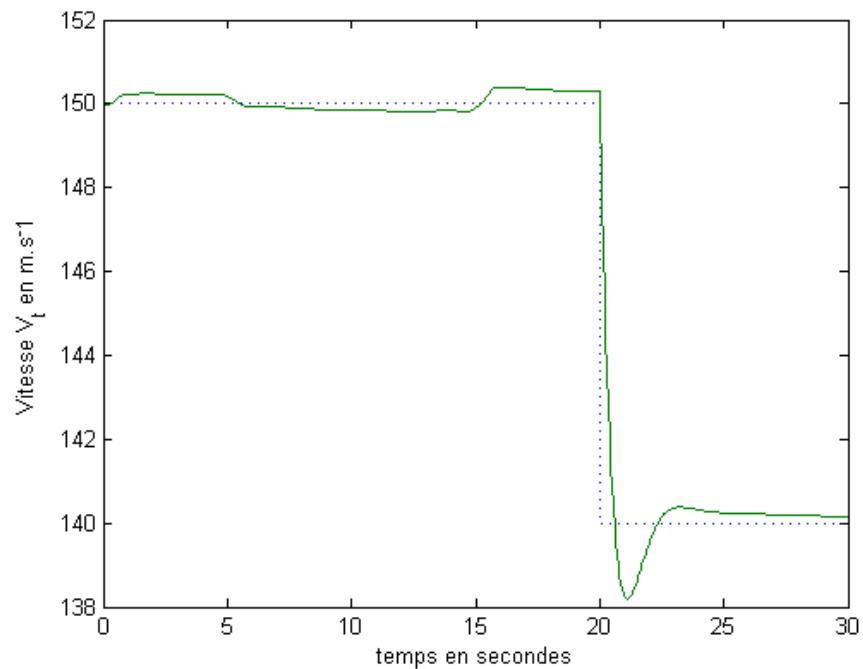


Figure 5.27 Suivi de trajectoire - réponse en vitesse

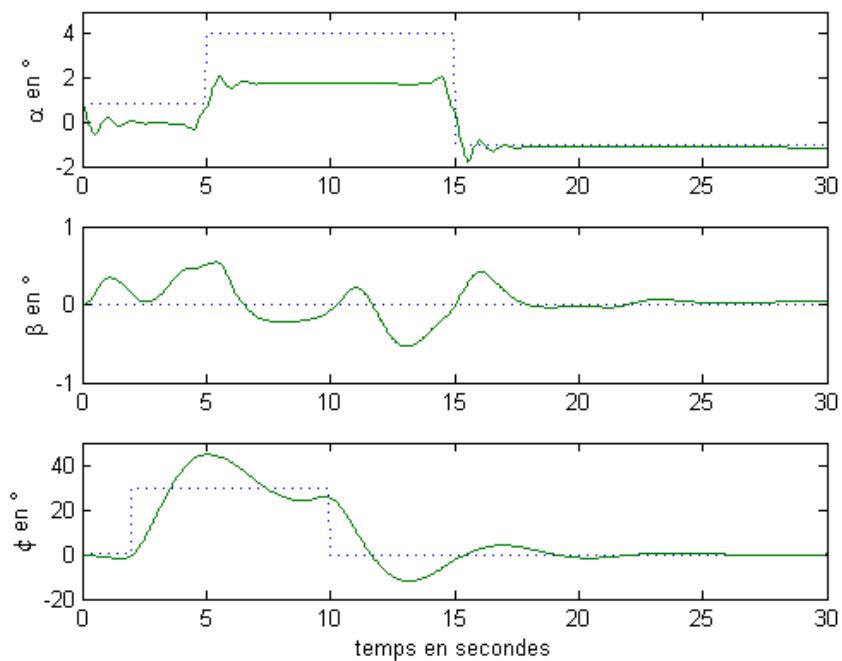


Figure 5.28 Suivi de trajectoire - réponse en , et

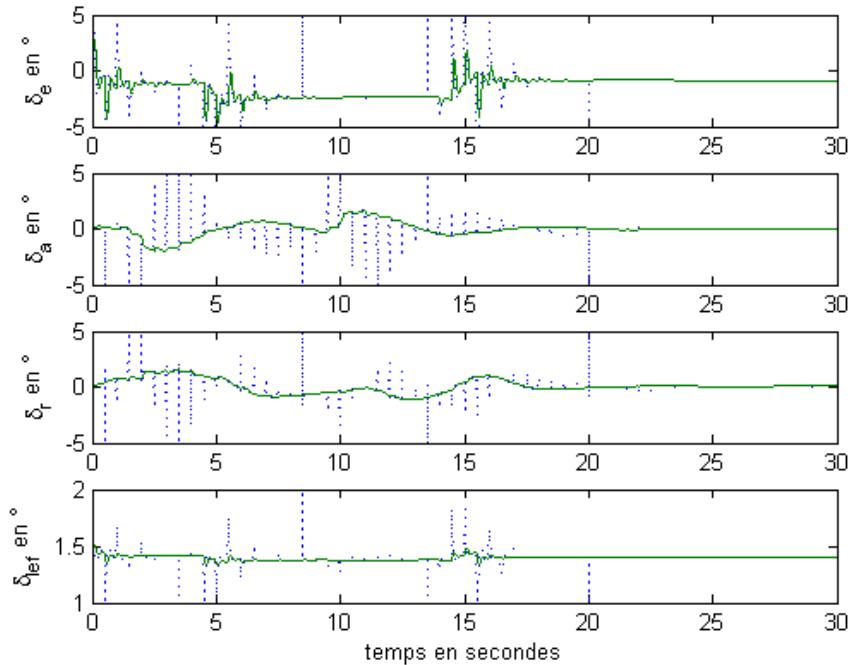


Figure 5.29 Suivi de trajectoire - réponse en_e, a, r et _{lateral}

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 comme le fait que la linéarisation n'est pas parfaite

$$\text{Figure 5.30: } \delta v \quad (5.7)$$

On conclut de ces simulations que la commande prédictive telle qu'appliquée ici n'est pas seulement 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 cher aisement le

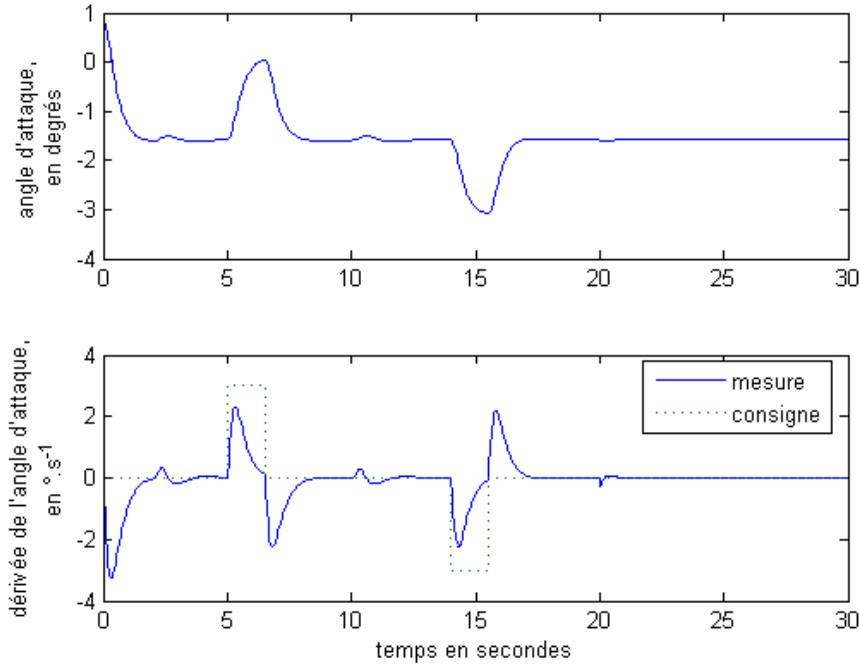


Figure 5.30 Performance du système linearisé par retour de sortie sans PI

comportement en boucle fermée du système, afin de garantir les différentes qualités de maniabilité.

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 reprend 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 linearisation par tour 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 ;(;).

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 signifiants de l'avion sans avoir recours au sequencing 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 (\dot{x}, \dot{y}) , mais une commande sur la trajectoire à suivre, par exemple V_t , h et γ . 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 γ_{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 et l'assiette de roulis c_z .

Sous l'hypothèse que le retour de sortie effectue une linearisation exacte, la dynamique en $(\dot{x}, \dot{y}, \dot{z})$ de la boucle fermée par inversion dynamique, augmenté par une dynamique désirée de type proportionnel + intégral, a la forme suivante:

L'altitude h , l'assiette de tangage et l'azimut sont legis par les equations

suivantes :

$$\begin{aligned}
 h &= V_t \cos \cos \sin \quad V_t \sin \cos \sin \quad V_t \sin \cos \cos \cos \\
 &= q \cos \quad r \sin \\
 &= \frac{q \sin + r \cos}{\cos}
 \end{aligned} \tag{6.2}$$

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, et , 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 , , et .

