
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

École Nationale Polytechnique

Département d'Automatique



Projet de fin d'études
En vue de l'obtention du diplôme
D'Ingénieur d'Etat
Thème :

Modélisation, Simulation, Conception de lois de Commande et Diagnostic d'un Système d'Inversion de Poussée d'un Avion.

Étudié par :

MOKRANE Hani Kamil
BENSALAH Amina

Proposé et dirigé par :

M.DEBIANE Achour
M.TADJINE Mohamed

Juin 2013

Remerciements

À l'issue de ces cinq années de dures labeurs à l'Ecole Nationale Polytechnique, nous tenons à remercier l'ensemble des personnes ayant contribué à notre épanouissement tant sur le plan intellectuel que social.

Après quatre mois de stage au sein du pôle R&D de l'entreprise CERTIA, nous remercions en premier lieu son manager M.DEBIANE Achour qui nous a d'abord gratifié de sa confiance en nous mettons au cœur du développement d'un projet de l'envergure de PRESAGE “ Plateforme Réelle Et Simulée d'Actionnement Générique et Évolutive ”, plus encore, il n'a cessé de nous encourager et nous aider en nous prodiguons de précieux conseils dans notre démarche scientifique.

Nous tenons aussi à remercier notre encadreur à l'Ecole Nationale Polytechnique M.TADJINE Mohamed qui avec ses compétences et qualités scientifiques a permis le développement de nombreuses idées innovantes.

Nous exprimons ainsi toute notre reconnaissance aux deux personnes cités ci haut pour leur soutien permanent et sympathie dont ils ont su faire preuve tout en espérant que la collaboration entre l'Ecole Nationale polytechnique et CERTIA R&D ne s'arrêtera pas en si bon chemin.

Nous tenons à exprimer notre gratitude et notre plus grande estime à M.BOUCHERIT Mohamed Seghir qui nous a fait l'honneur d'être le président de ce jury.

Aussi, nous exprimons nos plus vifs remerciements à M.CHAKIR Messaoud qui nous honore en participant à l'évaluation de ce travail.

Par ailleurs, nous tenons à adresser nos plus sincères remerciement à toutes l'équipe CERTIA R& D, équipe tellement unis que vous êtes devenu une famille : Nassim le polytechnicien,

Abdelkader, Hicham, Smati, Yacine, Nassim, Saber et Taous sans oublier Mme RARRBO Ouahiba et le restaurateur “ BNINE ” sans qui nos pauses déjeuner aurait été bien ternes .

Nous tenons à remercier tous nos enseignants et encadrement pédagogique de l'Ecole Nationale polytechnique et toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Merci à tous ceux qui comptent, famille et amis, pour leur soutien sans faille.

Nous saluons enfin le lecteur de ces lignes et espérons qu'il trouvera ce qu'il est venu chercher dans les pages qui suivent.

Dédicace

Je dédie ce travail à mes parents qui ont toujours été là pour moi, mes amis : Dalia, Amina et leurs moments de folie, Mourad, Chafik, Moh... à tous ceux que j'ai oublié...

À une amie qui m'est très chère, Aziza, une personne qui m'a soutenu avec un état d'esprit « no stress ».

Mes cousines Racha et Mimi qui m'ont témoigné leurs soutiens dans les bons comme dans les mauvais moments.

L'équipe CERTIA R&D ne pouvait répondre absente à ces dédicaces : Au poste de roi du Matlab et du Mupad NASSIM, expert « bond graph » ABDELKADER et sans oublier le géant du mailing HICHEM. Par ailleurs, je voudrais adresser un vif remerciement à mon binôme AMINA, pas seulement pour son travail sans relâche, son professionnalisme, son sérieux, son abnégation et sa volonté de fer, mais aussi et surtout pour sa capacité me supporter par moment ...

Kamil

Dédicace

À la mémoire de ma grand-mère

À mes chers parents,

À mes sœurs Asma et Nour el Houda,

À mes frères Ibrahim et Ilyes,

À toute ma famille,

À mon binôme Kamil,

À toutes mes amies de NEW,

À Krimo, Hayet, Rekia et Mohamed,

À Louisa et ses parents,

À Salima et Lyna,

À toute l'équipe de CERTIA R&D,

À tous mes amis de l'ENP,

À tous ceux que j'aime,

Je dédie ce travail,

Amina

Table des matières

Acronymes	xiii
Introduction Générale	1
	5
I Modélisation	5
	6
1 Le système d'inversion de poussée	7
1.1 Introduction	7
1.2 Description du système d'inverseur de poussée et sa tâche au niveau de l'avion	8
1.3 Intérêt technico-commercial	8
1.4 Principe de fonctionnement	9
1.5 Les différents types d'inverseurs	10
1.5.1 Inverseurs à portes pivotantes.	10
1.5.2 Inverseurs à cascades.	11
1.5.3 Inverseurs à obstacles.	12
1.6 Limitation d'emploi de l'inverseur de poussée	13
1.7 Exigences de l'inverseur de poussée	13
1.8 Conclusion	14
2 Conception d'une plateforme virtuelle du système d'inversion de poussée	15
2.1 Outils de modélisation et de simulation dynamique Dymola/Modelica	15
2.1.1 Environnement de modélisation et simulation dynamique Dymola	16
2.1.1.1 Architecture de Dymola	17
2.1.1.2 Interface graphique de Dymola	18
2.1.1.3 Bibliothèques de Dymola	19

2.1.2	Le langage Modelica	20
2.1.2.1	Points forts de Modelica	20
2.1.2.2	Les points faibles de Modelica	21
2.1.3	Exemple de conception d'un modèle Modelica sous Dymola	21
2.1.3.1	Construction d'un modèle élémentaire	21
2.1.3.2	Construction d'un assemblage de modèles	21
2.2	Modélisation du système d'inversion de poussée	22
2.2.1	Description du fonctionnement des sous-système d'inverseur de poussée	23
2.2.1.1	Sous-Système d'actionnement	23
2.2.1.2	Sous-système de chargement	28
2.2.2	La modélisation sous Dymola	30
2.2.2.1	Sous-système d'actionnement	30
2.2.2.2	Sous-système de chargement	33
2.3	Conclusion	34
		15
		35
II	Commande	35
		36
3	Commande par H_∞ Loopshping	37
3.1	Introduction	37
3.2	L'Automatique et la commande robuste	37
3.3	Modèles d'état du système d'inverseur de poussée	39
3.3.1	Représentation d'état à six variables	40
3.3.1.1	Modèle non linéaire	40
3.3.1.2	Modèle linéaire	42
3.3.2	Représentation d'état à quatre variables	42
3.3.2.1	Modèle non linéaire	43
3.3.2.2	Modèle linéaire	43
3.3.3	Simulation du système en BO	44
3.3.3.1	Systèmes non linéaires	44
3.4	Synthèse de la loi de commande par H_∞ Loopshaping	45
3.4.1	Définitions	45

3.4.2	Problème H_∞ standard	47
3.4.3	Problème de sensibilité mixte	48
3.5	Application au système d'inverseurs de poussée	51
3.5.1	Représentation d'état du système	51
3.5.2	Le système à 5 états	53
3.5.3	Résultat de simulation	55
3.5.3.1	Simulation du système en boucle ouverte	55
3.5.3.2	Simulation du système en BF sans perturbations en sortie	56
3.5.3.3	Simulation du système en BF avec perturbations en sortie	57
3.5.3.4	Simulation en présence d'erreurs de modélisation	59
3.6	Discussion	61
3.7	Conclusion	62
4	Commande par Backstepping	63
4.1	Introduction	63
4.2	Rappels et définitions d'automatique	64
4.2.1	Notion de stabilité au sens de Lyapunov	64
4.2.2	Éléments de théorie	64
4.3	Théorie du Backstepping	66
4.3.1	Principe de la technique du Backstepping	67
4.3.2	Système sous forme STRICT FEEDBACK	69
4.3.2.1	Système monodimensionnels sous forme STRICT FEEDBACK	69
4.3.2.2	Système multidimensionnels sous forme STRICT FEEDBACK	70
4.3.3	Commande en poursuite de trajectoire référence par Backstepping	70
4.3.3.1	Modèle sans incertitudes	70
4.3.3.2	Modèle avec incertitudes	72
4.4	Application au système d'inversion de poussée	73
4.4.1	Conception et application de la commande par backstepping	73
4.4.2	Simulation et discussion	74
4.4.2.1	Modèle sans incertitudes	75
4.4.2.2	Modèle avec incertitudes	77
4.5	Conclusion	78
		63
	Conclusion Générale	79

Bibliographie	81
A Modelica	89
A.1 Historique et généralités	89
A.2 Caractéristiques de Modelica	90
A.3 Éléments du langage Modelica	91
A.4 Les étapes de traduction du compilateur Modelica	96
B Annexe B	98
B.1 Applications Lipschitziennes	98
B.2 Preuve de la stabilité du modèle sans incertitudes	98
B.3 Preuve de la stabilité du modèle avec incertitudes	100
C Annexe C	101
C.1 Les expressions des fonctions non-linéaires	101
C.2 Les expressions des régulateurs par H_∞ dans les 3 phases	102

Table des figures

1.1	Vue éclatée d'une nacelle d'avion (BEN11)	8
1.2	Le principe de fonctionnement d'un inverseur (BLI99)	10
1.3	Principe de fonctionnement d'un inverseur à portes pivotantes-les portes sont fermées -	11
1.4	Principe de fonctionnement d'un inverseur à portes pivotantes-les portes ouvertes-	11
1.5	Principe de fonctionnement d'un inverseur à cascades	12
1.6	Principe de fonctionnement d'un inverseur à obstacles	12
2.1	Dynamic Modeling Laboratory	16
2.2	Modélisation et simulation multi-ingénierie (DAS81)	16
2.3	Architecture de Dymola (DEM06)	17
2.4	Le mode Modélisation les différentes vues du Dymola	18
2.5	Structure de la bibliothèque dans Dymola	19
2.6	Le langage de modélisation Modelica	20
2.7	Conception d'un modèle élémentaire Modelica	21
2.8	Construction d'un assemblage de modèles élémentaires Modelica	22
2.9	Architecture matérielle usuelle d'un actionneur électromécanique	23
2.10	Exemple d'un actionneur électromécanique	24
2.11	EMA F-18 NASA	24
2.12	L'actionneur du démonstrateur PRESAGE	24
2.13	Conduite du système d'actionnement	25
2.14	Architecture matérielle usuelle d'un actionneur électromécanique	25
2.15	Le bloc diagramme de l'actionneur	26
2.16	Transmission de puissance par le réducteur de vitesse	27
2.17	Vis à billes à recirculation interne	28
2.18	Effort aérodynamique appliqué sur l'inverseur de poussée	28
2.19	Diagramme du modèle du système de chargement	29

2.20	Système d'actionnement développé sous Dymola	31
2.21	Comparaison entre l'allongement du tube du modèle : (a) AMESim et (b) Dymola	31
2.22	Comparaison entre l'accélération du tube du modèle : (a) AMESim et (b) Dymola	32
2.23	Comparaison entre la charge aérodynamique du modèle : (a) AMESim et (b) Dymola	32
2.24	Système de chargement	33
2.25	Comparaison entre le déplacement de la structure mécanique du modèle : (a) AMESim et (b) Dymola	34
3.1	La réponse du système en BO du système : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique	44
3.2	Allongement du tube L_{tube} en BO : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique	45
3.3	La vitesse d'allongement du tube L_{tube} en BO : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique	45
3.4	Schéma de commande	47
3.5	Le problème standard de la commande	48
3.6	Le schéma de synthèse à deux blocs	49
3.7	Le schéma de commande H_∞ Loopshaping	51
3.8	Allures de la charge aérodynamique	54
3.9	Réponse du système en BO : (a) [Overstow - Flush] (b) [Flush - Deployed] (c) [Deployed - Full reverse]	55
3.10	Réponse du système en BF : (a) [Overstow - Flush] (b) [Flush - Deployed] (c) [Deployed - Full reverse]	56
3.11	La forme de la perturbation	57
3.12	Système avec erreurs de modélisation additives	57
3.13	Réponse du système en BF avec perturbation phase (a) [Flush - Deployed] (b) [Flush - Deployed] avec zoom sur l'intervalle [1.68 : 1.76]s	58
3.14	Réponse du système en BF avec perturbation phase [Overstow : Flush]	58
3.15	La forme de la perturbation dans le 2ème essai	59
3.16	Système avec erreurs de modélisation additives	60
3.17	Système avec erreurs de modélisation additives et une perturbation en sortie	60
3.18	Réponse du système en BF [Overstow - Flush] en présence : (a) d'erreurs de modélisation (b) d'erreurs de modélisation et d'une perturbation	60

3.19	Diagrammes de bode des fonctions de sensibilités et sensibilité complémentaire :	
	(a) [Overstow - Flush] (b) [Flush - Deployed] (c) [Deployed - Full reverse]	61
4.1	Diagramme bloc du système (4.3)	67
4.2	Diagramme du système équivalent (4.4)	67
4.3	Backstepping de $-\varphi(x)$ à travers l'intégrateur	68
4.4	Vitesse référence du moteur	71
4.5	Evolution des l'états :((a) x_1 et x_{1d} (b) x_2 et x_{2d} (c) x_3 et x_{3d} (d) x_4 et x_{4d} .	75
4.6	Évolution du signal de commande	76
4.7	Poursuite de vitesse de référence du moteur	76
4.8	La vitesse de la porte de l'inverseur de poussée	76
4.9	La vitesse de l'allongement du tube	77
4.10	Réponse du système en présence d'incertitudes	77
4.11	Réponse du système en présence d'incertitudes à $t = 0.4$	77
4.12	les incertitudes sur les fonctions non linéaires f_1 et f_3	78
A.1	Principe d'héritage entre des classes de Modelica	92
A.2	Exemple d'héritage entre des classes de Modelica	93
A.3	Classe FluidPort	95
A.4	Classe Flange	95
A.5	Les différentes étapes de traduction du code Modelica	97

Liste des tableaux

3.1	valeurs numériques paramètres du système de conduite	41
C.1	Expressions mathématiques des non-linéarités	101
C.2	Valeurs numériques des paramètres des non linéarités	101

Nomenclature

Notation	Dénomination	Unité
l_{tube}	Allongement du tube	m
BL	Bras de Levier	
n	Bruit de mesure	
K_{em}	Coefficient de force contre-électromotrice	$N.m.A^{-1}$
PV	Constante du système vis-écrou	$m.rad^{-1}$
C_2	Couple appliqué à l'arbre de sortie du réducteur	$N.m$
C_1	Couple appliqué à l'arbre d'entrée du réducteur	$N.m$
x_{tube}	Déplacement du bras de levier	m
e	Erreur entre la référence et la sortie du système	
ϕ_{La}	Flux magnétique à travers le bobinage du moteur	T
$S(s)$	Fonction de sensibilité	
$T(s)$	Fonction de sensibilité- inverse	
F_{vis}	Frottement de la vis	$N.m.s.rad^{-1}$
δ_{f_i}	Incertitudes des fonctions non linéaires	
r	Indice de réduction	
L_a	Inductance statorique	H
J_c	Inertie de la charge	$kg.m^{-2}$
J_{vis}	Inertie de la vis	$kg.m^{-2}$
J_a	Inertie des parties tournantes de l'actionneur	$kg.m^{-2}$
Σ	Matrice de covariance des données	
X_d	Matrice de données représentant le fonctionnement normal du système	
Ξ	Matrice diagonale contenant les valeurs propres de Σ	
$h_{J_{porte}}$	Moment cinétique de la porte	$kg.m^2.s^{-1}$

X_{cr}	Matrice des données centrées représentant le fonctionnement normal	
$h_{J_{vis}}$	Moment cinétique de la vis	$Kg.m^2.s^{-1}$
h_{J_m}	Moment cinétique de l'arbre	$Kg.m^2.s^{-1}$
N	Nombre de mesures effectuées	
m	Nombre de variables mesurées	
l	Nombre des composantes principales	
ψ_{MR}	Non linéarité du système de conduite -vis-	
ψ_{tube}	Non linéarité du système de conduit -tube-	
K_{arbre}	Raideur de l'arbre	$N.m.rad^{-1}$
K_{tube}	Raideur du tube	$N.m^{-1}$
ν	Rendement du réducteur	
R_a	Résistance statorique	Ω
u	Signal de commande	
d	Signal de perturbation en sortie	
r	Signal de référence	
$x(k)$	Vecteur de mesure à l'instant k	
ω_2	Vitesse de l'arbre de sortie du réducteur	$rad.s^{-1}$
ω_1	Vitesse de l'arbre d'entrée du réducteur	$rad.s^{-1}$
ω_m	Vitesse du moteur	$rad.s^{-1}$

ACRONYMES

ACP	Analyse en composantes principales
EMA	ElectroMechanical Actuator'
PBW	Power-By-Wire
Dymola	Dynamic Modeling Laboratory
ETRAS	Electrical Thrust Reverser Actuation System
EDP	Equation aux Dérivées Partielles
EDA	Equation Différentielle et Algébrique
EDO	Equation Différentielle Ordinaire
HIL	Hardware-In-TheLoop
SPE	Squared Prediction Error

INDICES

c	Charge
aero	Charge aérodynamique
d	désiré
r	Espace résiduelle
p	Espace principale
m	Moteur
cr	Variable centrée

Introduction Générale

L'aéronautique est un domaine qui connaît et connaîtra un essor encore plus important dans les dix prochaines années .A titre d'exemple, la récente commande de la compagnie indonésienne Lion Air d'A320 Néo s'élevant à plus de 18 milliards d'euros est la preuve que ce secteur créera encore plus d'emploi à l'avenir. Par ailleurs, le monde dans lequel nous vivons aujourd'hui est en pleine mutation, la crise fait encore (et toujours) des dégâts dans pratiquement tous les secteurs d'activités. Ainsi, une entreprise aujourd'hui a besoin d'un excellent pôle R&D, qui sera à même d'assurer une innovation et une amélioration de ses performances au quotidien. Ce n'est qu'ainsi que cette entreprise assurera sa pérennité.