La combinaison de ces équations permet d'exprimer le système sous la forme :

$$\begin{array}{ccccc}
 2 & 3 & 0 & 2 & 3 & 1 \\
 \bullet & & & & & \\
 \cdot & & & & & \\
 \cdot & & & & & \\
 = f & & ; & & ; & C_x; C_y; C_z \\
 h & & h & & c & \\
 4 & 5 & @4 & 5 & & A \\
 - & & & & &
 \end{array} \tag{6.3}$$

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 fixe, dans le cas longitudinal soit $\dot{h} = 0$. 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 :

$$\cos \alpha = 1 \text{ et } \sin \alpha = 0$$

De même on suppose que l'angle de tangage est petit. Dans ce cas, les équations en altitude h et en angle de tangage du système deviennent :

$$\begin{aligned} h &= V_t t + h_0 \\ \dot{\alpha} &= q_c \end{aligned} \quad (6.4)$$

Les équations d'inversion donnent :

$$\begin{aligned} q_c &= \frac{1}{\frac{1}{1 + \frac{qS}{2mV_t}} C_{zq} \cos \alpha + C_{xq} \sin \alpha} - \frac{g}{V_t} (\cos \alpha \dot{\alpha} + \sin \alpha \dot{\beta}) \\ &\quad - \frac{qS}{mV_t} C_{z0} \cos \alpha \dot{\alpha} + \sin \alpha (C_{x0} + \frac{m}{qS \cos \alpha} V_t) C_x C_z \tan \beta \\ &\quad - \frac{m}{qS \cos \alpha} g (\cos \alpha \dot{\beta} + \sin \alpha \dot{\alpha}) \end{aligned} \quad (6.5)$$

Choisissons la condition de référence comme étant $V_t = 150 \text{ m:s}^{-1}$ et $h = 5000 \text{ m}$.

On y linearise q_c de sorte à obtenir :

$$q_c = -\frac{g}{V_t} \tan \beta + Z \quad (6.6)$$

Dans le cas d'une application au F-16 on a $a_{loZ} = 0;017$ et $Z = 0;065$.

La commande adaptative est alors appliquée au système suivant :

$$\begin{matrix} 2 & 3 & 2 \\ 6 & -7 & 6 & 0 & 1 & 0 & 0 & 3 & 2 & 3 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} = \begin{matrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} + \begin{matrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \quad (6.7)$$

$$\begin{matrix} 2 & 3 & 2 \\ 6 & -7 & 6 & 0 & 1 & 0 & 0 & 3 & 2 & 3 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} = \begin{matrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} + \begin{matrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{matrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

De même que précédemment les différents paramètres sont yes. 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 :

$$T_{ech} = 0;05s$$

$$H_p = 3s$$

$$H_p = 1s$$

$$Q = \begin{matrix} 2 & 3 \\ 4^{\frac{1}{7500^2}} & 0 \\ 0 & \frac{1}{20^2} \end{matrix}$$

$$R = \frac{1}{10^2}$$

Ils permettent un bon compromis entre le temps de réponse, une erreur statique faible et une commande limitée en angle d'attaque.

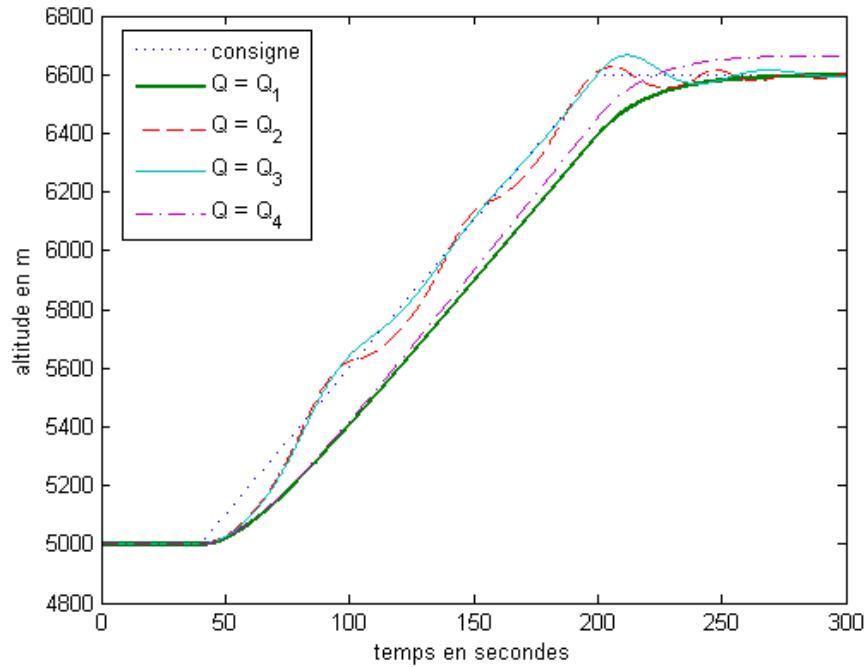


Figure 6.1 Commande en altitude - influence de Q

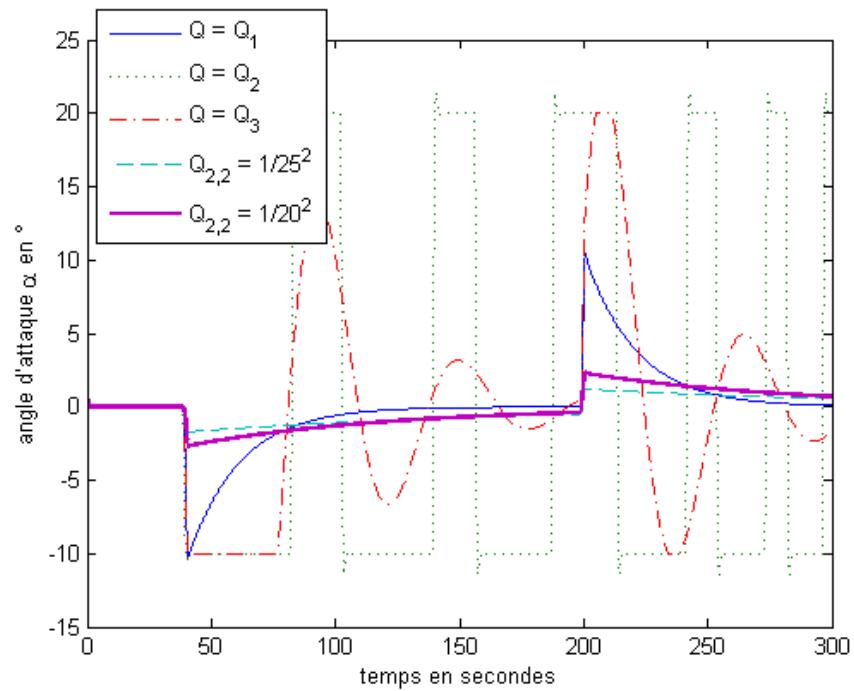


Figure 6.2 Commande en altitude - angle d'attaque gérée

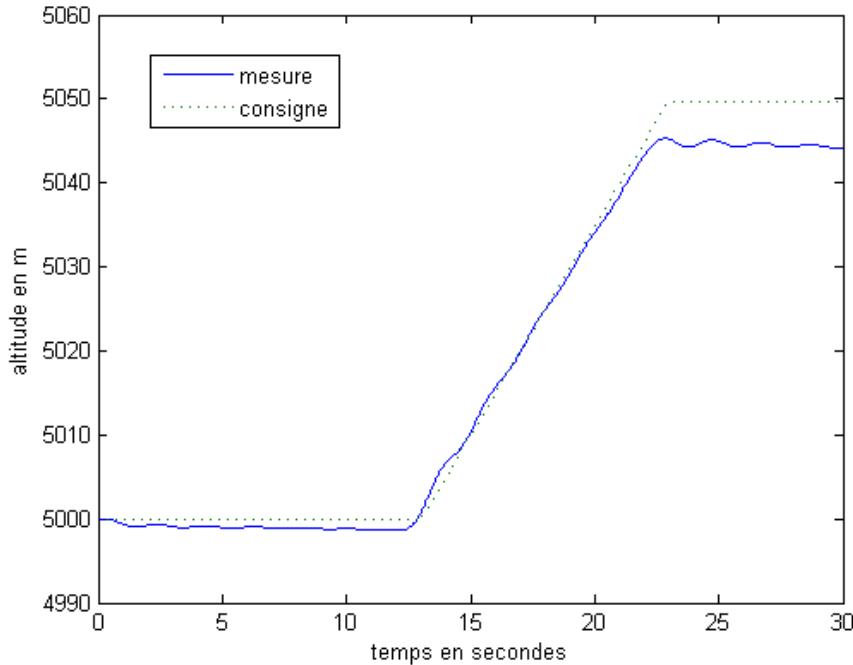


Figure 6.3 Commande en altitude - réponse en altitude

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 et de roulis sont prises nulles. La consigne en vitesse est prise constante égale à 15 m:s^{-1} .

Les figures 6.3 et 6.4 illustrent les réponses en altitude et en vitesse, suite à une rampe de pente $\dot{h} = 5 \text{ m:s}^{-1}$ appliquée à l'altitude entre les instants $t = 13 \text{ s}$ et $t = 23 \text{ s}$. 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 adapté. 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.

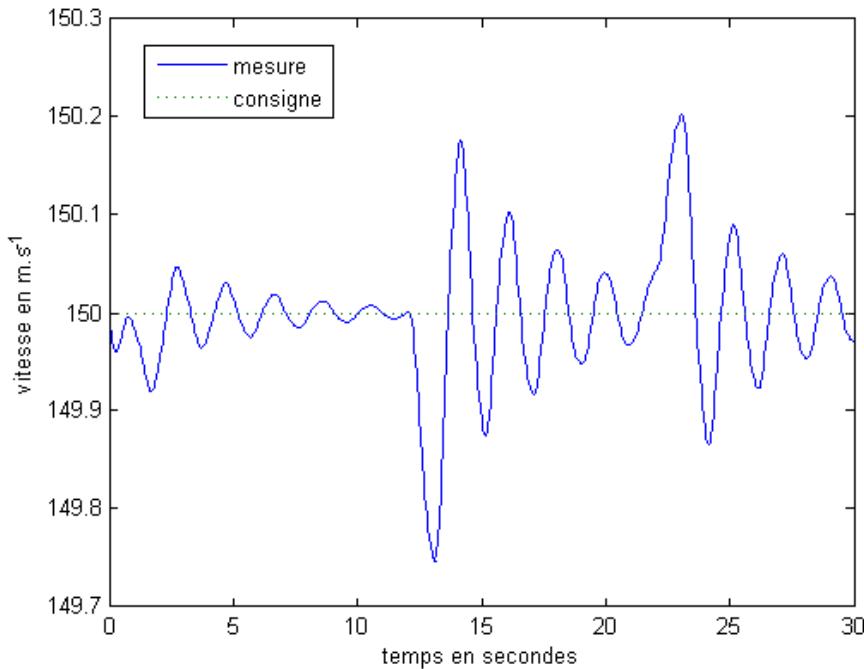


Figure 6.4 Commande en altitude - réponse en vitesse

À l'heure d'aujourd'hui (Keviczky et Balas (2006)), selon laquelle le contrôleur développé à partir d'un unique modèle linéaire est valable sur une large partie de l'enveloppe de vol, le contrôleur précédent est conservé, mais la condition de l'état initial 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 $\dot{h} = 5 \text{ m:s}^{-1}$ mais $\dot{h} = 10 \text{ m:s}^{-1}$. 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 découpées par l'inversion dynamique et la stabilité du système.

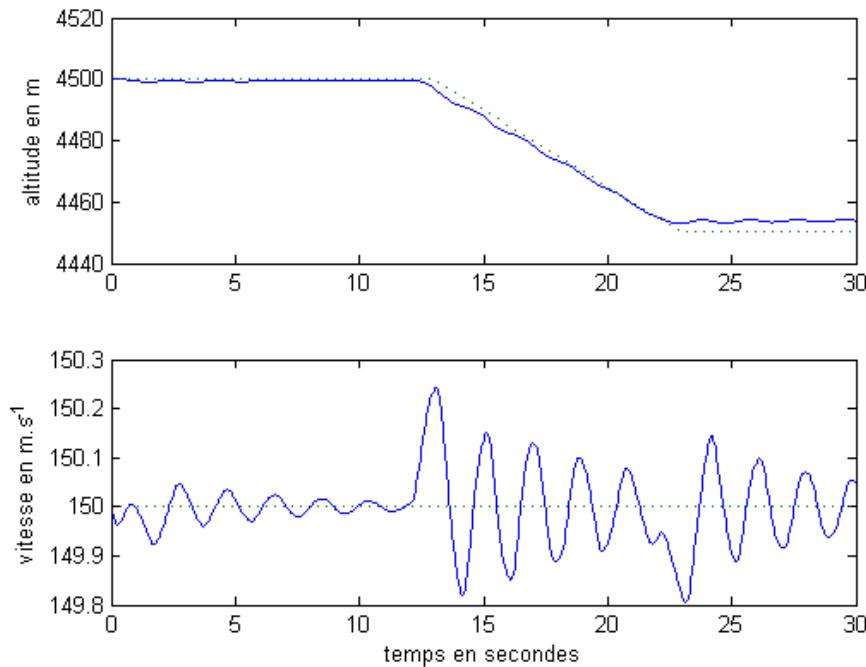


Figure 6.5 Commande en altitude - $C.IV_t = 150\text{m:s}^{-1}$ et $h = 4500\text{m}$

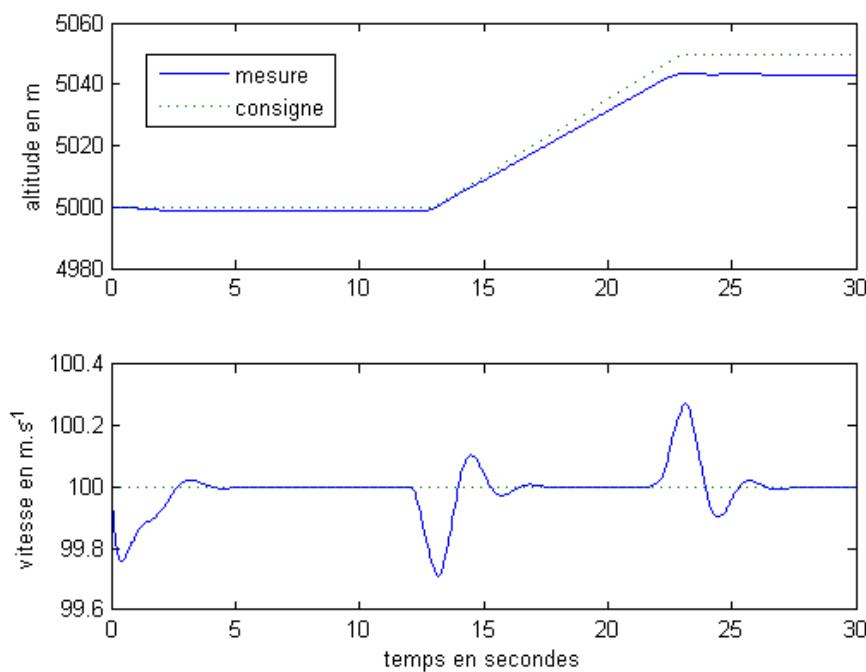


Figure 6.6 Commande en altitude - $C.IV_t = 100\text{m:s}^{-1}$ et $h = 5000\text{m}$

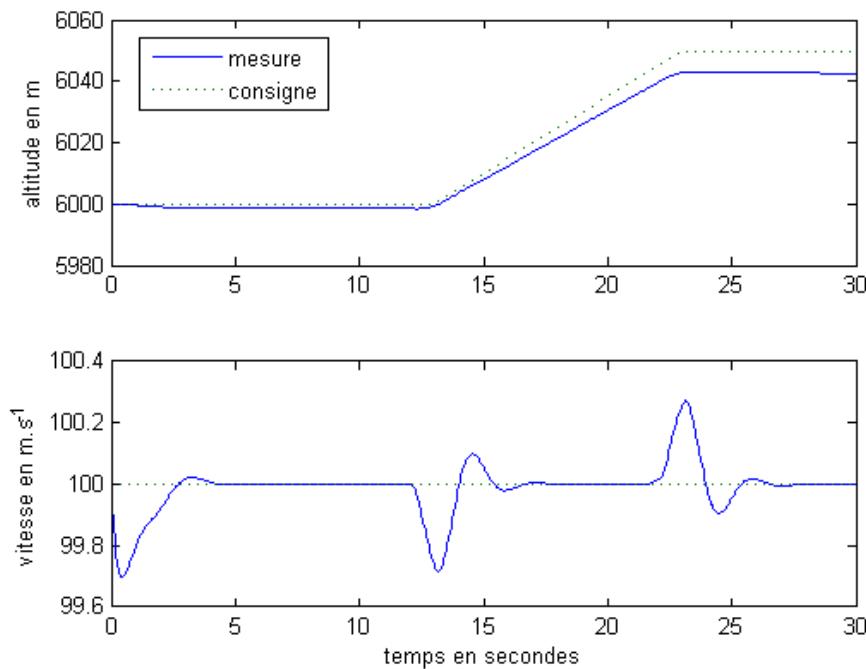


Figure 6.7 Commande en altitude - $C.IV_t = 100\text{m:s}^{-1}$ et $h = 6000\text{m}$

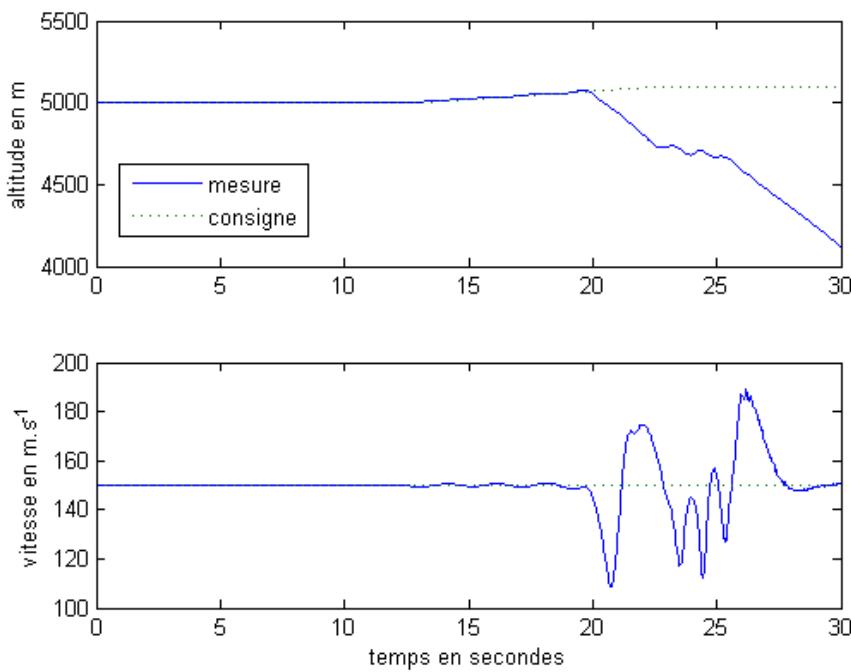


Figure 6.8 Commande en altitude - pente de 10s^{-1}

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 suffisamment rapide pour fournir les données nécessaires aux différentes boucles de contrôle.