Aussi, c'est dans ce contexte socio-économique que s'est articulé notre projet de fin d'études non pas en faisant abstraction du monde qui nous entoure mais plutôt en jetant un regard attentif à ce dernier.

Notre projet s'inscrit comme une partie du vaste projet de recherche appelé PRESAGE “ Plateforme Réelle Et Simulée d'Actionnement Générique et Adaptive”, ce projet labélisé par le pôle de compétitivité Mov'eo et financé par deux organismes français, le ministère de la Défense et la DGA (Direction Générale de l'Armement). A travers son pôle R&D basé à Alger, l'entreprise CERTIA “ Centre d'Études et de Recherche pour les Techniques Industrielles Appliquées ” en collaboration avec notre encadrement pédagogique au département d'Automatique de l'ENP, nous a donné tous les moyens afin d'apporter nos connaissances acquises durant ces années de durs labeurs au service du développement de la R&D en Algérie à travers PRESAGE. L'objectif principal de ce projet est la virtualisation des processus de validation et de vérification en développant une plateforme d'essai virtuelle permettant par modélisation et simulation d'établir les tests désirés. Cette plateforme a pour but d'apporter une aide à la conception, la formation ainsi qu'à l'utilisation future des moyens d'essais.

PRESAGE doit aussi permettre des gains tant en termes d'optimisation qu'en terme de définition des essais (que doit-on faire avec le moyen d'essai, quelle sont les fonctions

nécessaires à tester, quelles sont les approches les plus optimisées...etc).

Les principales retombées scientifiques attendues sont de proposer une approche générique de construction de modèles pour les systèmes complexes, le développement d'une plateforme de co-simulation permettant de gérer des modèles de forme et de structure différente ou encore d'automatiser les procédures de tests sur bancs d'essais virtuels pour tout type de champ applicatif.

Du point de vue économique, PRESAGE vise à réduire considérablement les cycles et les coûts de développement des bancs d'essais d'intégration, rendre possible la simulation en cas de pannes sans avoir à endommager ou détruire des équipements réels et ainsi contribuer à la sécurisation des processus de développement.

Ce travail est en parfaite concordance avec la tendance " plus électrique " dans le domaine aéronautique. A fortiori, l'avion de ligne des années 2015-2025 verra un accroissement de ses fonctions électriques. Les Boeing 787 et l'Airbus A380 ouvrent la voie, les futurs monocouloirs moyen-courriers leur emboîtent le pas (A350 ou encore le prochain Comac C919 chinois).

Trois raisons expliquent cet engouement : d'abord l'objectif permanent de réduire la masse globale de l'avion et donc sa consommation en carburant. En effet, l'utilisation de systèmes électriques à la place de circuits hydrauliques donne de la flexibilité en termes de conception. Cela permet ainsi d'optimiser les architectures des systèmes de l'avion, et d'aboutir finalement à des gains de masse. Ensuite, cette même flexibilité de conception offre un important potentiel de customisation des systèmes. Enfin, la maintenance promet d'être plus efficace car les circuits électriques peuvent être contrôlés en temps réel et leur vérification est plus facile que la recherche d'une fuite hydraulique. Pour autant, pas questions pour les ténors du domaine de " remplacer " du jour au lendemain la canalisation et les pompes hydrauliques par des câbles et des moteurs électriques : le règne annoncé de l'énergie électrique est une véritable révolution qui engage l'architecture même des aéronefs pour obtenir des gains substantiels. Et parce que la conception des avions est un processus de longue haleine, les acteurs de la filière ont obligation de travailler ensemble dès aujourd'hui pour préparer l'avion du futur.

Le but de notre travail qui s'articule en trois parties est le suivant : apporter des améliorations et une plus-value au travail déjà réalisé sur la modélisation et la commande d'un système aéronautique particulier, l'inverseur de poussée. Ce dernier et dans le cas test PRESAGE, a été développé par la société AIRCELLE, filière du groupe SAFRAN.

En premier lieu, nous allons donner dans la partie modélisation une grande importance à la définition du système en question non sans expliquer son rôle prépondérant dans un aéronef.

En deuxième partie, nous allons exposer deux solutions que nous préconisons en termes de choix de techniques de commande robuste pour le système d'inversion de poussée, tout en comparant avec la commande réalisée en amont et basé sur un PID simple.

En troisième lieu, nous nous sommes attelés à étudier un problème récurrent et d'autant plus important dans les systèmes aéronautique, le diagnostic de défauts.

Nous présentons ci-dessous l'organisation du notre mémoire :

Chapitre 1 :

Le premier chapitre est consacré à la présentation du système d'inverseur de poussée dans son environnement avion. Nous décrivons son principe de fonctionnement ainsi les intérêts technico-commerciaux et ses limitations d'emploi.

Chapitre 2 :

Le second chapitre comporte deux parties .La première partie est consacrée à une brève présentation du progiciel Dymola .La seconde partie est dédiée à une étude architecturale des principaux éléments du notre système afin d'aboutir à la modélisation du système d'inversion de poussée.

Chapitre 3 :

Dans le troisième chapitre nous allons donner le modèle d'état utilisé, sur lequel nous avons appliqué la commande par H_∞ Loopshaping. Par la suite, nous allons exposer les résultats de simulation correspondant en développant la démarche pour l'application de cette technique de commande sur le système d'inversion de poussée dans le cas test « PRESAGE ».

Chapitre 4 :

Dans ce chapitre , nous avons tenté l'application d'une commande robuste ,le Backstepping par poursuite de trajectoire référence .Nous avons bien sur développé en amont le fondement théorique de la technique pour après donner les résultats de simulation.

Première partie

Modélisation

Introduction

L'histoire nous a enseigné que chaque civilisation empreinte un chemin ou une courbe en forme de cloche symbolisant son apogée et sa déclinaison après avoir atteint un point culminant. La recherche comme n'importe quel civilisation connaît des moments de grande découverte et de réussite, a contrario, elle peut connaître des instants plus durs et qui demandera aux acteurs (ingénieurs, doctorants, industriels ... etc) des efforts pour pouvoir atteindre leurs objectifs respectifs. Une des clefs de voute permettant d'atteindre une synergie entre les différents objectifs visés par n'importe quel projet scientifique (dans notre cas la commande du système d'inversion de poussée cas test " PRESAGE ") est une bonne modélisation. A fortiori, la conception et la réalisation d'un système de commande n'est pas chose aisée. C'est la raison pour laquelle la partie modélisation représente un pion central dans notre travail et ce pour assurer des résultats qui seront à mêmes de répondre au cahier de charge imposé. Ce dernier est développé en préambule de la partie commande.

En effet, nous allons introduire en premier lieu le système d'inversion de poussée, puis nous développerons et expliquerons nos motivations pour l'utilisation de certains outils de modélisation (AMESim et Dymola). Par la suite, nous donnerons une description "physique" du système en question avant de développer ses différents sous-systèmes sous la plateforme Dymola.

In fine, nous donnerons une représentation d'état du système et comment nous nous sommes attelés à la modifier structurellement à des fins de commande .Ce dernier reste un point prépondérant dans notre travail car la modélisation doit être toujours liée à un objectif de commande, et ce pour faciliter la synthèse de la loi de commande. En l'occurrence l'application à la commande par H_∞ Loopshaping et par poursuite de trajectoire référence par Backstepping.

CHAPITRE 1

Le système d'inversion de poussée

1.1 Introduction

L'atterrissage et le décollage sont les phases les plus délicates du pilotage d'un avion. A l'atterrissage, il est nécessaire de stopper l'avion sur des distances raisonnables, même si la piste est verglacée ou que le train d'atterrissage soit défaillant. Un des dispositifs utilisés est l'inverseur de poussée.

La partie du projet PRESAGE étudiée est la modélisation et la commande d'une nouvelle génération d'inverseurs de poussée développée par d'AIRCELLE, équipementier de l'aéronautique fabricant de nacelles et se place dans cette activité au deuxième rang mondial. Le système d'actionnement électrique de l'inverseur de poussée s'inspire fortement de son équivalent hydraulique. Parmi les ambitions du travail qui est réalisé, il y a celle d'optimiser le système en effectuant une réelle rupture technologique par rapport à l'hydraulique et en optimisant la consommation énergétique, le coût d'exploitation et la masse.

Ce premier chapitre présente une description détaillée du système d'inverseur de poussée et sa tâche au niveau de l'avion. Au départ, nous décrirons son principe de fonctionnement ainsi son intérêt technico-commercial. Ensuite, nous parlerons des différents types d'inverseurs de poussée. Enfin, nous terminerons ce chapitre par la limitation d'emploi du système en question.

1.2 Description du système d'inverseur de poussée et sa tâche au niveau de l'avion

L'inverseur de poussée est le sous-ensemble le plus important de la nacelle d'avion en masse et en coût. Il joue un rôle essentiel dans l'atterrissage de l'avion. Dans cette phase, lorsque l'avion est en phase de roulage, l'inverseur obstrue la tuyère d'éjection des gaz afin de diriger le jet du moteur vers l'avant de la nacelle. On crée une force de freinage appelée contre-poussée, qui contribue au ralentissement de l'avion. Voici un schéma présentant toutes les parties d'une nacelle.

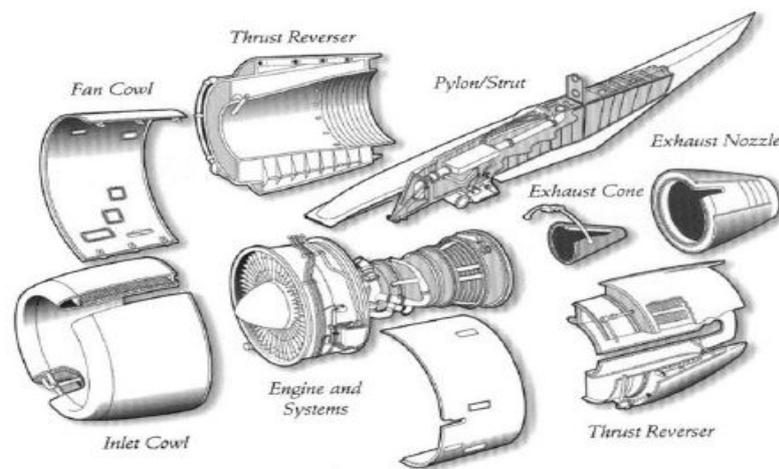


FIGURE 1.1 – Vue éclatée d'une nacelle d'avion (BEN11)

1.3 Intérêt technico-commercial

La recherche permanente de l'augmentation de la charge embarquée payante et du rayon d'action des avions, permise par les progrès de l'aérodynamique et de la motorisation, se fait au détriment de la masse des avions et de leur vitesse d'atterrissage.

En outre, l'existence d'une poussée résiduelle des turbomachines au ralenti et le long délai de réponse à la sollicitation du pilote (délai d'autant plus gênant que l'avion est plus rapide), compliquent ce problème du freinage.

Les freins des avions modernes sont très efficaces, mais sur piste mouillée, glacée ou recouverte de neige, cette efficacité peut être réduite par la perte d'adhérence des pneus de

l'avion sur la piste. Lors de manœuvres très délicates comme celle du décollage interrompu, l'assistance d'un système complémentaire aux freins de roue peut être salutaire.

C'est pourquoi l'utilisation de l'inverseur de poussée a été très largement acceptée que ce soit sur les avions de transport civils ou les avions de combat. De plus, l'une des préoccupations majeures des compagnies aériennes est de réduire les coûts de fonctionnement. L'inverseur répond à cette attente en diminuant la distance d'atterrissage. Cela entraîne une économie non négligeable pour les compagnies aériennes qui peuvent donc réduire certains frais.

Le freinage étant assuré par l'effort des freins sur les roues et par l'augmentation de traînée due aux volets hypersustentateurs, on diminue ainsi l'usure des pneumatiques et des freins. On diminue aussi, comme la distance d'atterrissage est réduite, le temps d'occupation de la piste, ce qui permet d'économiser du carburant et aussi de libérer la piste plus rapidement, ce qui, vu l'encombrement actuel des aéroports, est loin d'être négligeable. Grâce à l'inverseur de poussée, on peut estimer le gain de distance d'atterrissage à environ 25% sur piste sèche et jusqu'à 50% sur piste verglacée ou mouillée.

1.4 Principe de fonctionnement

Le principe consiste à chercher l'énergie nécessaire au freinage d'un avion à la source même de l'énergie cinétique de l'appareil, c'est à dire au sein du jet de la tuyère à réaction. Après l'atterrissage, lorsque l'avion est en phase de roulage, un obstacle solide l'**inverseur** est mis en travers du flux du moteur pour dévier voir retourner le jet propulsif vers l'avant. Cet inverseur produit une force vers l'arrière que l'on appelle contre-poussée et qui vient compléter efficacement le freinage classique par les trains principaux. Le principe de fonctionnement d'un inverseur est présenté sur la figure 1.2.