L'évolution constante de la rapidité des processeurs et l'utilisation de processus 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 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 (, ,).

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 réversible.

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 et ne nécessite plus d'intervention humaine mais requiert un temps de processeur important. Celui-ci est hors ligne, et l'utilisation de processeurs plus rapides que le Pentium III à 450 MHz utiliser 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 linearisé. La linearisation par retour de sortie prend comme argument la dynamique désirée des variables contrôlées ; l'utilisation d'un type proportionnel + intégral permet de garantir une certaine robustesse au contrôle.

Les gains de cette dynamique désirée sont directement liés aux qualités de maniabilité, 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 linearisé 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 recours au découplage des mouvements longitudinaux et latéraux et obvie au "gain scheduling".

REFERENCES

- 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, 980-984.
- DE SANTIS, R. M. (2006). Commande non linéaire - notes de cours 5ème ed., Montréal, Québec : 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, 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 Conference504-511.
- 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 Dynamics24:5, 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 Conference6, 4278-4282.
- HAYKIN, S. (1999). Neural networks : a comprehensive foundation2nd 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 1982-1986.

- ITO, D. et al (1979). Reentry vehicle flight controls design guidelines : dynamic inversion, Lyndon B. Johnson 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 flight 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 comparative 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, text{p}RAM-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 aircraft dynamics using neural networks", Proceedings on the International Joint Conference on Neural Networks, 3, 2076-2080.
- N RGAARD, 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, these 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, 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, 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 reconstructable 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 Equations 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, l'angle de tangage et l'angle de roulis. 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) :

$$\begin{aligned}
 &= p + \tan(q \sin \theta + r \cos \theta) \\
 &= q \cos \theta - r \sin \theta \\
 &= \frac{q \sin \theta + r \cos \theta}{\cos \theta}
 \end{aligned} \tag{I.1}$$

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'écritent:

$$\begin{aligned}
 x_E &= u \cos \theta \cos \phi + v (\cos \theta \sin \phi \sin \psi - \sin \theta \cos \psi) \\
 &\quad + w (\cos \theta \sin \phi \cos \psi + \sin \theta \sin \psi) \\
 y_E &= u \sin \theta \cos \phi + v (\sin \theta \sin \phi \sin \psi + \cos \theta \cos \psi) \\
 &\quad + w (\sin \theta \sin \phi \cos \psi - \cos \theta \sin \psi) \\
 h &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi
 \end{aligned} \tag{I.2}$$

I.2 Equations 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 ci-dessus, les équations du modèle non linéaire a 6 degrés de liberté dans le système d'axes de l'avion sont (Stevens & Lewis, 1992) :

$$\begin{aligned}
 u &= rv - qw - g \sin \theta + \frac{\bar{q}SC_x + T}{m} \\
 v &= pw - ru + g \cos \theta \sin \phi + \frac{\bar{q}SC_y}{m} \\
 w &= qu - pv + g \cos \theta \cos \phi + \frac{\bar{q}SC_z}{m}
 \end{aligned} \tag{I.3}$$

$$\begin{aligned}
 I_x p - I_{xz} r &= \bar{q} S b C_q (I_z - I_y) qr + I_{xz} qp \\
 I_y q &= \bar{q} S c C_m (I_x - I_z) pr - I_{xz} p^2 - r^2 + h_{moteur} r \quad (I.4) \\
 I_z r - I_{xz} p &= \bar{q} S b C_n (I_y - I_x) pq - I_{xz} qr - h_{moteur} q
 \end{aligned}$$

où C_x, C_y, C_z, C_l, C_m et C_n sont les coefficients aérodynamiques. La poussée 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 $(V_t; \alpha; \beta)$ et $(u; v; w)$:

$$\begin{aligned}
 V_t &= \sqrt{u^2 + v^2 + w^2} & u &= V_t \cos \alpha \cos \beta \\
 \alpha &= \tan^{-1} \frac{w}{u} & v &= V_t \sin \alpha \cos \beta \\
 \beta &= \sin^{-1} \frac{v}{V_t} & w &= V_t \sin \alpha \sin \beta \quad (I.5)
 \end{aligned}$$

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 :

$$\begin{aligned}
 V_t &= \frac{\bar{q} S}{m} (C_x \cos \alpha \cos \beta - C_y \sin \alpha \cos \beta - C_z \sin \alpha \cos \beta) + \frac{T}{m} \cos \alpha \cos \beta \\
 &\quad + g(\cos \alpha \cos \beta \sin \beta + \sin \alpha \cos \beta \cos \beta - \sin \alpha \sin \beta \cos \beta) \\
 &= \frac{\bar{q} S}{m V_t \cos \alpha} (C_z \cos \alpha + C_x \sin \alpha) + q \tan \alpha (\rho \cos \alpha + r \sin \alpha) \\
 &\quad + \frac{g}{V_t \cos \alpha} (\cos \alpha \cos \beta \cos \beta + \sin \alpha \sin \beta) - \frac{T \sin \alpha}{m V_t \cos \alpha} \\
 &= \frac{\bar{q} S}{m V_t} (C_x \cos \alpha \sin \beta + C_y \cos \alpha \cos \beta - C_z \sin \alpha \sin \beta) + p \sin \alpha \cos \beta - r \cos \alpha \\
 &\quad + \frac{g}{V_t} \cos \alpha \sin \beta \cos \beta + \frac{\sin \alpha}{V_t} g \cos \alpha \sin \beta - g \sin \alpha \cos \beta \cos \beta - \frac{T \cos \alpha}{m} \quad (I.6)
 \end{aligned}$$

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 :

$$\begin{aligned} p &= (c_1 r + c_2 p - c_4 h_{moteur}) q + \bar{q} S b (c_3 C_l + c_4 C_n) \\ q &= (c_5 p + c_7 h_{moteur}) r - c_6 p^2 - r^2 + c_7 \bar{q} S c C_m \\ r &= (c_8 p - c_2 r - c_9 h_{moteur}) q + \bar{q} S b (c_4 C_l + c_9 C_n) \end{aligned} \quad (I.7)$$

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 sur 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	m	9,144
surface des ailes	m ²	27,87
corde aérodynamique moyenne	m	3.45
position de référence du centre de gravité	%	0.35
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 _degrees');  
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 _degrees/s');  
    disp ('2. _q_= _100 _degrees/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 (m ethode
% 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 e t
    % 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_l(i,j,k)+( CXlef_l(i,j) Cm0_l(i,j,26)) ...
(1 deltaLef_i (l)/25)+( q cbar)/(2 v) (Cmq_l(i)+...
dCmqLef_l(i) (1 deltaLef_i (l)/25))+ dCm_l(i)+...
(xcg_ref xcg) (CZ0_l(i,j,k)+( CXlef_l(i,j) ...
CZ0_l(i,j,26)) (1 deltaLef_i (l)/25)+( q cbar)/(2 v) ...
(CZq_l(i)+ dCZqLef_l(i) (1 deltaLef_i (l)/25)));
CZ0_l(i,j,k)+( CXlef_l(i,j) CZ0_l(i,j,26)) ...
(1 deltaLef_i (l)/25)+( q cbar)/(2 v) (CZq_l(i)+...
dCZqLef_l(i) (1 deltaLef_i (l)/25));
end
end
case 2
[ Cl0_l , Cllef_l , Clda20_l , Clda20Lef_l , Cldr30_l , Clr_l , dClrLef_l , ...
Clp_l , dClpLef_l , dClbeta_l , CY0_l , CYlef_l , CYda20_l , CYda20Lef_l , ...
CYdr30_l , CYr_l , dCYrLef_l , CYp_l , dCYpLef_l , Cn0_l , Cnlef_l , ...
Cnda20_l , Cnda20Lef_l , Cndr30_l , Cnr_l , dCnrLef_l , Cnp_l , dCnpLef_l , ...
dCnbeta_l ] = 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) = [ Cl0_l (i,j,k)+( Cllef_l (i,j) ...
                Cl0_l (i,j,26)) (1 deltaLef_i (l)/25)+( Cldr30_l (i,j) ...
                Cl0_l (i,j,26)) (deltar_i (l)/30)+( Clda20_l (i,j) ...
                Cl0_l (i,j,26)+( Clda20Lef_l (i,j) Cllef_l (i,j) ...
                (Clda20_l (i,j) Cl0_l (i,j,26))) (1 deltaLef_i (l)/25)) ...
                (deltaa_i (l)/20)+( p b)/(2 v) (Clp_l (i)+ dClpLef_l (i) ...
                (1 deltaLef_i (l)/25)+( r b)/(2 v) (Clr_l (i)+...
                dClrLef_l (i) (1 deltaLef_i (l)/25))+ dClbeta_l (i); ...
                CY0_l(i,j)+( CYlef_l (i,j) CY0_l(i,j)) ...
                (1 deltaLef_i (l)/25)+( CYdr30_l(i,j) CY0_l(i,j)) ...
                (deltar_i (l)/30)+( CYda20_l(i,j) CY0_l(i,j)+...
                (CYda20Lef_l(i,j) CYlef_l (i,j) (CYda20_l(i,j) ...
                CY0_l(i,j)) (1 deltaLef_i (l)/25)) (deltaa_i (l)/20)+...
                (p b)/(2 v) (CYp_l(i)+ dCYpLef_l(i) ...
                (1 deltaLef_i (l)/25)+( r b)/(2 v) (CYr_l(i)+...
                dCYrLef_l (i) (1 deltaLef_i (l)/25)); ...
                Cn0_l(i,j,k)+( Cnlef_l (i,j) Cn0_l(i,j,26)) ...
                (1 deltaLef_i (l)/25)+( Cndr30_l(i,j) Cn0_l(i,j,26)) ...
                (deltar_i (l)/30)+( Cnda20_l(i,j) Cn0_l(i,j,26)+...
                (Cnda20Lef_l(i,j) Cnlef_l (i,j) (Cnda20_l(i,j) ...
                Cn0_l(i,j,26))) (1 deltaLef_i (l)/25)) ...
                (deltaa_i (l)/20)+( p b)/(2 v) (Cnp_l(i)+ dCnpLef_l (i) ...
                (1 deltaLef_i (l)/25)+( r b)/(2 v) (Cnr_l(i)+...
                dCnrLef_l (i) (1 deltaLef_i (l)/25))+ dCnbeta_l (i) ...
                (cbar/b) (xcg_ref xcg) (CY0_l(i,j)+( CYlef_l (i,j) ...
                CY0_l(i,j)) (1 deltaLef_i (l)/25)+( CYdr30_l(i,j) ...
                CY0_l(i,j)) (deltar_i (l)/30)+( CYda20_l(i,j) ...
                CY0_l(i,j)+( CYda20Lef_l(i,j) CYlef_l (i,j) ...

```

```

        (CYda20_I(i,j) CY0_I(i,j))) (1 deltaLef_i (l)/25)) ...  

        (deltaa_i (l)/20)+( p b)/(2 v) (CYp_I(i)+dCYpLef_I(i) ...) ...  

        (1 deltaLef_i (l)/25))+( r b)/(2 v) (CYr_I(i)+...  

        dCYrLef_I (i) (1 deltaLef_i (l)/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 ...
            ... croisee pour chaque etape:');
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 entraigne 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;
        perfentrenementprecedent = 10;

        % validation croisee pour determiner le nombre d'epoques
        % d'entrainement:
        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 cote et un nouveau est entraigne.

            % creation d'un reseau a m1 neurones sur la couche cachee
            net = newff(minmax(inputTrain), [m1 3], 'tansig' 'purelin' 'g',...
                'trainlm');
            % il est entraigne une epoque a la fois :
            net.trainParam.epochs = 1;
            % compteurs et variables d'arret d'entrainement :
            j = 1;
            cont = 0;
            compteur = 0;
            % tant que le maximum d'epoques d'entrainement n'est pas
            % atteint et que la performance du reseau continue de diminuer
            % celui ci est entraigne
            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) >= perfentrenementprecedent)
                    compteur = compteur + 1;
                else compteur = 0;
                end
                % au bout de cinq epoques o la performance ne change pas,
                % l'entrainement prend fin
                if (compteur == 10)
                    cont = 1;
                    display('entrainement sarrête à cause de la prematurité');
                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 à m1 donne');
                end
                perfentrenementprecedent = tr.vperf(1,2);
                j=j+1;
            end % on sort de la boucle d'entrainement

        end % on sort de la boucle donc : perflocale <= performance
    end
end

```

```
% le reseau N a m1 neurones est :  
    resN = res ;  
    perfres = perfocale ;  
% on choisi le meilleur :  
if (perfres < perfM1)  
    display ('choix _du_reseau_a_m1_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 (2000) 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 $\Omega \subset \mathbb{R}^n$.

Définition 1 Le système IV.1 est dit linéarisable entrée-état s'il existe une région de \mathbb{R}^n , un difféomorphisme $\phi : \Omega \rightarrow \mathbb{R}^n$ et un retour d'état

$$u = \psi(x) + \phi(x)v$$

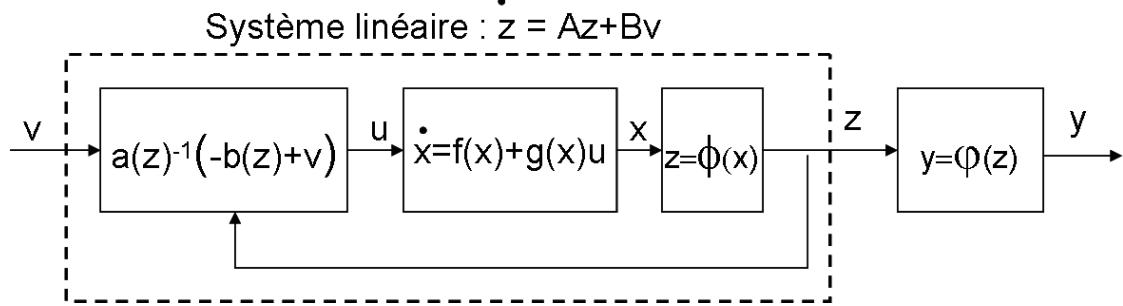


Figure IV.1 Linearisation entree etat

tel que le nouveau vecteur d'état $\underline{z} = (\underline{x})$ et la nouvelle entrée v vérifient la relation

$$\underline{z} = Az + Bv$$

Le schéma bloc du système linearisé 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 linearisable entree etat, dans une région de R^n , par retour d'état statique si les deux conditions suivantes sont satisfaites :

$$r = \text{rang}(g; \text{ad } g; \dots; \text{ad}^{n-1} g) = n, \quad 8x 2$$

$\text{vect}(g; \text{ad } g; \dots; \text{ad}^{n-1} g)$ est involutive sur

$\text{ad } g$ 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

$$\begin{aligned} z &= [\omega_1(x); L_{f^{-1}}(x); \dots; L_{f^{n-1}}(x)]^T \\ (x) &= \frac{L_g L_{f^{-1}}(x)}{L_g L_{f^{n-1}}(x)} \\ (x) &= \frac{1}{L_g L_{f^{n-1}}(x)} \end{aligned} \quad (\text{IV.3})$$

où $\omega_1(x)$ est solution du système d'équations suivant :

$$\begin{aligned} L_g L_{f^{-1}}^i(x) &= 0 \text{ pour } i = 0, \dots, n-2 \\ L_g L_{f^{n-1}}(x) &\neq 0 \end{aligned} \quad (\text{IV.4})$$

IV.2 Linearisation entrée-sortie

Considérons le système suivant, non linéaire MIMO et à une entrée de commande :

$$\begin{aligned} \underline{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \quad (\text{IV.5})$$

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 $\Omega \subset \mathbb{R}^n$.

Le degré relatif r de ce système est alors déterminé.

Définition 2 Soit Δ 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

$$\forall x \in \Delta \quad L_g L_f^k h_i(x) = 0 \quad 1 \leq j \leq m, 1 \leq i \leq p, 0 \leq k \leq r_i - 1$$

et

$$A(x_0) = \begin{pmatrix} 2 & 3 & 2 & 3 \\ L_{g_1} L_f^{r_1} h_1(x) & \cdots & L_{g_m} L_f^{r_1} h_1(x) \\ L_{g_1} L_f^{r_2} h_2(x) & \cdots & L_{g_m} L_f^{r_2} h_2(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_p} h_p(x) & \cdots & L_{g_m} L_f^{r_p} h_p(x) \end{pmatrix}$$

est inversible.

Le degré relatif total d'un système MIMO est alors $r = \sum_{i=1}^p r_i$.