En vol normal, l'inverseur canalise les flux en minimisant les traînées internes, pour ne pas détériorer les performances aérodynamiques du moteur. La distance de freinage d'un avion équipé d'un inverseur est typiquement autour de 70% de celle sans inverseur, mais il devient indispensable sur piste humide ou verglacée puisqu'il limite la distance d'atterrissage dans des proportions pouvant atteindre 50%.(BLI99)

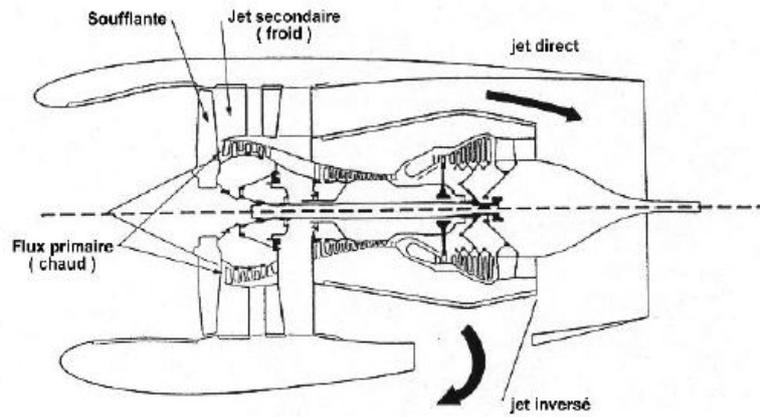


FIGURE 1.2 – Le principe de fonctionnement d'un inverseur (BLI99)

1.5 Les différents types d'inverseurs

Les inverseurs de poussée sont classés en trois catégories principales :

1. Inverseurs à portes pivotantes.
2. Inverseurs à cascades.
3. Inverseurs à portes pivotantes.

Le choix dépend essentiellement du cas d'application. Chaque type peut présenter des avantages pour la réalisation d'exigences particulières en fonction :

- De la fonction de l'avion.
- Du domaine de vol.
- De la position et du type de nacelle, pour lequel il existe 96 combinaisons et configurations.
- D'autres paramètres géométriques.
- Des coûts d'achat et de maintenance ainsi que de la masse.

1.5.1 Inverseurs à portes pivotantes.

Des portes sont découpées dans le capot de la tuyère. En mode direct, ces portes sont fermées et donnent à la tuyère une forme identique à celle qu'elle aurait sans inverseur. Une fois les portes ouvertes, leurs parties inférieures bloquent le canal secondaire obligeant le flux secondaire à sortir par l'ouverture ainsi créée. Leurs parties supérieures aident à dévier le jet et jouent également un peu un rôle d'aérofreins. Dans ce type d'inverseurs, c'est le flux secondaire qui donne toute la contre-poussée, tout en devant annuler la poussée du flux

primaire non inversé. Bien que le contrôle de la nappe soit plus difficile, cet inverseur a le mérite d'être plus simple : sa masse et son coût de maintenance sont donc assez faible .

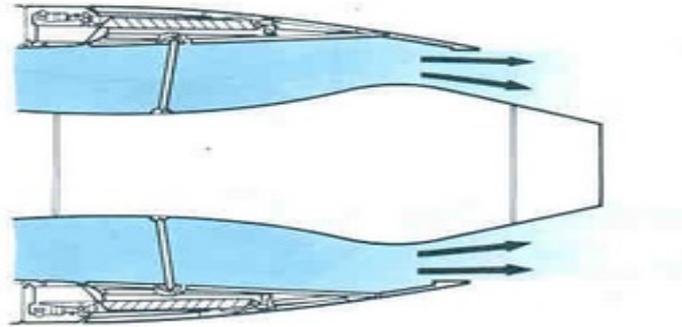


FIGURE 1.3 – Principe de fonctionnement d'un inverseur à portes pivotantes-les portes sont fermées -

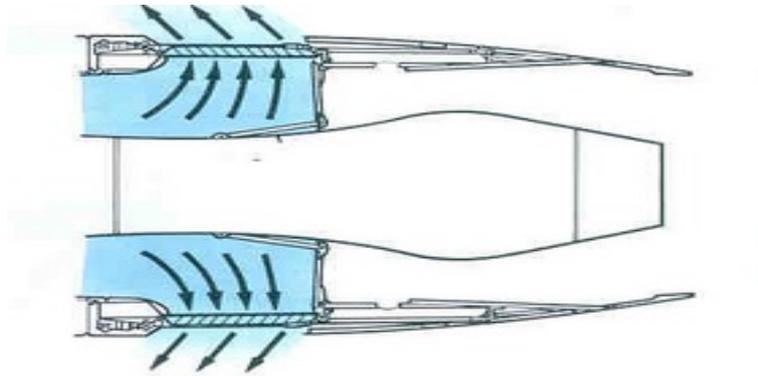


FIGURE 1.4 – Principe de fonctionnement d'un inverseur à portes pivotantes-les portes ouvertes-

1.5.2 Inverseurs à cascades.

La solution classique d'inverseur de poussée sur des turboréacteurs consiste en l'utilisation d'inverseurs à cascades. Ceux-ci offrent des avantages dans les cas d'installation à fortes contraintes géométriques.

Les inverseurs à cascades comportent une série de panneaux mobiles qui obturent la veine entre le fan et la sortie pendant que simultanément une partie de la structure externe recule axialement créant une ouverture d'écoulement pour le flux inversé.

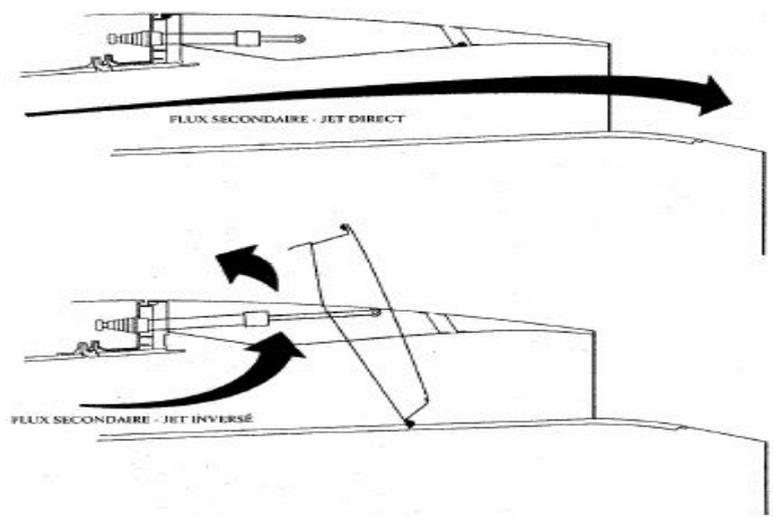


FIGURE 1.5 – Principe de fonctionnement d'un inverseur à cascades

1.5.3 Inverseurs à obstacles.

L'inversion de poussée sur les petits turboréacteurs, poussée inférieure à 20 – 25.000 livres, avec un faible taux de dilution est généralement réalisée grâce à l'ouverture de 2 portes en aval qui bloquent et dirigent le flux du moteur vers l'avant. En position fermée, ces portes constituent une partie essentielle de la structure de la nacelle. Une variante de ce type d'inverseur a été développée et brevetée par Hurel-Dubois¹. Elle permet une performance accrue, des avantages en croisière par réduction de la traînée, une meilleure géométrie, une remarquable résistance structurale et une parfaite sortie tuyère en terme de perte de charge.

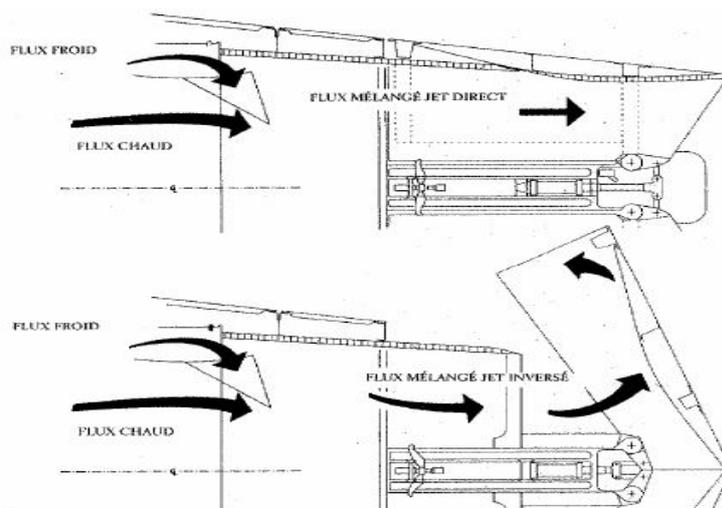


FIGURE 1.6 – Principe de fonctionnement d'un inverseur à obstacles

1. Une société aéronautique française, de Construction des Avions créée en 1947

Dans le cas test "PRESAGE", le type d'inverseur de poussée étudié est un inverseur à obstacles.

1.6 Limitation d'emploi de l'inverseur de poussée

L'utilisation de l'inverseur de poussée est limitée par différents problèmes :

- Les charges sur la voilure pour les moteurs installés sous ailes : En effet, on note à bas régime une augmentation de traînée proportionnelle à la surface interceptée par les obstacles déployés dans le vent (portes). En terme de portance, un inverseur de poussée dégrade l'écoulement autour de la voilure et entraîne logiquement une certaine baisse de la portance.
- Interaction avec les mesures anémométriques : Les prises de pression anémométriques, qui donnent la vitesse de l'avion, se situent sur la pointe avant du fuselage. Les nappes déviées par l'inverseur viennent perturber le champ de pression autour de ces prises, entraînant ainsi une fausse lecture de la vitesse.
- Réingestion : Il s'agit de la recirculation des gaz éjectés par l'inverseur dans l'entrée d'air. Elle est due principalement à une réflexion du jet inversé sur un élément extérieur : par exemple, sur le BR 710 situé à l'arrière du fuselage, le risque de réingestion vient de la réflexion du jet sur la voilure. La limite de réingestion résulte alors d'un compromis entre l'efficacité de cet inverseur imposant un régime minimal moteur et la durée d'utilisation.
- Ouverture intempestive en vol : le principal risque associé à un inverseur de poussée réside dans ce problème. C'est pourquoi sont installés sur chaque porte trois verrous. Ce risque est traditionnellement étudié en vol et fait l'objet d'un système de sécurité.

1.7 Exigences de l'inverseur de poussée

La configuration finale de l'inverseur de poussée doit satisfaire aux quatre objectifs suivants :

- Assurer la compatibilité en débit avec le moteur lors des phases de déploiement du volet et en mode d'inversion complète.
- Garantir la contre-poussée spécifiée par le motoriste.
- Orienter les jets inversés afin d'interdire la réingestion entre moteurs et les interactions avec toute partie de l'avion (empennage, fuselage, aile, volet,...etc).
- Répondre aux exigences de sécurité en cas d'utilisation inhabituelle (décollage avorté, déploiement intempestif).

1.8 Conclusion

Dans ce premier chapitre, après avoir mis l'accent sur le double intérêt technico-commercial de l'inverseur de poussée. Nous avons décrits les différents types d'inverseurs de poussée présents sur les aéronefs tout en soulignant les contraintes ou limitations d'emploi de ce composant majeur de la nacelle d'avion. Nous pourrions ajouter en conclusion de ce qui a été développé que compte tenu de son rôle majeur dans un avion, la fiabilité de l'inverseur de poussée et de la nacelle plus généralement, doit être optimisée au maximum et ce pour éviter des catastrophes. En effet, il y'a eu pas mal de crash d'avions dont la cause était une rupture dans la nacelle.

Conception d'une plateforme virtuelle du système d'inversion de poussée

2.1 Outils de modélisation et de simulation dynamique Dymola/Modelica

Les nombreuses avancées technologiques dont ont bénéficié les logiciels de simulation dynamique, ont grandement bénéficié aux acteurs de la modélisation et de la simulation dynamique des processus multiphysique. En effet, l'utilisation de logiciels permettant une orientation axée objets et non blocs a permis une grande flexibilité d'utilisation. Ainsi, des logiciels tels que LMS AMESim ou Dymola/Modelica permettent de se baser sur un socle ou une base de données suffisamment puissante et efficace qui sera à même d'une part d'assurer des performances mais aussi d'améliorer ces dernières en minimisant les erreurs de modélisation.

Dans notre projet, l'utilisation du progiciel Dymola/Modelica répond à une volonté forte des industriels qui donnent encore plus d'importance à l'aspect modélisation. A fortiori, le langage de modélisation Modelica est sans contexte la référence dans le domaine aéronautique.

Nous présenterons dans ce qui suit une brève description du progiciel Dymola/Modelica en donnant les grandes lignes qui nous ont permis de modéliser correctement un système multiphysique sur ce type de plateforme.

Dans le cas test PRESAGE, la porte mobile est soumise à deux sources d'efforts l'actionneur électrique qui sert à l'ouvrir "Système d'actionnement" et les efforts aérodynamiques "Système de chargement". Pour cela, une étude architecturale des principaux éléments du

système d'actionnement et de charge est conduite dans la deuxième partie de ce chapitre afin d'aboutir un modèle physique qui représente le mieux le système réel. Enfin, nous présenterons les différents sous modèles de l'inverseur de poussée développés sous l'environnement Dymola.

2.1.1 Environnement de modélisation et simulation dynamique Dymola

Un des outils complets pour la modélisation et la simulation de systèmes complexes est Dymola¹ utilisé en automobile (EJJ10) (HAM12), aéronautique, robotique et autres applications.

Les fonctionnalités uniques de Dymola sont à la base de solutions révolutionnaires pour la modélisation et la simulation, capables de simuler le comportement dynamique et les interactions complexes entre les systèmes de multiples domaines d'ingénierie : mécanique, électrique, thermodynamique, pneumatique, hydraulique et systèmes de commande. Les utilisateurs sont ainsi à même de construire des modèles plus intégrés et obtiennent des résultats de simulations plus réalistes (DAS81).



FIGURE 2.1 – Dynamic Modeling Laboratory

Le logiciel Dymola est notamment utilisé collectivement par les constructeurs automobiles allemands DaimlerChrysler (DYN05), BMW, Audi et Volkswagen pour la modélisation et la simulation de systèmes d'air conditionné. Toyota qui utilise également Dymola (DAS07) au sein de ses divisions moteur, boîte de vitesse et châssis, a créé une bibliothèque pour le contrôle de ses moteurs diesel.

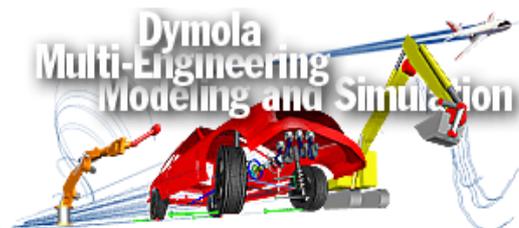


FIGURE 2.2 – Modélisation et simulation multi-ingénierie (DAS81)

1. Dynamic Modeling Laboratory

La flexibilité de Dymola en fait un outil polyvalent, parfait pour la modélisation de conceptions et technologies nouvelles ou en alternative. Aussi, Dymola offre la possibilité de réutilisation des modèles. En d'autres termes, il permet d'utiliser des composants dans des contextes différents, et de réemployer un modèle pour différentes études. De plus, il possède des interfaces avec d'autres logiciels (Simulink, Excel, Matlab) (ELH08). Une des fonctions les plus puissantes de Dymola est la simulation temps-réel. Cette fonction est utilisée pour les systèmes HIL où un composant matériel est remplacé par un composant logiciel sur des environnements tel que tel que dSPACE et xPC (HAD08). En termes de simulation 3D, Dymola n'est pas en reste. En effet, il permet d'importer des fichiers DXF ou STL et une animation en 3D en temps réel (PLA09).

Les objectifs de l'utilisation de Dymola dans le projet presage :

- Développer une bibliothèque des modules d'inverseur de poussée.
- Évaluer les capacités fonctionnelles du logiciel Dymola/Modelica pour les simulations des systèmes d'inversion de poussée.

2.1.1.1 Architecture de Dymola

Dymola est basé sur l'utilisation des modèles Modelica stockés dans des bibliothèques. Il peut aussi importer d'autres données et graphiques. Un traducteur contenu dans Dymola permet de générer du code C à partir des équations. Le code C généré peut être exporté à Simulink ou à des plateformes HIL² (HAD08).