Le difféomorphisme transforme le système original non linéaire et affine en la commande en un système linéaire sous la forme canonique de Brunovsky :

$$z = (\begin{matrix} z_1 & z_2 & \dots & z_p \end{matrix}) = (\begin{matrix} z_{11} & z_{12} & \dots & z_{1r_1} \\ z_{21} & z_{22} & \dots & z_{2r_2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{p1} & z_{p2} & \dots & z_{pr_p} \end{matrix}) = (\begin{matrix} 2 & 3 & 2 & 3 \\ L_f^{r_1} h_1(x) & L_f^{r_2} h_2(x) & \cdots & L_f^{r_p} h_p(x) \end{matrix}) \quad (\text{IV.6})$$

$$\dot{z} = \frac{d}{dt} (\begin{matrix} z_1 & z_2 & \dots & z_p \end{matrix}) = (\begin{matrix} z_{11} & z_{12} & \dots & z_{1r_1} \\ z_{21} & z_{22} & \dots & z_{2r_2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{p1} & z_{p2} & \dots & z_{pr_p} \end{matrix}) = (\begin{matrix} 2 & 3 & 2 & 3 \\ L_f^{r_1} h_1(x) + \sum_{k=1}^m L_{g_k} L_f^{r_1} h_1(x) u_k & L_f^{r_2} h_2(x) + \sum_{k=1}^m L_{g_k} L_f^{r_2} h_2(x) u_k \\ \vdots & \vdots & \ddots & \vdots \\ L_f^{r_p} h_p(x) + \sum_{k=1}^m L_{g_k} L_f^{r_p} h_p(x) u_k & \vdots & \vdots & \vdots \end{matrix}) \quad (\text{IV.7})$$

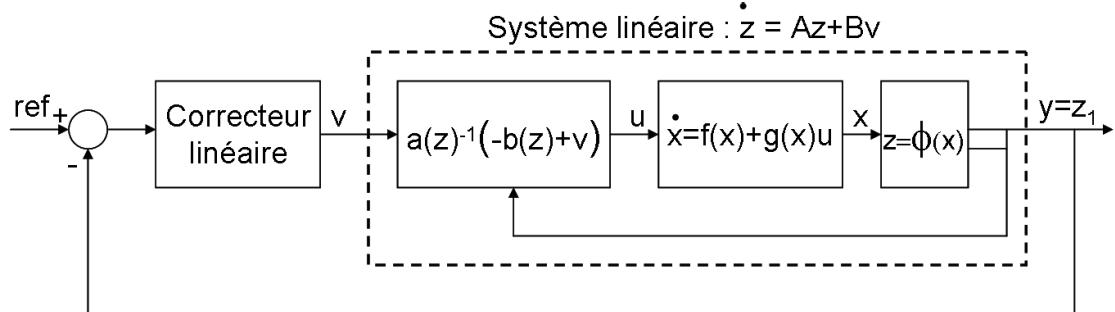


Figure IV.2 Linearisation entrée-sortie

Le retour linearisant résultant peut alors être défini par :

$$u = A^{-1}(v - B) \quad (\text{IV.8})$$

ou

$$\begin{aligned} A &= f A_{ij} & g &= f L_g L_f^{r_i-1} h_i(x) g \\ B &= f B_{ij} g & &= f L_f^{r_i} h_i(x) g \end{aligned} \quad (\text{IV.9})$$

Théorème 2 Un système non linéaire est complètement linearisable entrée-sorties dans un domaine $D_0 \subset D$ 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 linearisation est une méthode relativement simple. Une fois le système linearisé par retour d'état, une correction classique peut être utilisée en cascade, constituant ainsi une boucle externe (voir IV.2).

ANNEXE V

COMMANDER PRÉdictive LINéaire

La commande prédictive repose sur les grandes idées suivantes (Dumur, 2002)

la création d'un état 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 basé sur la formulation de Maciejowsky (1990).

Soit le système à commander suivant:

$$\begin{aligned}\underline{x} &= f(x; r) \\ \bar{y} &= h(x; r)\end{aligned}\tag{V.1}$$

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 discrétisé. À chaque période d'échantillonnage k , on

obtient :

$$\begin{matrix} x(k+1) \\ \downarrow & \downarrow \\ y(k) \\ \downarrow & \downarrow \\ z(k) \\ \downarrow & \downarrow \\ u(k) \end{matrix} = \begin{matrix} A_k x(k) + B_k r(k) \\ C_k x(k) + D_k r(k) \end{matrix} \quad (V.2)$$

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 & Baas, 2006) :

$$J(k) = \sum_{i=1}^{H_p} (\hat{y}(k+i|k) - y_{ref}(k+i|k))^T Q (\hat{y}(k+i|k) - y_{ref}(k+i|k)) + \sum_{i=0(H_c)} r(k+i|k)^T R r(k+i|k) \quad (V.3)$$

où $\hat{y}(k+i|k)$ est la prévision des sorties à instants après le temps présent k , basée sur les données jusqu'au temps k ; H_p est l'horizon de prévision, H_c l'horizon de contrôle et H_c la période de changement du contrôle.

représente un état appelé "slack state". Il permet de transformer des contraintes discontinues en contraintes plus souples. r est le gain associé à cet état.

Afin d'obtenir les prévisions, un modèle de prévision 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 intégrateurs

à n de convertir le changement sur la commande r en la valeur de la commande r. La commande prédictive requiert la connaissance parfaite du modèle du système à asservir ; à n 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 prédition est alors :

$$\begin{aligned}
 & \begin{matrix} 2 & 3 & 2 \\ 6 & 6 & 6 \\ 6 & 6 & 6 \\ 4 & 4 & 4 \end{matrix} \hat{x}(k+1) = \begin{matrix} A_k & 0_{n \times (p_1+p_2)} \\ 0_{(p_1+p_2) \times n} & I_{p_1+p_2} \end{matrix} \hat{x}(k) + \begin{matrix} 3 & 2 & 3 & 2 & 3 \\ 7 & 6 & 7 & 6 & 7 \\ 7 & 6 & 7 & 6 & 7 \\ 5 & 4 & 5 & 4 & 5 \end{matrix} \hat{d}(k) + \begin{matrix} 0_{(p_1+p_2) \times m} \\ D_k \end{matrix} \hat{d}(k) + \begin{matrix} r(k) \\ r(k-1) \\ I_m \end{matrix} \\
 & \begin{matrix} r(k) \\ 2 & 3 \\ 6 & 6 \\ 6 & 6 \\ 4 & 4 \end{matrix} = \begin{matrix} 0_{m \times n} & 0_{m \times (p_1+p_2)} \\ I_{p_1+p_2} & 0_{(p_1+p_2) \times m} \end{matrix} \hat{x}(k) + \begin{matrix} 2 & 3 \\ 3 & 6 \\ 6 & 6 \\ 4 & 4 \end{matrix} \hat{d}(k) + D_k r(k) \\
 & \hat{y}(k) = C_k \hat{x}(k) \\
 & \hat{z}(k) = D_k \hat{x}(k)
 \end{aligned} \tag{V.4}$$

À n de simplifier la notation, il est écrit sous la forme suivante :

$$\begin{aligned}
 \hat{x}(k+1) &= \bar{A}_k \hat{x}(k) + \bar{B}_k r(k) \\
 \hat{w}(k) &= \bar{C}_k \hat{x}(k) + \bar{D}_k r(k)
 \end{aligned} \tag{V.5}$$

V.2 Le problème de programmation quadratique équivalent

Le modèle de prédition permet de calculer les prédictions utilisées par le critère de performance.

$$\begin{aligned}
 & \frac{\hat{w}(k+1jk)}{C_k A_k} = \frac{\hat{w}(k+2jk)}{C_k A_k^2} = \dots = \frac{\hat{w}(k+H_cjk)}{C_k A_k^{H_c}} = \hat{w}(k) \\
 & \frac{\hat{w}(k+H_c+1jk)}{C_k A_k^{H_c+1}} = \dots = \frac{\hat{w}(k+H_pjk)}{C_k A_k^{H_p}}
 \end{aligned}$$

Cette relation est réécrite avec des notations compactes :

$$\overline{W}(k) = \overline{Aa_k}^{\wedge}(k) + \overline{Bb_k} \quad \overline{R}(k) \quad (V.7)$$

Le critère de performance devient alors :

$$\begin{aligned} J(k) = & \quad \overline{R}(k)^T \overline{Bb_{k_y}}^T \overline{QBb_{k_y}} \quad \overline{R}(k) + \quad \overline{R}(k)^T \overline{2Bb_{k_y}}^T \overline{Q} \quad \overline{Aa_{k_y}}^T(k) \quad \overline{Y_{ref}}(k) \\ & + \quad \overline{Aa_{k_y}}^T(k) \quad \overline{Y_{ref}}(k) \quad ^T \overline{Q} \quad \overline{Aa_{k_y}}^T(k) \quad \overline{Y_{ref}}(k) \quad + \quad \overline{R}(k)^T \overline{R} \quad \overline{R}(k) + \\ & \quad . \end{aligned} \quad (V.8)$$

$\omega \overline{Y_{ref}}(k) = \begin{bmatrix} h \\ y_{ref}(k+1) \\ \vdots \\ y_{ref}(k+H_p) \end{bmatrix}^T$, l'indice y indique que seules les lignes associées aux sorties 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" :

$$\underline{u}_k \leq \alpha(k + 1jk); \dots; \alpha(k + H_pjk) \leq \bar{u}_k \quad (V.9)$$

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. A cette fin, la variable additionnelle \underline{z} est introduite :

$$\underline{z}_k \leq \hat{z}(k + 1jk); \dots; \hat{z}(k + H_pjk) \leq \bar{z}_k \quad (V.10)$$

Afin que les contraintes originelles "dures" ne soient pas transgessées inutilement, le gain doit être choisi suffisamment grand.

L'équation V.6 permet d'exprimer les contraintes sur z en fonction de $R(k)$. En effet, $\bar{Z}(k) = [\underline{z}(k + 1jk); \dots; \hat{z}(k + H_pjk)]^T$ est égale par l'équation suivante :

$$\bar{Z}(k) = \overline{Aa_{k_z}} \gamma(k) + \overline{Bb_{k_z}} \bar{R}(k) \quad (V.11)$$

où l'indice z indique que seules les lignes associées aux sorties sont sélectionnées.

La contrainte V.10 devient alors :

$$\begin{array}{ll} \overline{Bb_{k_z}} & \overline{R}(k) \\ \overline{Bb_{k_z}} & \overline{R}(k) \end{array} \quad \begin{array}{l} \underline{z}_k + \overline{Aa_{k_z}}^\gamma(k) \\ \bar{z}_k - \overline{Aa_{k_z}}^\gamma(k) \end{array} \quad (V.12)$$

De manière analogue on peut réécrire les contraintes sur:

$$\begin{array}{ll} \overline{Bb_{k_u}} & \overline{R}(k) \\ \overline{Bb_{k_u}} & \overline{R}(k) \end{array} \quad \begin{array}{l} \underline{u}_k + \overline{Aa_{k_u}}^\gamma(k) \\ \bar{u}_k - \overline{Aa_{k_u}}^\gamma(k) \end{array} \quad (V.13)$$

Le problème de programmation quadratique équivalente à résoudre à chaque itération en fonction de $R(k)$ et est le suivant:

$$\begin{aligned} \min_{R(k)} & \quad \overline{R}(k)^T \overline{Bb_{k_y}}^T \overline{Q} \overline{Bb_{k_y}} \overline{R}(k) + \overline{R}(k)^T 2 \overline{Bb_{k_y}}^T \overline{Q} \overline{Aa_{k_y}}^\gamma(k) \overline{Y_{ref}}(k) \\ & + \overline{Aa_{k_y}}^\gamma(k) \overline{Y_{ref}}(k) \\ & + \overline{Bb_{k_u}}^T \overline{Q} \overline{Aa_{k_u}}^\gamma(k) \overline{Y_{ref}}(k) + \overline{R}(k)^T \overline{R} \overline{R}(k) + \\ & + \overline{Bb_{k_z}}^T \overline{Q} \overline{Aa_{k_z}}^\gamma(k) \overline{Y_{ref}}(k) \\ \text{avec} & \quad \begin{array}{ll} \overline{Bb_{k_u}} & 0 \\ \overline{Bb_{k_z}} & 0 \\ \overline{Bb_{k_z}} & 1 \\ \overline{Bb_{k_z}} & 1 \end{array} \quad R(k) = \begin{array}{ll} 2 & 3 \\ 2 & 3 \\ 6 & 6 \\ 6 & 6 \end{array} \quad \begin{array}{ll} \underline{u}_k + \overline{Aa_{k_u}}^\gamma(k) & \underline{z}_k + \overline{Aa_{k_z}}^\gamma(k) \\ \bar{u}_k - \overline{Aa_{k_u}}^\gamma(k) & \bar{z}_k - \overline{Aa_{k_z}}^\gamma(k) \end{array} \\ & \quad \begin{array}{ll} 0 & 1 \\ 0 & 0 \\ 4 & 4 \\ 4 & 5 \end{array} \quad \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \end{array} \end{aligned} \quad (V.14)$$

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 $[V_t; \dot{V}_t; \dot{\alpha}; \dot{\beta}; \dot{\gamma}; p; q; r; x_E; y_E; h]$. 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 e_{com} , a_{com} , r_{com} et \dot{r}_{com} . De plus le calcul de la valeur de \dot{e}_{ref} en induit un autre \dot{r}_{ref_calcul} . Le vecteur d'état du F-16 est ainsi :

$$[V_t; \dot{V}_t; \dot{\alpha}; \dot{\beta}; \dot{\gamma}; p; q; r; x_E; y_E; h; e_{com}; \dot{e}_{ref_calcul}; P_a; r_{com}; a_{com}; \dot{r}_{ref_calcul}].$$

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 $\dot{\alpha}$ et $\dot{\beta}$ peuvent présenter une singularité lorsque $\dot{\alpha} = 90^\circ$. 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 $\dot{\alpha}$ est compris en valeur absolue

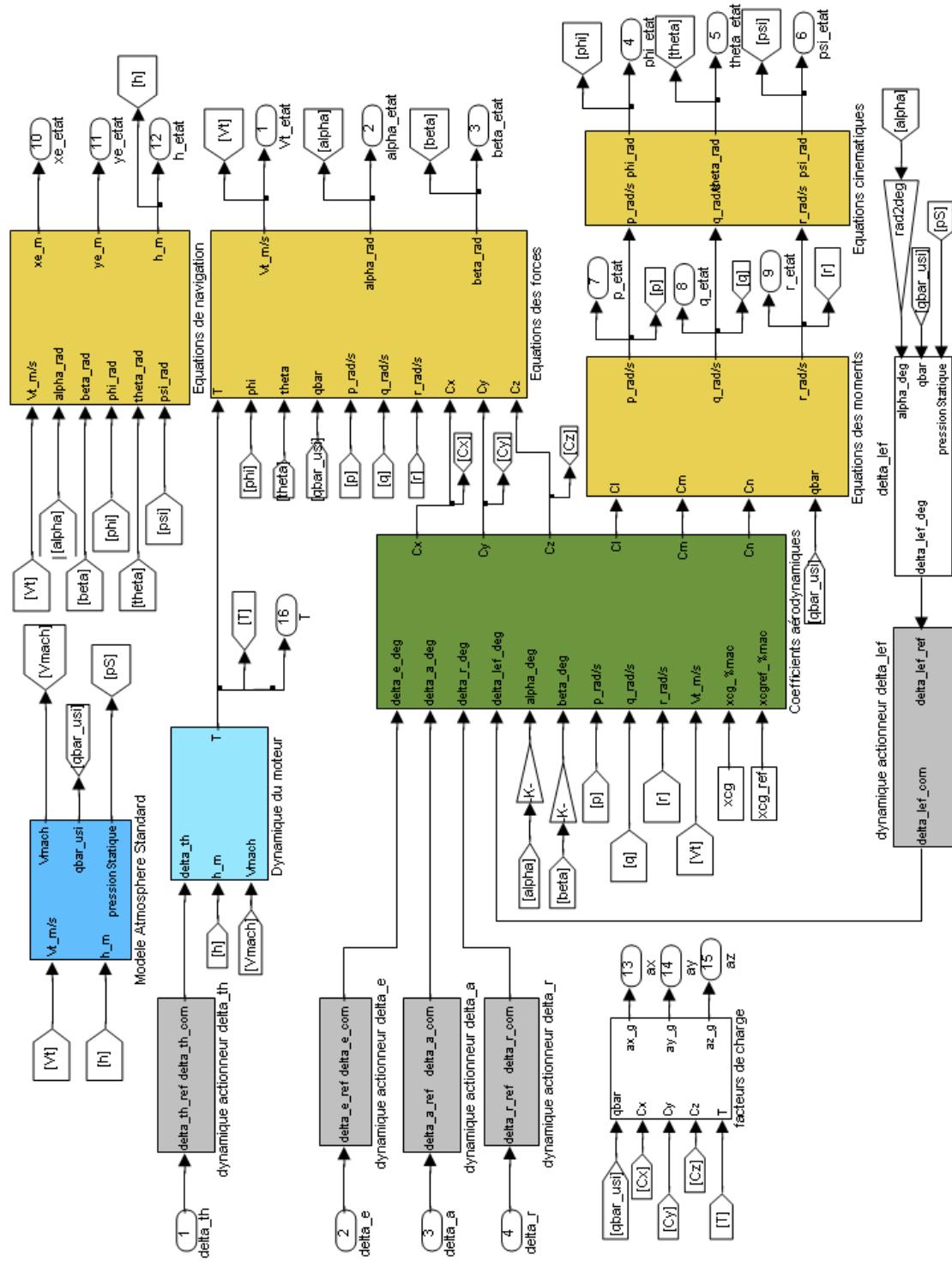


Figure VI.1 Simulateur

entre 89.99 et 90.01 , la valeur utilisée pour le calcul de –et –est soit 89.99 soit 90.01 , selon la valeur deq.

VI.3 Le modèle aérodynamique

Le modèle aérodynamique utilisé dépend de la simulation effectuer, selon qu'il s'agisse du comportement du F-16 "real" ou du F-16 identie.

VI.3.1 Pour la simulation du comportement "real" du F-16

Les données aérodynamiques proviennent d'essais en soufflerie statiques et dynamiques à faible vitesse sur un modèle réduit 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 prêter plus efficacement. 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 donne est égale à celle de son plus proche voisin dans l'espace des variables d'état. 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.

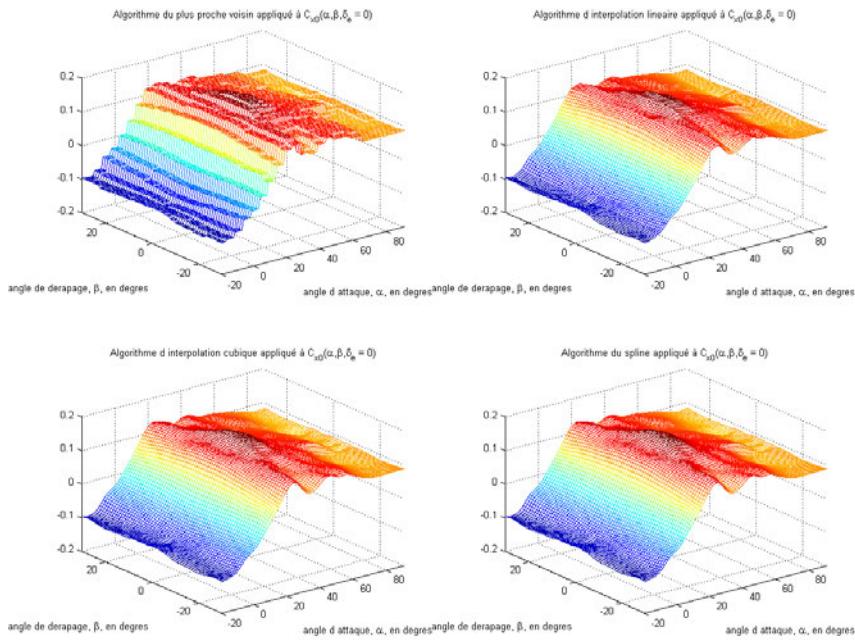


Figure VI.2 Comparaison des di erents algorithmes d'interpolation en 2D

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 don ees. Cet algorithme r esulte moins de m emoire que le pr ecedent, mais  g  rement plus de temps de calcul. Le r esultat de l'interpolation est une fonction continue et d erivable 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 eriv e  soit continue. La d eriv e  seconde peut ne pas  tre continue, elle peut effectivement faire des sauts aux points de l'ensemble des don ees. Cette m ethode est plus coûteuse a la fois en m emoire et en temps de calcul que les deux premiers, cependant le r esultat de l'interpolation est une fonction continue et de d eriv e  continue.