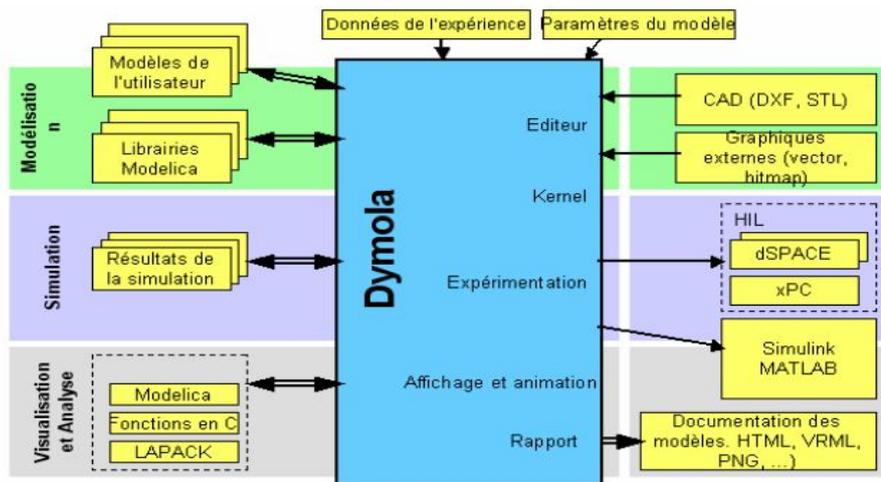


FIGURE 2.3 – Architecture de Dymola (DEM06)

2. Hardware-In-TheLoop est une méthode de simulation caractérisée par l'association de véritables composants, connectés à une partie temps-réel simulée

2.1.1.2 Interface graphique de Dymola

Dymola possède trois fenêtres différentes : L'écran principal, la fenêtre des librairies et la vue de commande. Il existe aussi deux types de modes : Le mode Modélisation est utilisé pour créer et assembler les différents composants formant mon modèle, en utilisant la technique du glisser-déposer. A contrario, il est aussi possible d'écrire le script directement et déclarer les variables par la même manière. Cependant, la première technique reste plus intuitive. Quant au deuxième mode, il s'agit du mode Simulation qui est utilisé pour paramétrer et visualiser les résultats de simulation (en 2D et/ou 3D), ce dernier mode est aussi utilisé pour optimiser les performances de calculs .

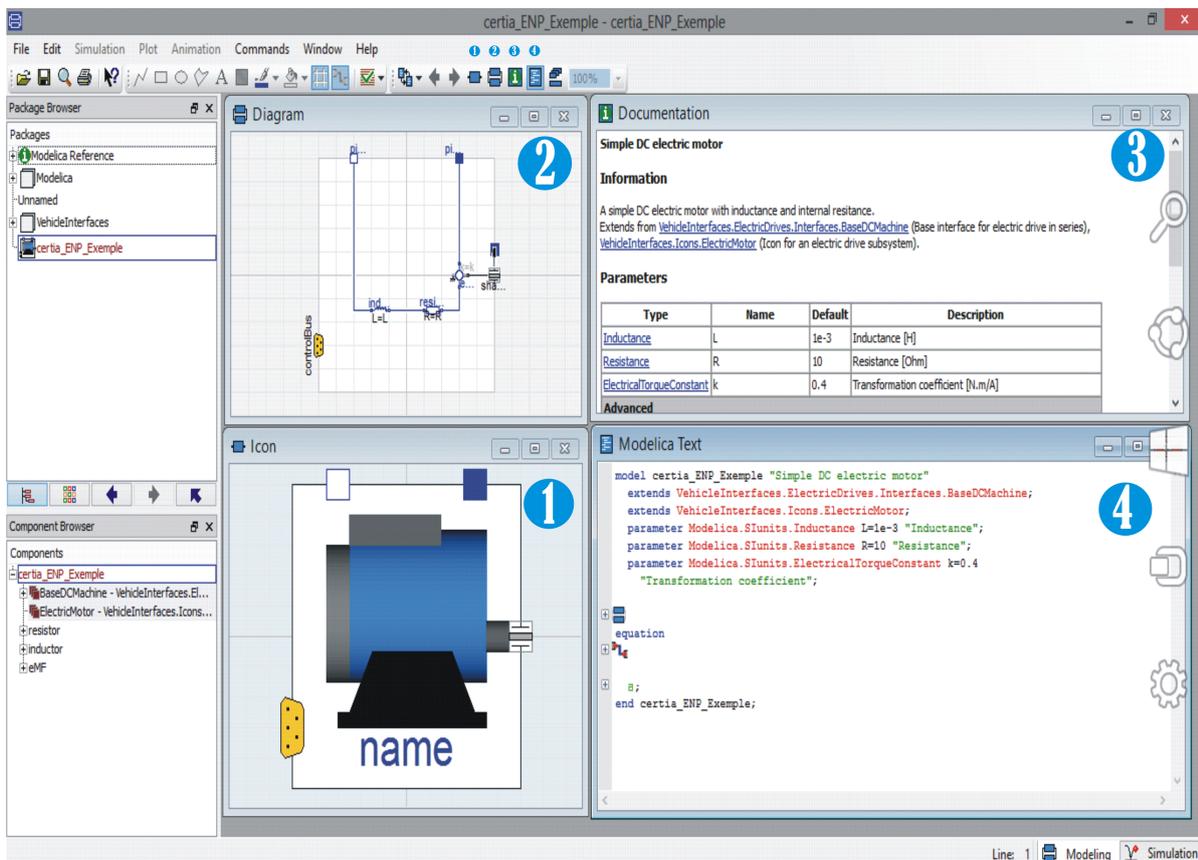


FIGURE 2.4 – Le mode Modélisation les différentes vues du Dymola

1- Vue Icône. 2- Vue Diagramme. 3- Documentation. 4- Vue script Modelica.

2.1.1.3 Bibliothèques de Dymola

L'organisation de la bibliothèque dans Dymola respecte une structure arborescente dont la racine se dénomme Modelica.

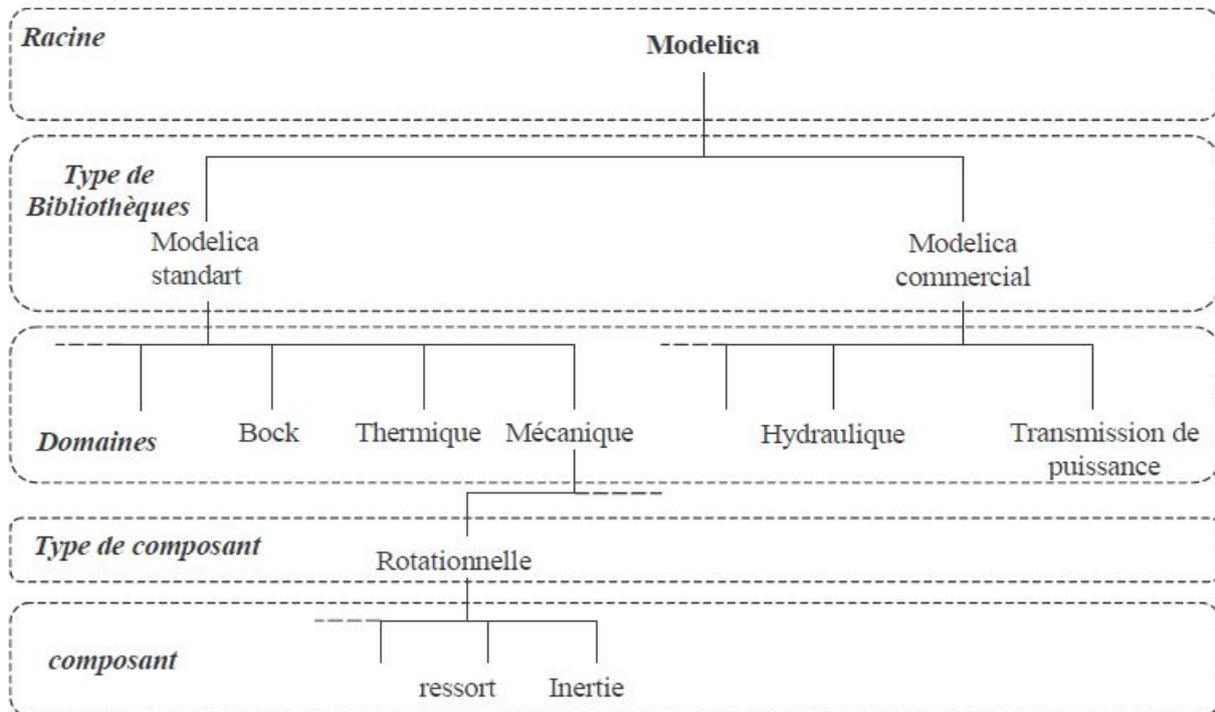


FIGURE 2.5 – Structure de la bibliothèque dans Dymola

La force de Dymola réside aussi dans la variété de ces bibliothèques. En effet, il suffit de jeter un coup d'œil sur le panel de composants qui existe pour comprendre qu'on peut parfaitement modéliser fidèlement notre système. Voici une liste non-exhaustive des nombreuses bibliothèques qui existent sous Dymola :

Blocks : Cette bibliothèque contient les éléments continus et discrets nécessaires à la création de fonction de transfert , filtres et sources de signal .

Complex Blocks : Elle permet d'avoir des signaux complexes, elle peut par exemple être utilisée pour la création de circuits électriques avec courants et tensions périodique.

StateGraph : Elle est utilisé essentiellement pour modéliser des systèmes à évènements discret .

Electrical : Elle contient les éléments électriques de base (Résistance, réactance ,interrupteurs, machines DC ... etc.)

Mechanics : Contient les éléments nécessaire à la création de systèmes mécanique ayant un comportement de translation et/ou de rotation .

Hydraulics : Dans cette librairie développé par Modelon (Développeur de solutions pour l'engineering) ,on peut trouver tous les éléments à caractère hydraulique .Exemple : Verin , servovalve, clapet anti-retour ... etc.

2.1.2 Le langage Modelica

Dymola est une interface graphique de modélisation et de simulation en langage de modélisation libre Modelica³. Il hérite ainsi donc toutes les propriétés de ce langage de simulation, ce qui signifie que les utilisateurs ont la possibilité de créer leurs propres bibliothèques de modèles et utiliser celle qui sont disponible afin de trouver celles qui s'adaptent le mieux à leurs besoins spécifiques en terme de conception, modélisation et simulation. Ce langage ainsi qu'une partie de ces outils sont présentés en détail dans la suite.



FIGURE 2.6 – Le langage de modélisation Modelica

2.1.2.1 Points forts de Modelica

Les logiciels d'utilisation générale se limitent pas à un domaine particulier, cependant ils imposent à leurs utilisateurs une perte de temps nécessaire à la traduction des modèles au formalisme de l'outils. La nouveauté apportée par Modelica est qu'il est un langage de modélisation permettant utilisation directe des équations tel que déduites des relations fondamentales (NFO06), ainsi que des objets et connexions. De ce fait il donne la possibilité aux modeleurs d'avoir une vision physique plutôt que mathématique de leur système. Les points forts de Modelica se résument en :

- Langage open source maintenu par un groupe de travail international (Association Modelica).
- Modélisation acausale.
- Modélisation par des équations EDA (Equation Différentielle et Algébrique) et EDO (Equation Différentielle Ordinaire), EDP (Equation aux Dérivées Partielles).
- Langage orienté objet (classe, instance, héritage).
- Élaboration de modèles à partir de composants (schématisation).

3. L'annex A présente une description détaillée du langage Modelica

2.1.2.2 Les points faibles de Modelica

L'utilisation du langage Modelica est une solution efficace pour la modélisation des systèmes complexes, cependant, ce langage présente quelques inconvénients :

- Langage compliqué.
- Langage loin du solveur et son implantation numérique : directives pour le solveur numérique pas claires et insuffisantes.

2.1.3 Exemple de conception d'un modèle Modelica sous Dymola

2.1.3.1 Construction d'un modèle élémentaire

La procédure de conception d'un modèle Modelica élémentaire (JAR10) :

1. Déclarer les types de variables mises en jeu à l'aide de classes type;
2. Déclarer les paramètres mis en jeu (i.e. nommer les paramètres et leur donner une valeur numérique);
3. Déclarer les variables mises en jeu (i.e. nommer les variables, préciser leur type et, si besoin, leur donner une valeur initiale grâce à l'attribut start);
4. Écrire les équations du modèle.

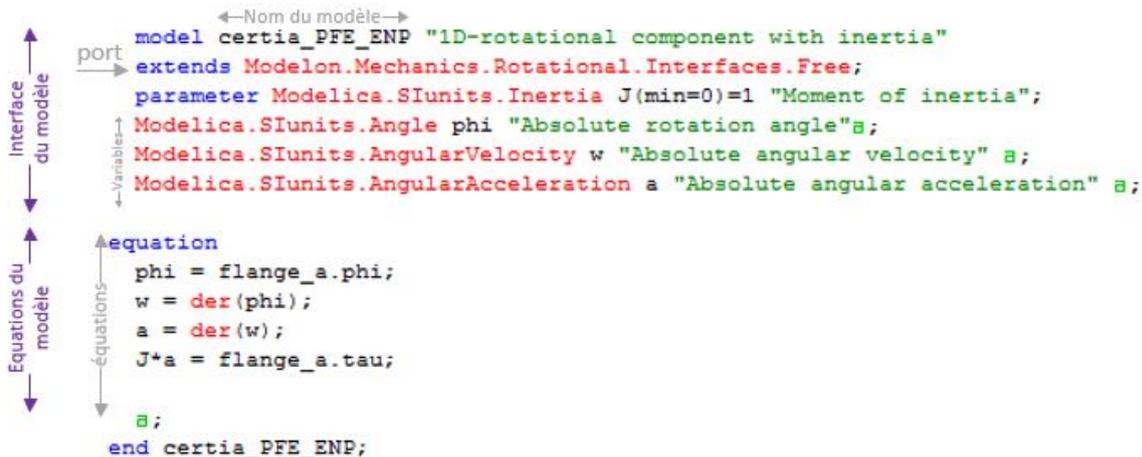


FIGURE 2.7 – Conception d'un modèle élémentaire Modelica

2.1.3.2 Construction d'un assemblage de modèles

La procédure de conception d'un assemblage de modèles élémentaires Modelica (JAR10) :

1. Déclarer les types de variables mises en jeu à l'aide de classes type;
2. Définir les modèles de connecteurs à l'aide de classes connector;

3. Définir des modèles génériques partiels à l'aide de classes partial model. Grâce au mécanisme d'héritage, ces modèles serviront de base pour la construction des modèles élémentaires ;
4. Construire les modèles élémentaires (ou modèles de composant) ;
5. Écrire un modèle principal dédié à l'assemblage des modèles élémentaires ;
6. Instancier les différents modèles élémentaires qui ont été préalablement construits ;
7. Assembler les instances de modèles à l'aide d'équations de type connect.

```

model certia_enp_pfe
  Modelon.Mechanics.Rotational.Inertia inertia
  B;
  Modelon.Mechanics.Rotational.Fixed fixed
  B;
  Modelon.Mechanics.Rotational.Ending ending
  B;
equation
  connect(fixed.flange_a, inertia.flange_a) B;
  connect(ending.flange_a, inertia.flange_a) B;
  B;
end certia_enp_pfe;

```

FIGURE 2.8 – Construction d'un assemblage de modèles élémentaires Modelica

2.2 Modélisation du système d'inversion de poussée

L'objectif de cette partie consiste à modéliser les différents sous-systèmes de l'inverseur de poussée. En premier lieu, nous allons donner une description détaillée des composants des sous-systèmes en question (système d'actionnement, système de chargement). Une fois le système décrit, la suite de cette partie présente les modèles Dymola obtenus pour les différentes parties de l'inverseur de poussée du cas test « PRESAGE ». Un travail a été effectué au amont par l'équipe CERTIA sur la plateforme AMESim LMS/Mathworks⁴ à partir desquelles nous nous sommes fortement inspirés pour faire notre travail sous Dymola. Aussi, lors de la validation nous avons pris chaque sous-modèles AMESim pour les comparer avec nos modèles Dymola.

4. LMS Imagine.Lab AMESim (anciennement AMESim) est un logiciel de simulation pour la modélisation et l'analyse de systèmes 1D multi-domaines.

La figure ci-dessus montre les parties distinctes qui composent l'actionneur électrique.

Dans le cas test PRESAGE, l'EMA intègre un transformateur mécanique entre le moteur et la porte de l'inverseur de poussée. Ce transformateur contient une boîte de réduction et un système de conversion mécanique du mouvement, de la rotation vers la translation via la vis à bille. Les moteurs électriques utilisés travaillent, en général, à grande vitesse.

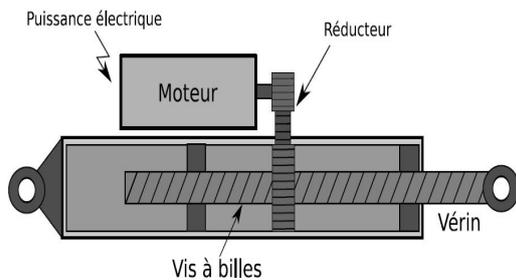


FIGURE 2.10 – Exemple d'un actionneur électromécanique



FIGURE 2.11 – EMA F-18 NASA

Le démonstrateur PRESAGE est relatif à l'intégration virtuelle d'un système innovant d'actionnement électromécanique d'un inverseur de poussée ETRAS⁸. L'inverseur est constitué de deux portes indépendantes (c'est-à-dire non synchronisées mécaniquement) qui sont articulées en liaison pivot sur la structure fixe du moteur comme le montre la figure ci-dessous 2.12. Chaque porte est mue par un vérin électromécanique linéaire dont la tige est en liaison cardan avec la porte et dont le corps est en liaison cardan avec la structure du moteur.

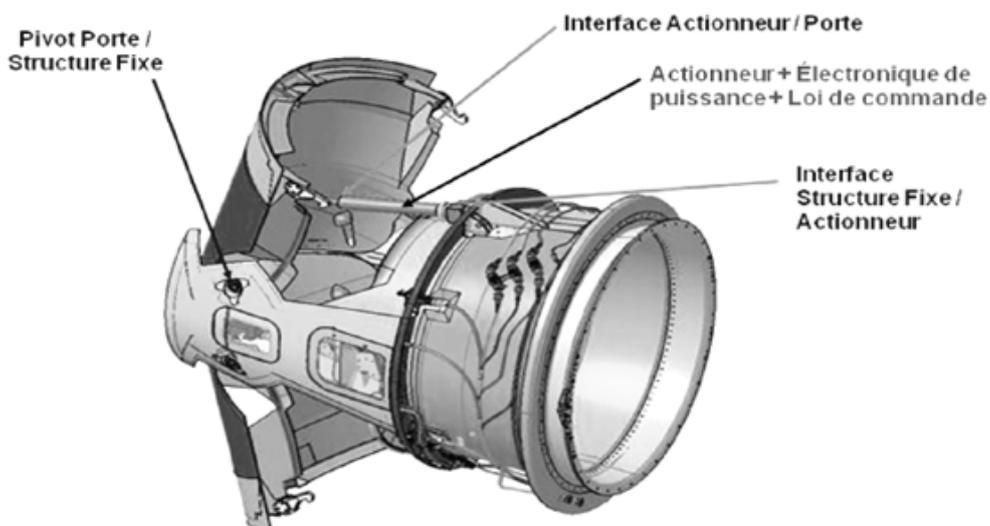


FIGURE 2.12 – L'actionneur du démonstrateur PRESAGE

8. Electrical Thrust Reverser Actuation System

Description du fonctionnement du système d'actionnement

L'action principale de l'actionneur est de positionner la porte de l'inverseur par rapport à la nacelle en développant les efforts nécessaires pour la charge aérodynamique.

L'actionneur d'inversion de poussée remplit la fonction de vérin linéaire. Le vérin source de puissance électrique PBW est composé de :

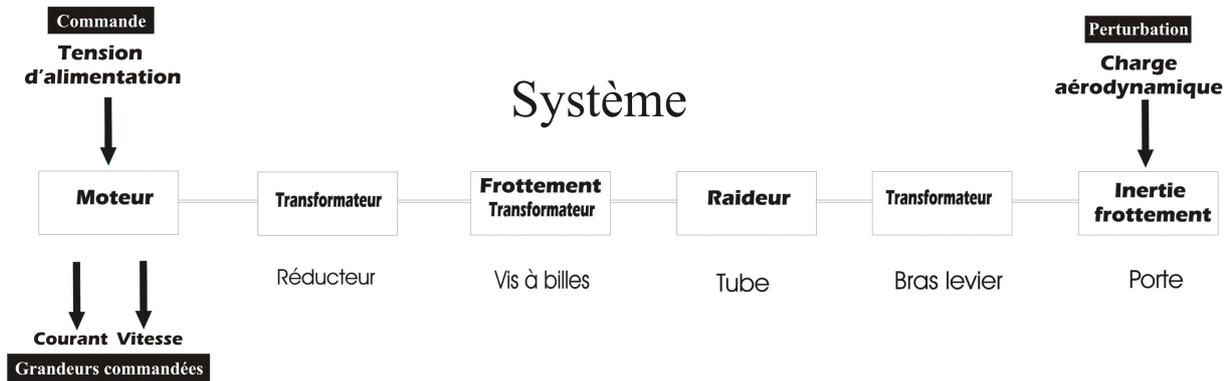


FIGURE 2.13 – Conduite du système d'actionnement

- Moteur électrique
- Frein a friction
- Système vis-écrou
- Arbre de transmission
- Tube mobile
- Réducteur

La figure ci-dessous, illustre les différentes parties de l'actionneur qui seront par la suite décrits.

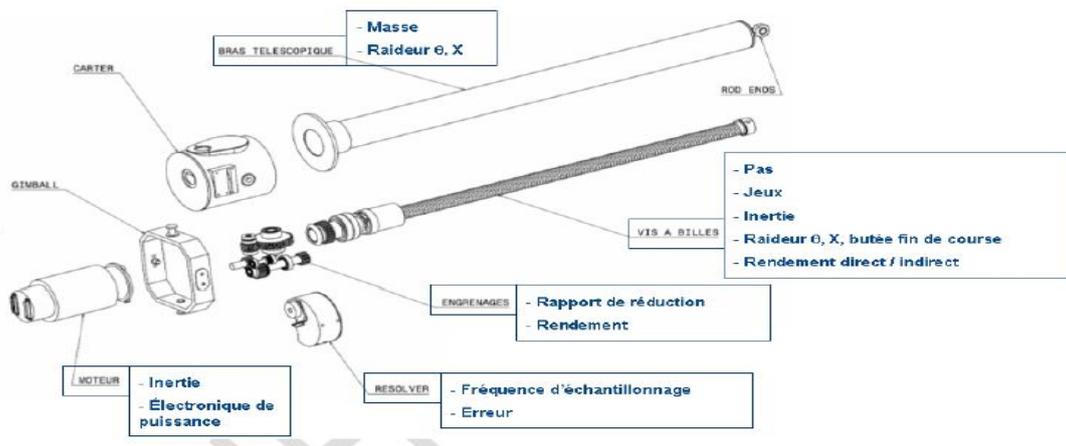


FIGURE 2.14 – Architecture matérielle usuelle d'un actionneur électromécanique

Un moteur fourni un couple moteur C pour entraîner en rotation à une vitesse ω_1 un arbre 1, muni d'un réducteur. Celui-ci entraîne, via un engrenage de rapport m un arbre

intermédiaire 2, qui grâce à un système vis-écrou, provoque la translation d'une tige en sortie. La figure 2.15 représente le bloc diagramme de l'actionneur, et illustre son principe de fonctionnement.

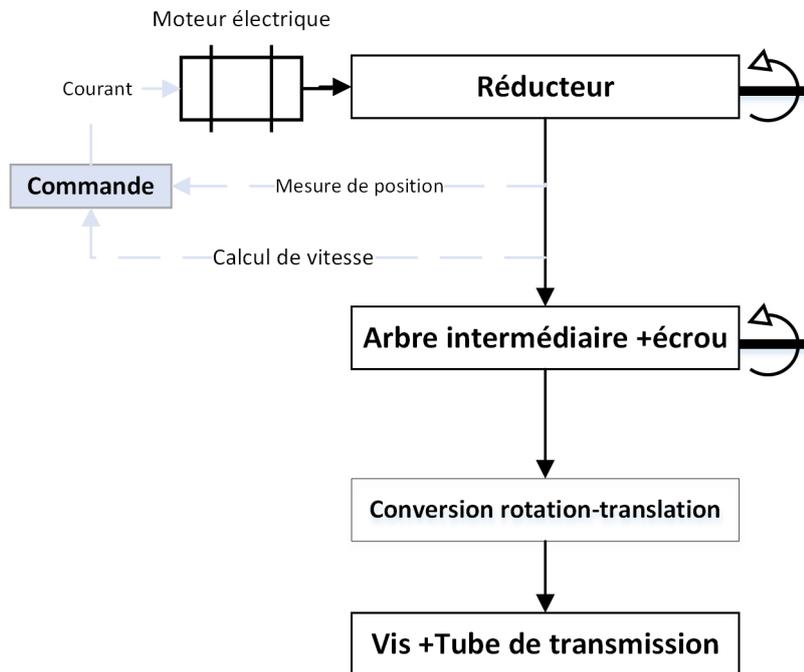


FIGURE 2.15 – Le bloc diagramme de l'actionneur

1. Moteur électrique "Brushless" :

Les moteurs à courant continu sont utilisés pour les vérins de faible puissance principalement dans le domaine médical. Les moteurs synchrones fournissent des puissances élevées mais sont peu adaptés au positionnement précis. Quant aux moteurs Brushless synchrones, ils sont utilisés là où un positionnement très précis est requis en combinaison avec une puissance importante et une grande accélération.

2. Réducteur de vitesse

Dans un système électromécanique, il est souvent nécessaire de prévoir une adaptation des couples, des vitesses et des moments d'inertie entre l'actionneur et la charge mécanique. Généralement les caractéristiques du moteur sont à prédéterminer en fonction de celle de la charge et il convient alors d'introduire " l'indice de transmission " comme paramètre de dimensionnement du système considéré. Un réducteur est donc un organe mécanique de couplage qui, pour un système de rotation, est constitué d'un train d'engrenage associé à deux arbres l'un menant tournant à la vitesse ω_1 et

l'autre mené tournant à la vitesse ω_2 . Si en plus on désigne par C_1 et C_2 les couples respectivement appliqué à l'arbre d'entrée et transmis par l'arbre de sortie à la charge mécanique, on définit alors le réducteur par :

$$\omega_1 = r \omega_2 \text{ et } C_2 \omega_2 = \eta C_1 \omega_1 \quad (2.1)$$

Où η est le rendement (moyen).

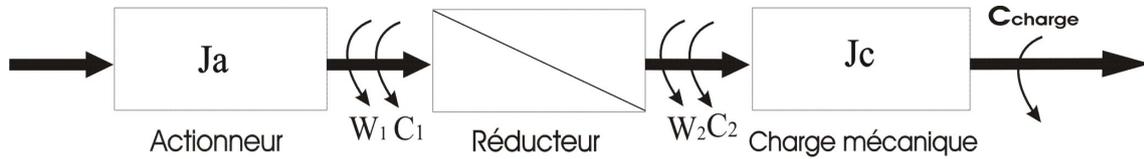


FIGURE 2.16 – Transmission de puissance par le réducteur de vitesse

où J_a, ω_1 représentent respectivement le moment d'inertie et la vitesse de l'arbre menant. J_c, ω_2 représentent respectivement le moment d'inertie et la vitesse de l'arbre mené. r, η représentent respectivement l'indice de réduction et le rendement du réducteur.

3. Arbre de transmission

Le couple produit par le moteur est transmis à la vis via une chaîne de transmission, conçue de manière à fournir un rapport de réduction déterminé avec une position bien particulière du moteur dû à l'encombrement ou juste à un accouplement qui autorise les défaut d'alignement et amortit les vibrations des deux axes.

4. Système vis-écrou

Les systèmes vis-écrou présentent des propriétés attractives pour la conception d'actionneurs électromécaniques linéaires (C6WP8). En particulier, ils permettent de réaliser un fort rapport de réduction entre le mouvement de rotation et le mouvement de translation.

Ce système transforme le mouvement de rotation de la vis en mouvement rectiligne de l'écrou. Une vis à billes comprend essentiellement une vis, un écrou fileté intérieurement et un ensemble de billes circulant dans le chemin formé par les filetages de l'écrou et de la vis(SAC84). Les billes qui circulent continuellement diminuent le frottement entre la vis et l'écrou et transmettent la charge. Par suite du faible frottement de roulement, le rendement est élevé. Les vis à billes s'imposent donc dans le cas de mouvements de précision.



FIGURE 2.17 – Vis à billes à recirculation interne

2.2.1.2 Sous-système de chargement

Description de fonctionnement du système de chargement

Le système de chargement a pour fonction d'appliquer un effort sur la porte de l'inverseur de poussée dans les conditions réelles de vol. La conception d'un tel système doit respecter le cahier de charges qui définit les conditions de fonctionnement réel.

La charge aérodynamique appliquée sur l'inverseur de poussée dépend de la configuration et la position de déploiement ou rétraction de l'inverseur de poussée. La courbe ci-dessous 2.18 représente l'évolution de l'effort aérodynamique dans chacune des configurations étudiées dans le cas PRESAGE.

- Déploiement normal (Normal Deploy) : déploiement normal de l'inverseur de poussée.
- Décollage abandonné (Rejected Take off) : déploiement normal de l'inverseur de poussée avec mach⁹ avion important.
- Rétraction normal (Normal Stow) : rétraction normale de l'inverseur de poussée.
- Emergency deploy : déploiement normal de l'inverseur de poussée avec augmentation du régime moteur au-dessus du nominal.

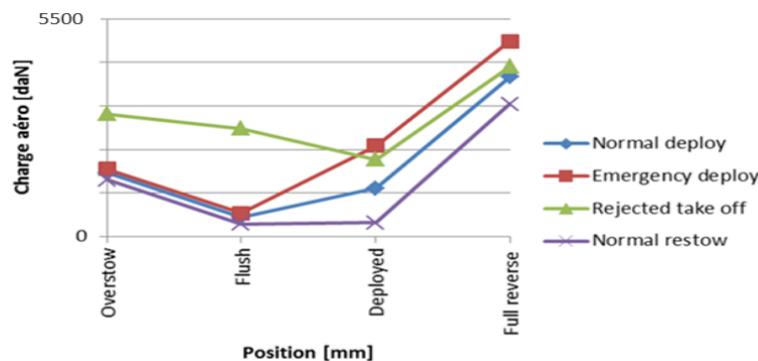


FIGURE 2.18 – Effort aérodynamique appliqué sur l'inverseur de poussée

9. Unité de mesure déterminant le rapport entre la vitesse d'un mobile à l'endroit où il se déplace avec la vitesse du son (CNR06).

Le système de chargement est composé d'une structure mécanique représentée par le vérin de charge aérodynamique, et d'une partie hydraulique de puissance afin de commander ce vérin. La figure ci-dessous 2.19 présente un diagramme du modèle du système chargement, dans laquelle nous illustrons ces deux parties mécanique et hydraulique.