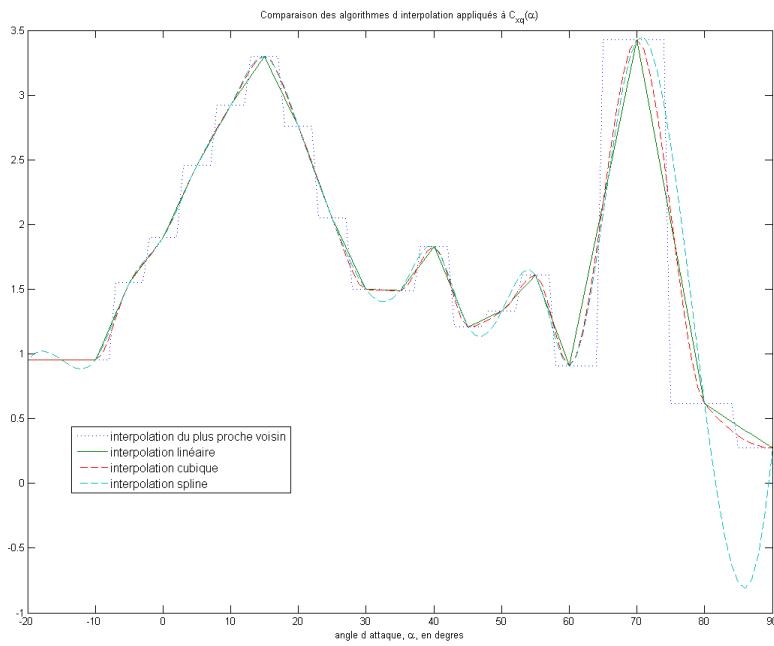


Figure VI.3 Comparaison des différents algorithmes d'interpolation en 1D

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 coûts 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 identie

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

Cette fonction transforme la position de θ_{th} , comprise entre 0 et 1, en un niveau de puissance, compris entre 0 et 100 :

$$P_c(\theta_{th}) = \begin{cases} 8 & \text{si } \theta_{th} \leq 0.77 \\ 64.94 + 217.38 \cdot \theta_{th} & \text{si } 0.77 < \theta_{th} \leq 117.38 \\ 117.38 & \text{si } \theta_{th} > 117.38 \end{cases} \quad (\text{VI.1})$$

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 :

$$P_a = \frac{1}{\tau} (P_c - P_a) \quad (\text{VI.2})$$

$$P_c = \begin{cases} 8 & \text{si } P_c(\theta_{th}) \leq 50 \text{ et } P_a \geq 50 \\ 60 & \text{si } P_c(\theta_{th}) \leq 50 \text{ et } P_a < 50 \\ 40 & \text{si } P_c(\theta_{th}) > 50 \text{ et } P_a \geq 50 \\ P_c(\theta_{th}) & \text{si } P_c(\theta_{th}) > 50 \text{ et } P_a < 50 \end{cases} \quad (\text{VI.3})$$

$$\begin{aligned}
 \frac{1}{\text{moteur}} &= \begin{cases} 8 & \text{si } P_c(\text{th}) \geq 50 \text{ et } P_a \geq 50 \\ 5 & \text{si } P_c(\text{th}) \geq 50 \text{ et } P_a < 50 \\ 5 & \text{si } P_c(\text{th}) < 50 \text{ et } P_a \geq 50 \\ \cdot \frac{1}{8} & \text{si } P_c(\text{th}) < 50 \text{ et } P_a < 50 \end{cases} \\
 \frac{1}{\text{moteur}} &= \begin{cases} 1 & \text{si } (P_c(\text{th}) - P_a) \geq 25 \\ 0;1 & \text{si } (P_c(\text{th}) - P_a) = 50 \\ \cdot 1;9 - 0;036(P_c(\text{th}) - P_a) & \text{si } 25 < (P_c(\text{th}) - P_a) < 50 \end{cases}
 \end{aligned}$$

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 h 15240 m et 0 M 1 dans le rapport de la NASA (Nguyen et al, 1979).

$$T = \begin{cases} \geq T_{\text{idle}} + (T_{\text{mil}} - T_{\text{idle}}) \frac{P_a}{50} & \text{si } P_a < 50 \\ \geq T_{\text{mil}} + (T_{\text{max}} - T_{\text{mil}}) \frac{P_a - 50}{50} & \text{si } P_a \geq 50 \end{cases} \quad (\text{VI.4})$$

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 :

$$\begin{aligned}
 T &= T_0 - 0;0065h & \text{si } h < 11000 \text{ m} \\
 T &= T_0 - 0;006511000 & \text{si } h \geq 11000 \text{ m}
 \end{aligned} \quad (\text{VI.5})$$

où T_0 est la température moyenne au niveau de la mer, soit 288,15 K

La vitesse du son, en m:s⁻¹, est :

$$a = \sqrt{\frac{p}{\gamma R T}} \quad (\text{VI.6})$$

où γ est le coefficient adiabatique de l'air ($\gamma = 1,402$ à $P_0 = 1 \text{ atm} = 101325 \text{ Pa}$ 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, en kg:m⁻³, est :

$$\rho = \rho_0 e^{(-\frac{g}{RT} h)} \quad (\text{VI.8})$$

La pression dynamique \bar{P} , en Pa, est alors :

$$\bar{P} = \frac{1}{2} V^2 \quad (\text{VI.9})$$

et la pression statique P_s , en Pa, est :

$$P_s = P_0 e^{-\frac{g}{R T} h} \quad (\text{VI.10})$$

VI.6 La dynamique des actionneurs

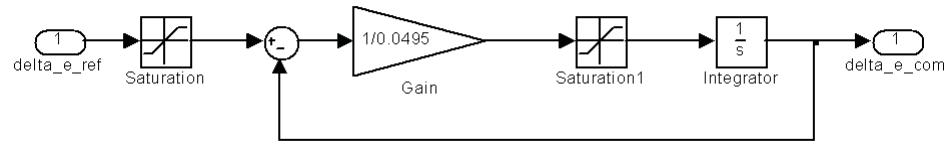


Figure VI.4 Dynamique des actionneurs

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	déflection maximale	taux de rotation limite	constante de temps	T_e
Elevateur	25	60 :s ⁻¹	0;0495s	
Ailerons	21;5	80 :s ⁻¹	0;0495s	
Gouvernail	30	120 :s ⁻¹	0;0495s	
Bord d'attaque	0 25	25 :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 :

$$\dot{p} = \dot{q} = \dot{r} = \dot{V}_t = \dot{\gamma} = \dot{\omega} = 0 \quad (\text{VI.11})$$

On distingue ensuite plusieurs manœuvres, qui imposent chacune des contraintes spécifiques.

Vola l'équilibre avec ailes horizontales :

$$\dot{\gamma} = \dot{\omega} = \dot{\alpha} = \dot{\beta} = 0 \quad (\text{VI.12})$$

Virage coordonné où $\dot{\alpha}$ est le taux de rotation constant :

$$\dot{\gamma} = \dot{\omega} = \dot{\alpha} = 0 \quad (\text{VI.13})$$

Montée stabilisée où $\dot{\beta}$ est le taux de montée constant :

$$\dot{\gamma} = \dot{\omega} = \dot{\beta} = 0 \quad (\text{VI.14})$$

Roulis stabilisé où $\dot{\alpha}$ est le taux de roulis constant :

$$\dot{\gamma} = \dot{\omega} = \dot{\alpha} = 0 \quad (\text{VI.15})$$

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 ma noeuvre
%   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 ('');
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 gaz , 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 effectuee
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],[...
            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],[...
            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_reponse:');
switch choix_perturbation
    case 1
        disp ('-');
        perturbation_deltae = input ('veuillez_entrer_la_perturbation_...
            _d_sir_e_sur_la_value_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 ('-');
    end

```

```
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 e s : ' );  
t_perturbation_deltaa = input ('veuillez entrer le temps initial ...  
de cette perturbation en secondes : ' );  
delta_perturbation_deltaa = input ('veuillez entrer la dure 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 e s : ' );  
t_perturbation_deltar = input ('veuillez entrer le temps initial ...  
de cette perturbation en secondes : ' );  
delta_perturbation_deltar = input ('veuillez entrer la dure 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 );
```