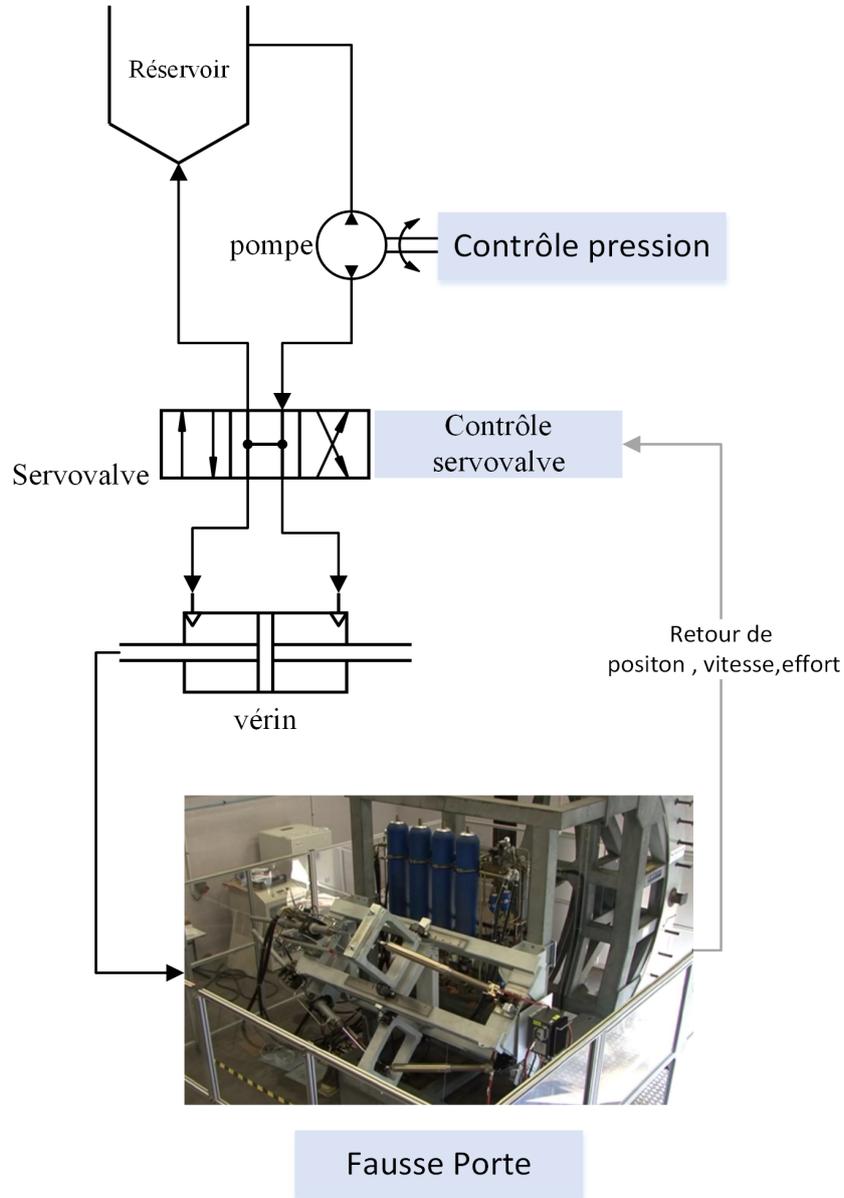


FIGURE 2.19 – Diagramme du modèle du système de chargement

2.2.2 La modélisation sous Dymola

Le projet PRESAGE « Plateforme Réelle Et Simulée d'Actionnement générique et Evolutive » a pour but le développement d'une plateforme d'essais virtuelle générique, applicable à différents domaines de l'industrie, et notamment l'aéronautique et l'automobile. C'est dans cette perspective que CERTIA exprime logiquement sa volonté d'effectuer la modélisation dans un environnement (Dymola) qui utilise le langage Modelica, référence dans le domaine aéronautique. En effet, cette plateforme visera à pré-évaluer des résultats d'essais d'intégration réels, à concevoir des algorithmes dédiés au diagnostic et au pronostic du banc d'essai et à rendre possible la simulation de pannes. PRESAGE a aussi pour ambition de proposer non seulement une approche générique de construction de modèles représentatifs pour les systèmes complexes mais aussi une architecture générique de bancs d'essais virtuels.

Par ailleurs, on peut dire que la modélisation des systèmes complexes sous des progiciels type AMESim ou Dymola n'est pas chose aisée, cependant, elle représentera un gain réel pour l'entreprise à moyen et à long terme. En effet, en utilisant ce type de modélisation et après avoir effectué toutes les vérifications et les tests de corrélation nécessaires. Le modèle obtenu aura un potentiel tel qu'il permettra le développement de structures de commande encore plus performante, ce qui aura aussi un impact direct sur la réduction des coûts de développement.

2.2.2.1 Sous-système d'actionnement

Un modèle d'actionneur électromécanique a été développé à partir de celui réalisé dans AMESim LMS, en se basant sur les rubriques translation et rotation de la bibliothèque mécanique standard de Dymola, et celles de la bibliothèque commerciale Modelon.

Les modèles ont été entièrement redéveloppés dans Dymola avec le langage Modelica. Chaque élément a été vérifié séparément pour comparer les résultats obtenus dans Modelica à ceux précédemment obtenus dans AMESim LMS.

Modèle sous Dymola

L'EMA dans le cas test PRESAGE doit fonctionner dans des conditions très sévères. Par conséquent, il est important de prendre en considération les efforts secondaires engendrés par le vérin linéaire. Dans notre modèle développé sous Dymola [2.20](#) des inerties équivalentes des éléments tournants et des efforts de frottement ont été prévus. À ceci s'ajoute des raideurs d'ancrages pour reproduire l'élasticité de la structure de la porte de l'inverseur de

poussée dans son environnement réel.

La figure 2.20 ci-dessous présente le modèle développé dans l'environnement Dymola/Modelica.

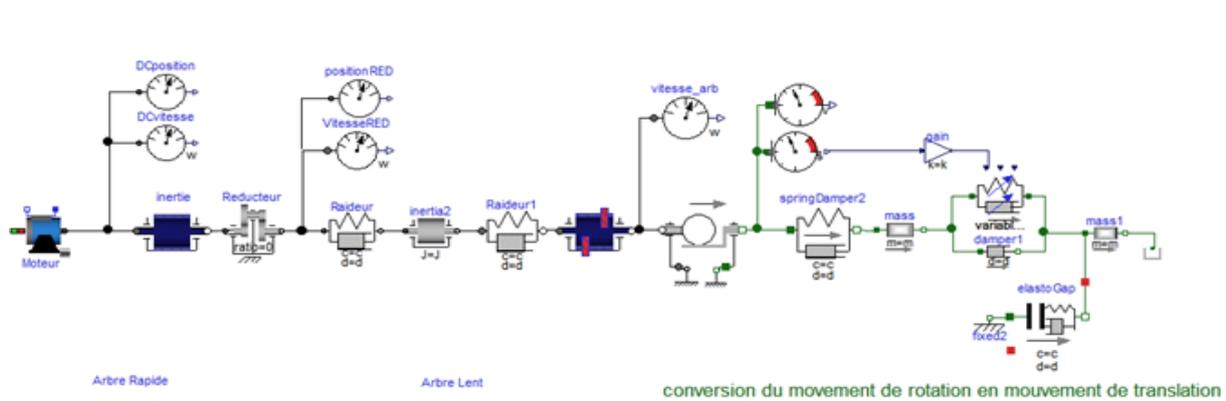
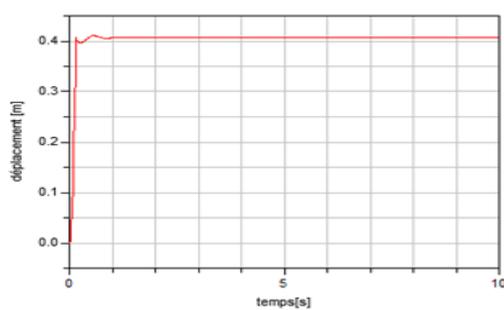


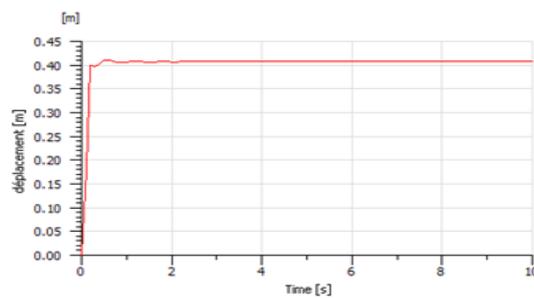
FIGURE 2.20 – Système d'actionnement développé sous Dymola

Validation du sous-modèle Dymola

Pour valider et effectuer une comparaison relative entre les résultats obtenus par les deux environnements de modélisation dynamique AMESim et Dymola, un essai a été effectué, pour une consigne en tension constante.

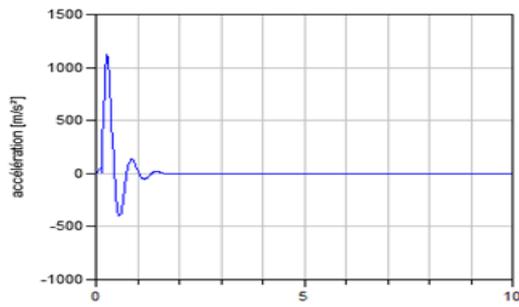


– (a)

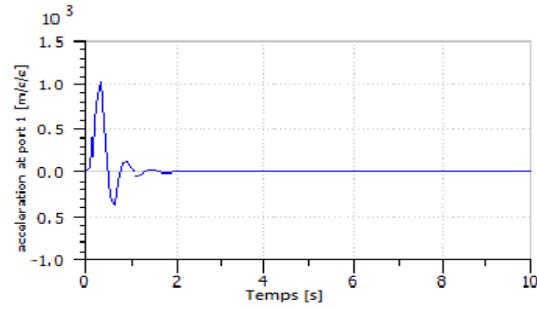


– (b)

FIGURE 2.21 – Comparaison entre l'allongement du tube du modèle : (a) AMESim et (b) Dymola

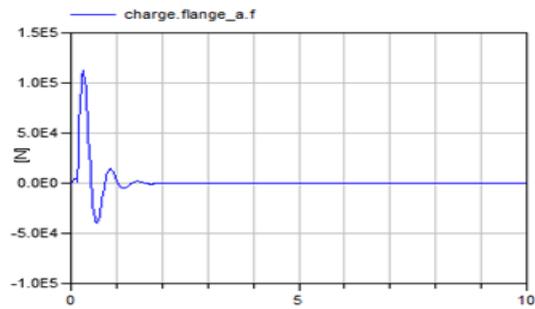


– (a)

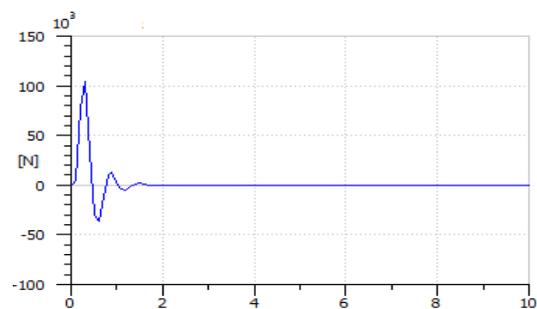


– (b)

FIGURE 2.22 – Comparaison entre l'accélération du tube du modèle : (a) AMESim et (b) Dymola



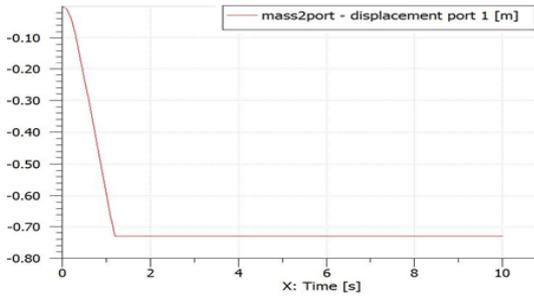
– (a)



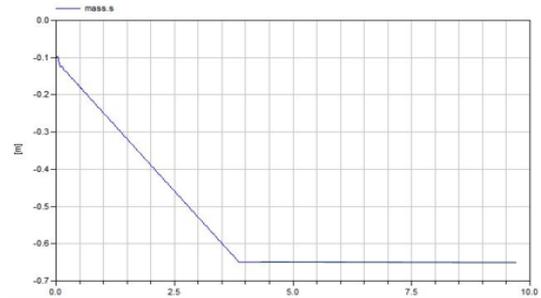
– (b)

FIGURE 2.23 – Comparaison entre la charge aérodynamique du modèle : (a) AMESim et (b) Dymola

Discussion Les figures ci-dessus 2.21, 2.22 et 2.23 montrent une bonne similitude entre les allures du modèle développé dans l'environnement de simulation dynamique Dymola/Modelica et celui développé dans l'environnement AMESim.



– (a)



– (b)

FIGURE 2.25 – Comparaison entre le déplacement de la structure mécanique du modèle :
(a) AMESim et (b) Dymola

2.3 Conclusion

Dans ce chapitre, après avoir donné une description physique du système d'inversion de poussée développé par la société AIRCELLE. Nous avons montré les différentes étapes qui nous ont permis de développer la plateforme virtuelle sous le progiciel Dymola tout en validant les différents sous modèles (système d'actionnement et de chargement). Lors de cette étape (validation), nous avons eu d'excellents résultats pour la partie système d'actionnement et des résultats nuancés pour la partie vérin de charge aérodynamique. Ces derniers pourront être maîtrisés par une meilleure connaissance des différences existantes entre les deux logiciels AMESim et Dymola et notamment dans la partie hydraulique.

Par ailleurs, lors du développement nous avons été contraints de coder certains composants pour reproduire exactement le comportement obtenu sous AMESim, ce développement a été rendu possible par l'existence d'un éditeur Modelica au sein du logiciel AMESim.

Deuxième partie

Commande

Introduction

En aéronautique comme dans tous les secteurs, la recherche et l'innovation sont plus que jamais des leviers majeurs et nécessaires à la pérennité de n'importe quel entreprise. CERTIA “ Centre Etudes Recherches Techniques Industries Appliquées”, acteur du grand projet PRESAGE “ Plateforme Réelle Et Simulée d'Actionnements, Générique et Adaptive” s'est donné les moyens humains (experts, ingénieurs, coopération avec les universités et écoles en France et en Algérie) et matériels nécessaires. C'est dans cette perspective que nous avons pris le bateau “ PRESAGE” pour le mener à bon port ,avec l'aide de l'encadrement pédagogique (ENP) et professionnel (CERTIA RD).

Notre travail est d'apporter une plus-value réelle à la commande déjà développée (basée sur un PID simple) pour le système d'inversion de poussée du nacelliste AIRCELLE.

Dans cette partie commande nous verrons deux techniques développées pour la commande robuste du système d'inversion de poussée. La première est la commande par H_∞ Loopshaping comme définie dans (MCF92) et (PAS11). La seconde est une classe particulière de la commande par Backstepping pour les systèmes triangulaire.

Commande par H_∞ Loopshping

3.1 Introduction

La synthèse H_∞ est une méthode qui sert à la conception de commandes optimales. En d'autres termes, il s'agit d'une méthode d'optimisation, qui prend en compte la définition mathématique des contraintes pour ce qui est du comportement attendu en boucle fermée. Par ailleurs, cette commande robuste est conçue de sorte à imposer des contraintes de type min et max dans un environnement fréquentiel (on verra ceci dans ce qui suit avec le filtre W_1 *parexemple*).

Nous structurons cette partie comme suit : après une brève introduction sur la relation qui lie l'automatique à la robustesse, nous allons donner le modèle d'état utilisé qui a été développé par l'ESIGELEC et obtenu à partir de la théorie du Bond Graph, modèle sur lequel nous allons appliquer notre commande par H_∞ Loopshaping. Par la suite, nous allons exposer les résultats de simulation en boucle ouverte. Enfin, nous aborderons l'application de cette technique de commande sur le système d'inversion de poussée dans le cas test « PRESAGE », tout en mettant l'accent sur la robustesse de la commande par H_∞ vis-à-vis des erreurs de modélisation ainsi que de perturbations externes.

3.2 L'Automatique et la commande robuste

En septembre 1992, J.M Maciejowsky (MAC89) définit la notion de robustesse de la manière suivante : “ Les propriétés du système se conservent dans un ensemble ouvert ” ou de façon équivalente “ De petites variations ne perturbent pas les propriétés d'un système”. L'ensemble ouvert dont parle J.M Maciejowsky n'est rien d'autre que l'espace où peut évoluer

le système sous l'effet des perturbations, par natures inconnues. On peut décrire cet ensemble de deux façons. La première est de donner ses " frontières " c'est-à-dire caractériser les évolutions maximales possibles pour le système. La seconde est de définir un point nominal, puis des directions de variations possibles autour de ce point nominal. Ces dernières peuvent être par nature de plusieurs types, représentant divers types d'incertitudes (erreurs de modélisation, non linéarités, variations paramétriques...).

Il y'a de cela moins d'un demi-siècle, au début de l'automatique, une des propriétés fondamentales recherchées était de rendre le système en boucle fermée moins sensible aux incertitudes de modélisation qu'en boucle ouverte. Cette propriété fondamentale a retrouvé un regain d'intérêt il y'a une vingtaine d'années après une longue période d'occultation. Les techniques fréquentielles des années 1950-1960, fondées essentiellement sur une approche graphique, ont débouché aux régulateurs PID et font d'eux actuellement les correcteurs les plus utilisés dans le milieu industriel. Ces techniques (lieu de Black, de Bode, de Nyquist, d'Evans...)(BOU10) ont introduit les notions de marge de gain, de phase et donc les premières exigences de robustesse. A cette période, un lien théorique entre l'insensibilité et la structure n'avait pas encore été formulé, et ces méthodes ne s'avéraient adaptées qu'au cas monovarié. En effet, ce n'est qu'au cours des années 1960-1980 que des développements théoriques de l'automatique ont permis de traiter de façon plus profonde les systèmes multivariés. En revanche, le problème de robustesse vis-à-vis des incertitudes de modélisation reste entier. Plus encore, les théories optimales des systèmes multivariés (LQ, LQG, observateur, filtre de Kalman...) supposent une connaissance parfaite du système. Pour répondre à cette exigence (robustesse), se sont donc parallèlement développées des techniques d'identification paramétrique. En d'autres termes, on considère qu'une meilleure connaissance des paramètres du système permettent d'atteindre de meilleures performances en commande.

Ces dernières théories ont montrés leurs limites lorsque la complexité grandissante des lois de commande rendait le système hypersensible aux incertitudes de modélisation (modes souples ignorés, incertitudes paramétriques...). L'approche auto-adaptative a tenté de résoudre ce conflit en ajustant en temps réel la connaissance que nous pouvions avoir du système, afin de parfaire la commande (GOO84). En d'autres termes, il existe un compromis à trouver entre le degré de fidélité du modèle et la faisabilité de la commande.

Depuis le début des années 1980, les développements théoriques se sont tournés à nouveau vers les problèmes liés à la robustesse des lois de commandes. Doyle et Stein (?), Safanov(SAF81), Lehtomaki(LEC84), posent les bases fondamentales d'une théorie exi-

geant d'une commande une robustesse aux incertitudes de modélisation et précisent les compromis entre de bonnes performances et une excellente robustesse. Les approches de type LQG/LQR et H_∞ (c'est cette dernière que nous développerons plus tard dans ce chapitre) permettent de traiter le cas des incertitudes fréquentielles (caractérisant bien les erreurs de modélisation telles les dynamiques négligées). Par ailleurs, les travaux de Kharitonov et plus particulièrement ce qu'on appelle les polynômes de Kharitonov (TAD11) permettent de caractériser la stabilité d'un système soumis à des perturbations paramétriques bornées.

3.3 Modèles d'état du système d'inverseur de poussée

Le LAGIS¹ et l'INSA² de Toulouse, responsables de la partie modélisation dans le projet PRESAGE, ont développées deux modèles mathématique représentant au mieux la dynamique du système d'inversion de poussée. Le premier modèle est à six états tandis que le second en contient quatre. Dans ce qui suit nous allons, en premier lieu, exposer les deux représentations d'état en question. En second lieu nous allons faire une simulation en boucle ouverte.

Les modèles mathématique développés par les deux établissements français cités plus haut ont été tirés à partir de la modélisation par Bond Graph. Le but du travail effectué dans ce mémoire n'est pas de décrire la manière avec laquelle ces deux représentations ont été obtenues. Par contre, le lecteur aura le loisir de trouver dans (VER05), (THM00) et (FIC04), les outils nécessaires qui lui permettront de passer d'une représentation par Bond Graph à un modèle d'état. Aussi, l'utilisation du Bond graph n'est pas fortuite, bien au contraire, cette méthodologie permet la représentation d'un objet actif finalisé, physique ou immatériel en interaction avec ses environnement par l'intermédiaire du flux (énergétique, informationnels ou matériels). En d'autres termes, l'outil Bond Graph est un langage graphique unifié pour tous les domaines des sciences de l'ingénieur. Il est basé sur la notion de transfert d'énergie ou de puissance entre les éléments du système et s'appuie sur la notion de causalité.

1. Ecole Centrale de Lille

2. Institut national des sciences appliquées de Toulouse

3.3.1 Représentation d'état à six variables

3.3.1.1 Modèle non linéaire

Le modèle d'état issu du Bond Graph est donné par la représentation d'état suivante :

$$\begin{cases} \dot{x}(t) = f(x(t), l_{tube}(t)) + B.u(t) + H.C_{aero} \\ y(t) = C.x(t) \end{cases} \quad (3.1)$$

Dans cette représentation $x \in \mathbb{R}^6$ désigne le vecteur d'états du système considéré, $u \in \mathbb{R}$ Le signal de commande, $y \in \mathbb{R}$ désigne la variable à piloter et C_{aero} représente la charge aérodynamique appliquée sur la porte de l'inverseur.

Avec :

$$f(x(t), l_{tube}(t)) = \begin{cases} -\frac{R_a}{2L_a} x_1(t) - \frac{K_{em}}{J_m} x_2(t) \\ \frac{K_{em}}{2L_a} x_1(t) - \frac{F_m}{J_m} x_2(t) - \frac{K_{arbre}}{N} x_3(t) \\ \frac{1}{N} x_2(t) - \frac{1}{J_m} x_4(t) \\ K_{arbre} x_3(t) - \psi_{MR}(x_4(t), \psi_{tube}) - PV \psi_{tube}(x_5(t), l_{tube}(t)) \\ \frac{PV}{J_m} x_4(t) - BL(l_{tube}) \frac{1}{J_{porte}} x_6(t) \\ BL(l_{tube}) \psi_{tube}(x_5(t), l_{tube}(t)) - \frac{F_{porte}}{J_m} x_6(t) \end{cases}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad H = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{et} \quad C = \left(0 \quad \frac{1}{J_m} \quad 0 \quad 0 \quad 0 \quad 0 \right)$$

Tel que :

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{pmatrix} = \begin{pmatrix} \phi_{La}(t) \\ h_{jm}(t) \\ \theta_{arbre}(t) \\ h_{jvis}(t) \\ x_{tube}(t) \\ h_{jporte}(t) \end{pmatrix} \quad \text{et} \quad \dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{pmatrix} = \begin{pmatrix} u_{La}(t) \\ C_{jm}(t) \\ \omega_{arbre}(t) \\ C_{jvis}(t) \\ \dot{x}_{tube}(t) \\ C_{jporte}(t) \end{pmatrix}$$

Où :

ϕ_{L_a}	Flux magnétique à travers le bobinage du moteur.
h_{J_m}	Moment cinétique de l'arbre.
θ_{arbre}	Position de l'arbre rotatif.
$h_{J_{vis}}$	Moment cinétique de la vis.
x_{tube}	Déplacement du bras de levier.
$h_{J_{porte}}$	Moment cinétique de la porte.
$BL_{tube}, \psi_{MR}, \psi_{tube}$	Non linéarités du système de conduite.

L'allongement du tube mobile de l'actionneur peut être calculé à l'aide de l'équation suivante :

$$V_{tube} = \begin{pmatrix} 0 & 0 & 0 & \frac{PV}{J_{vis}} & 0 & 0 \end{pmatrix} x(t) \quad (3.2)$$

Les valeurs numériques des paramètres du système sont données dans le tableau :

	<i>Composant</i>	<i>Valeur</i>	<i>Unité</i>
Moteur	R_a	0.055	Ω
	L_a	0.004	H
	F_m	$7e - 4$	$N.m.rad^{-1}$
	J_m	0.200	$Kg.m^{-2}$
	K_{em}	15000.000	$N.m.A^{-1}$
Réducteur	N	5700.000	
Arbre	K_{arbre}	$6e - 7$	$Kg.m^{-2}$
Vis à billes	J_{vis}	0.054	$Kg.m^{-2}$
	F_{vis}	0.020	$N.m.rad^{-1}$
	PV	0.001	$m.rad^{-1}$
Porte	F_{porte}	200.000	$N.m.rad^{-1}$
	J_{porte}	30.000	$Kg.m^{-2}$

TABLE 3.1 – valeurs numériques paramètres du système de conduite

Les expressions des fonctions non-linéaires et les valeurs numériques des paramètres rencontrés sont données par la suite dans l'annexe C.

3.3.1.2 Modèle linéaire

Une approximation des non linéarités du modèle précédent nous permet d'avoir un modèle linéaire proche de la réalité physique du système. Sa représentation d'état est donnée par :

$$\begin{cases} \dot{x}(t) = f(x(t), l_{tube}(t)) + B u(t) + HC_{aero} \\ y(t) = Cx(t) \end{cases} \quad (3.3)$$

Avec :

$$A = \begin{pmatrix} -\frac{R_a}{2L_a} & -\frac{K_{em}}{F_m} & 0 & 0 & 0 & 0 \\ \frac{K_{em}}{2L_a} & -\frac{J_m}{F_m} & -\frac{K_{arbre}}{N} & 0 & 0 & 0 \\ 0 & \frac{1}{J_m N} & 0 & -\frac{1}{J_{vis}} & 0 & 0 \\ 0 & 0 & K_{arbre} & -\frac{F_{vis}}{PV} & -PVK_{tube} & 0 \\ 0 & 0 & 0 & \frac{J_{vis}}{PV} & 0 & -\frac{BL}{J_{porte}} \\ 0 & 0 & 0 & 0 & BLK_{tube} & -\frac{F_{porte}}{J_{porte}} \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad H = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{et} \quad C = \left(0 \quad \frac{1}{J_m} \quad 0 \quad 0 \quad 0 \quad 0 \right)$$

3.3.2 Représentation d'état à quatre variables

Un modèle mathématique simplifié à quatre variables d'état a pu être obtenu à partir de celui à six états. Pour ce faire, les variables ayant des dynamiques élevées sont enlevées du premier modèle. De la même manière que le cas précédent, deux modèles mathématiques à quatre variables d'états sont alors considérés.

3.3.2.1 Modèle non linéaire

La représentation d'état à quatre variables est donnée par le système (3.1) avec :

$$f(x(t), l_{tube}(t)) = \begin{cases} -\frac{R_a}{2L_a}x_1(t) - \frac{K_{em}}{J_m}x_2(t) \\ \frac{K_{em}}{2L_a}x_1(t) - \frac{\left(F_m + \frac{F_{vis}}{N^2}\right)}{J_m}x_2(t) - \frac{PV}{N}\psi_{tube}(x_3(t), l_{tube}(t)) \\ \frac{PV}{Nj_m}x_2(t) - \frac{1}{J_{porte}}BL(l_{tube}(t))x_4(t) \\ BL(l_{tube}(t))\psi(x_4(t), l_{tube}(t)) - \frac{F_{porte}}{J_{porte}}x_4(t) \end{cases}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, H = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \text{ et } C = \begin{pmatrix} 0 & \frac{1}{J_m} & 0 & 0 \end{pmatrix}$$

Tel que :

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} \phi_{La}(t) \\ h_{j_m}(t) \\ x_{tube}(t) \\ h_{j_{porte}}(t) \end{pmatrix} \text{ et } \dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{pmatrix} = \begin{pmatrix} u_{La}(t) \\ C_{j_m}(t) \\ \dot{x}_{tube}(t) \\ C_{j_{porte}}(t) \end{pmatrix}$$

L'allongement du tube mobile de l'actionneur peut être calculé à l'aide de l'équation suivante :

$$V_{tube} = \begin{pmatrix} 0 & \frac{PV}{J_{vis}} & 0 & 0 \end{pmatrix} x(t) \quad (3.4)$$

3.3.2.2 Modèle linéaire

Dans un voisinage d'un état d'équilibre, les termes non linéaires sont linéarisés par différenciation au premier ordre. Cette linéarisation nous conduit à une représentation d'état sous la forme (3.3) avec :

$$A = \begin{pmatrix} -\frac{R_a}{2L_a} & -\frac{K_{em}}{J_m} & 0 & 0 \\ \frac{K_{em}}{2L_a} & \frac{J_m}{F_m} & -\frac{K_{arbre}}{N} & 0 \\ 0 & \frac{J_m}{PV} & 0 & -\frac{BL}{J_{porte}} \\ 0 & 0 & BLK_{arbre} & \frac{F_{porte}}{J_{porte}} \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, H = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \text{ et } C = \begin{pmatrix} 0 & \frac{1}{J_m} & 0 & 0 \end{pmatrix}$$

3.3.3 Simulation du système en BO

Afin d'évaluer les caractéristiques de la conduite de notre actionneur électromécanique nécessaire à la synthèse de la commande, il est primordial de savoir le comportement de ce dernier en boucle ouverte. Cette partie est dédiée à l'évaluation des différents effets perturbateurs et à l'analyse du système en boucle ouverte.

3.3.3.1 Systèmes non linéaires

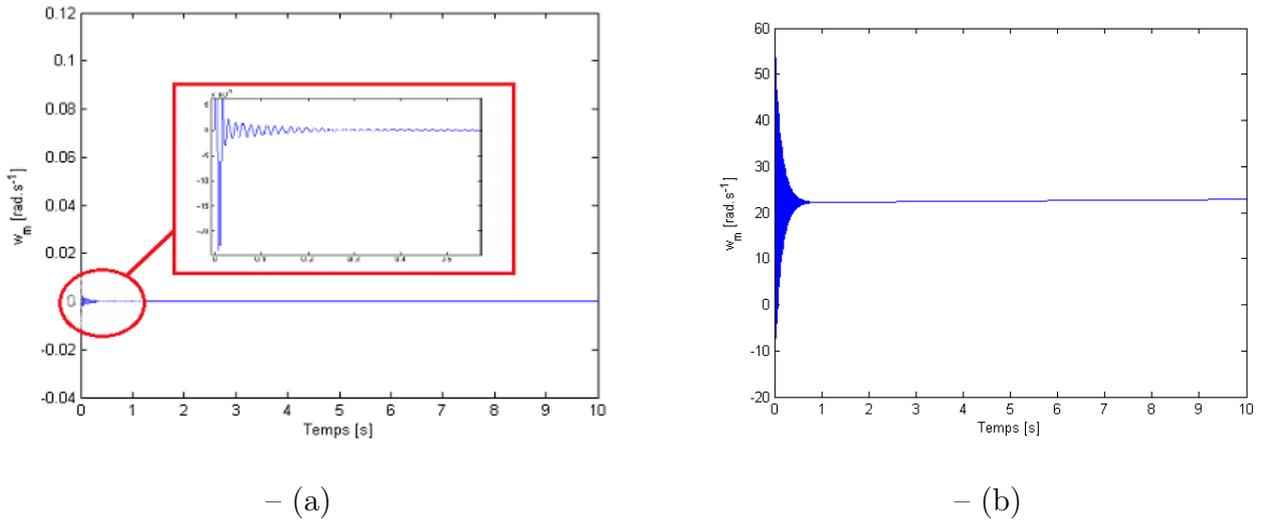


FIGURE 3.1 – La réponse du système en BO du système : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique

Discussion Les figures 3.1, 3.2, 3.3 montrent la convergence du système vers le point d'équilibre en absence de la charge aérodynamique.

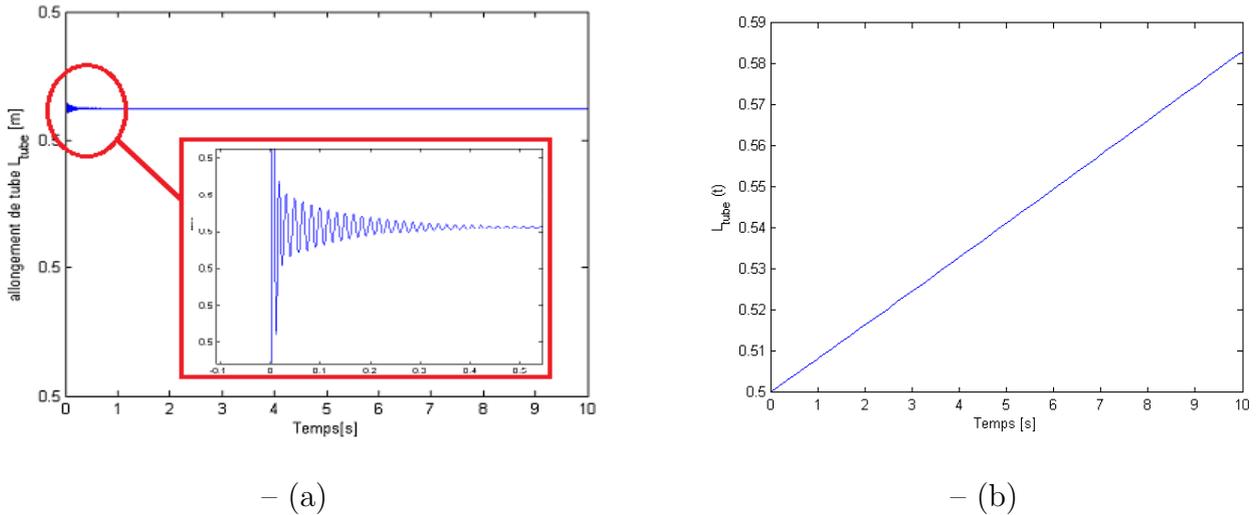


FIGURE 3.2 – Allongement du tube L_{tube} en BO : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique

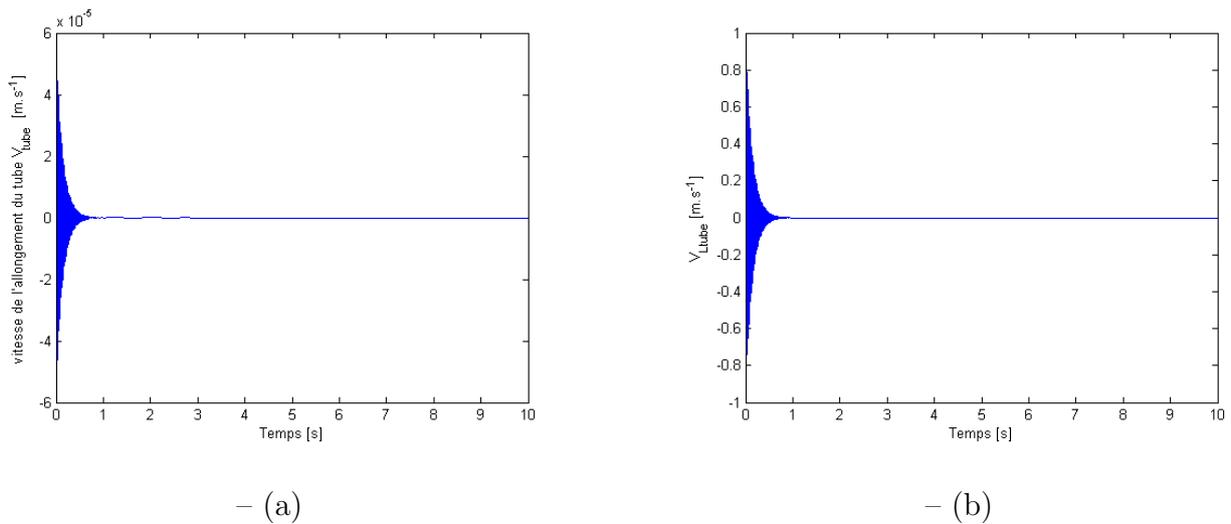


FIGURE 3.3 – La vitesse d'allongement du tube L_{tube} en BO : (a) en absence de la charge aérodynamique (b) en présence de la charge aérodynamique

3.4 Synthèse de la loi de commande par H_∞ Loopshaping

3.4.1 Définitions

i. Norme H_∞

La norme H_∞ d'un système donne le gain maximum possible sur toute les fréquences.

Ainsi, en utilisant cette norme, on peut chercher des régulateurs minimisant un critère donné. Le calcul de cette norme pour un signal temporel noté $x(t)$:

$$\|x(t)\|_\infty = \sup_t |x(t)| \quad (3.5)$$

La norme H_∞ d'un système multivariable est son amplification maximal. Pour un système $G(s)$ tel que :

$$y(s) = G(s)u(s) \quad (3.6)$$

On a :

$$\|G(s)\|_\infty = \max \frac{\|y(t)\|_2}{\|u(t)\|_2} = \int_0^\infty y^H(t)y(t) dt \quad (3.7)$$

On a aussi :

$$\|G(s)\|_\infty = \max \|y(t)\|_{w \in \mathbb{R}^+} \bar{\sigma}(G(jw)) \quad (3.8)$$

Où $\bar{\sigma}$ est la plus grande des valeurs singulières. En monovariante, on a :

$$\bar{\sigma}(G(jw)) = |G(jw)| \quad (3.9)$$

Et

$$\|G(s)\|_\infty = \max_{w \in \mathbb{R}^+} |G(jw)| \quad (3.10)$$

Ce dernier est appelé gain maximal sur l'ensemble des fréquences. Par ailleurs, il existe d'autres critères comme la norme H_2 (STE87), des synthèses multicritères sont également possible (SCH95), (CHI96), (ARZ02) et (ARZ04). Citons également les liens avec la théorie des jeux (JAN03), (ABO03), (ABO04).

En d'autres termes, la norme H_∞ d'un système donne le gain maximum possible sur toutes les fréquences. Ainsi, en utilisant cette norme on peut chercher des régulateurs minimisant un critère donné.

ii. *Performance en commande d'un système*

On appelle performances en commande d'un système le rejet de perturbation ou la poursuite du signal de commande.

iii. *Fonctions de sensibilité et sensibilité complémentaire*

Ce sont des quantificateurs de performances que nous allons utiliser par la suite lors de commande par H_∞ Loopshaping (PAS11), (MCF92). Considérons le schéma de commande suivant :

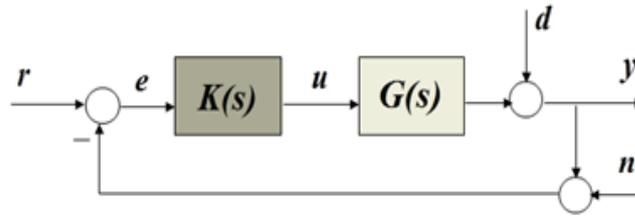


FIGURE 3.4 – Schéma de commande

Tel que :

- r** signal de référence.
- d** signal de perturbation en sortie.
- n** bruit de mesure.
- e** erreur entre la référence et la sortie du système.
- y** signal de sortie
- u** signal de commande

La fonction de sensibilité $S_y(s)$ est défini comme suit :

$$S_y(s) = \frac{Y(s)}{D(s)} = \frac{1}{1 + G(s)K(s)} \quad (3.11)$$

La fonction de sensibilité complémentaire $T(s)$ est défini par :

$$T_y(s) = \frac{G(s)K(s)}{1 + G(s)K(s)} \quad (3.12)$$

Aussi, nous avons la relation suivante :

$$S_y(s) + T_y(s) = I \quad (3.13)$$

3.4.2 Problème H_∞ standard

Nous avons vu précédemment que les performances et la robustesse peuvent être décrits comme des critères fréquentiels sur la fonction de sensibilité et la fonction de sensibilité complémentaire. Ainsi, pour réaliser des performances sur une plage de fréquence, cela revient à chercher le régulateur $R(s)$ (**TAD11**), qui minimise la quantité suivante :

$$\inf \|F_l(P, R)\|_\infty = \inf \sup_w \sigma(F_l(P, R)(jw)) \quad (3.14)$$

Le schème de commande **A.5** précédent peut transformé comme suit :

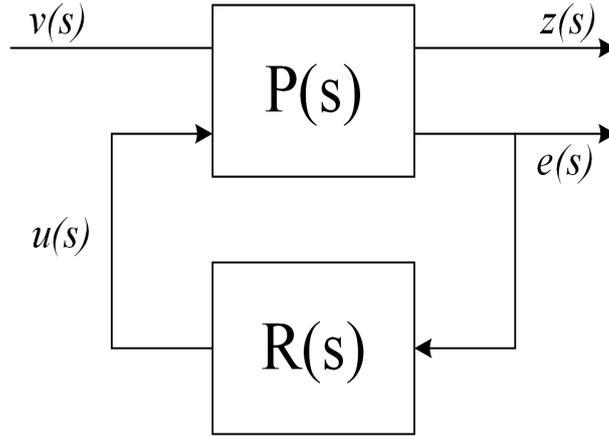


FIGURE 3.5 – Le problème standard de la commande

Dans le schéma ci-haut, on peut voir deux canaux de signaux :

- le canal de performance ou d’incertitude, d’entrée v et de sortie z .
- le canal de commande, d’entrée u et de sortie y .

Le transfert en boucle fermée des signaux extérieurs $v(s)$ vers les quantificateurs de performances $z(s)$ est :

$$z(s) = F_l(P, R).v(s) \quad (3.15)$$

$$\begin{cases} z = P_{11}v + P_{12}u \\ e = P_{21}v + P_{22}u \end{cases} \quad (3.16)$$

Après développement on trouve :

$$F_l(P, R) = P_{11} + P_{12}R[I - P_{22}R]^{-1}P_{21} \quad (3.17)$$

3.4.3 Problème de sensibilité mixte

Devant la complexité du système d’inverseurs de poussée et étant donné les nombreuses contraintes auxquelles est soumis la porte (effort aérodynamique) ,nous avons jugé utile de choisir une commande qui assurera d’une part des performances et d’autres part une robustesse vis-à-vis des incertitudes de modélisation. Pour cela, notre choix s’est porté vers la commande robuste H_∞ Loopshaping (TAD11), (MCF92) avec pour valeur du critère à minimiser :

$$F_l(P, R) = \begin{pmatrix} W_1 S_y \\ W_3 T_y \end{pmatrix} \quad (3.18)$$

Où W_1 et W_3 sont des pondérations qui nous permettrons d'imposer au système une certaine robustesse en stabilité et en performances, et cela à travers la bande passante désiré du système ou l'atténuation minimale à une éventuelle perturbation par exemple.

Dans cette méthode, on construit un modèle augmenté du système en choisissant les signaux exogènes v et z et en réglant un certain nombre de pondération (cités ci-haut) permettant de donner une forme particulière aux transferts en boucle fermée. De nombreux schémas de synthèse peuvent être utilisés suivant l'objectif à atteindre. Nous présentons dans la suite le schéma de synthèse à 2 blocs qui a l'avantage d'être à la fois simple tout en permettant de résoudre la majeure partie des problèmes rencontrés en commande. Comme précisé précédemment, le correcteur doit permettre une certaine robustesse. Cela s'obtient en assurant deux critères :

- la marge de module M qui est déterminé par la norme H_∞ de la sensibilité du système. Pour cela, on considère généralement la sensibilité en sortie S_y .

$$\Delta_m = \frac{1}{\|S_y\|_\infty} \quad (3.19)$$

- En haute fréquence, les systèmes ont généralement un gain décroissant ce qui fait que leurs dynamiques sont mal connues. Afin d'assurer un certain niveau de robustesse vis-à-vis de ces dynamiques, il est nécessaire d'assurer une limitation, voire une décroissance du gain du correcteur en haute fréquence.

Le schéma de synthèse à deux blocs présenté sur la figure 3.6 est le schéma le plus simple permettant à la fois de régler les deux objectifs de robustesse mentionnées ci-dessus. En effet, le correcteur $K(s) = R(s)$ est synthétisé de manière à minimiser la norme H_∞ du transfert entre r et z .

La pondération $W_1(s)$ permet à la fois de régler la bande passante du système et de choisir la marge de module.

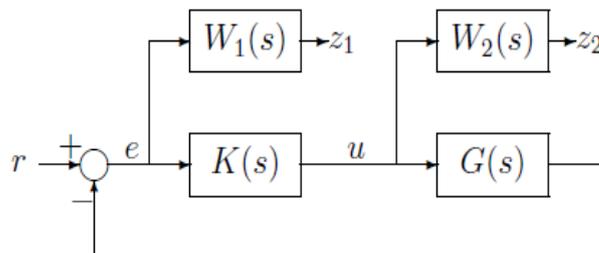


FIGURE 3.6 – Le schéma de synthèse à deux blocs

Prenons par exemple :

$$W_1 = \frac{s + a}{k(s + b)} \quad (3.20)$$

Avec $a < b$ et $k \geq 1$. Ce transfert a un gain statique de $\frac{a}{kb}$ (ie $s = 0$) en haute fréquence, un gain minimal de $\frac{1}{k}$ (en basse fréquence) et présente un gain à $3dB$ à la pulsation :

$$\omega_m = \sqrt{\frac{a^2 - 2k^2b^2}{2k^2 - 1}} \quad (3.21)$$

Si on impose $\|T_y\|_\infty \leq \gamma$, on a alors :

$$\bar{\sigma}(S_y(jw)) \leq \frac{\gamma}{|W_1|} \quad (3.22)$$

Ce qui va nous permettre d'assurer les caractéristiques suivantes :

- Une marge de gain supérieure à $\frac{1}{k}$.
- Une bande passante d'au moins ω_m .
- Une erreur statique inférieure à $\frac{kb}{a}$ (dans le cas test " PRESAGE " et comme précisé dans le cahier de charge, l'erreur dynamique doit être inférieure à 1% de la consigne).

D'autre part, choisissons W_2 comme suit :

$$W_2 = \frac{s}{l(cs + 1)} \quad (3.23)$$

Avec $\omega_m \ll 1$, une valeur de l très faible correspondra à un effet de roll-off important, c'est-à-dire une décroissance rapide du gain en haute fréquence.

On aura donc finalement,

$$\bar{\sigma}(T_y(jw)) \leq \frac{\gamma}{|W_2|} \quad (3.24)$$

Et ainsi,

$$\bar{\sigma}(R(jw)) \leq \frac{\gamma}{|W_2|} \quad (3.25)$$

Dans notre cas, pour choisir les pondérations nous avons imposé une atténuation aux perturbations de $-40 dB$ pour une bande passante désirée de $100 rd/s$.

On obtient finalement après calculs,

$$W_1 = \frac{s + 200}{2s + 0.2} \quad (3.26)$$

$$W_2 = \frac{s + 50}{0.001s + 100} \quad (3.27)$$

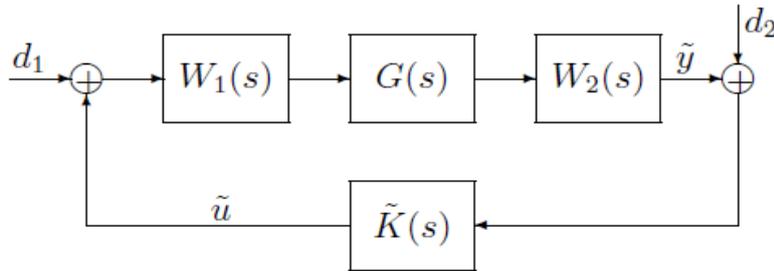


FIGURE 3.7 – Le schéma de commande H_∞ Loopshaping

En conclusion, la méthode du loopshaping se focalise sur le transfert en boucle ouverte du système système-correcteur (MCF92). Les pondérations W_1 et W_2 sont placés en série avec le système ,respectivement en amont et en aval. Elles sont destinés à fixer la bande passante du système, à augmenter le gain en basse fréquence pour assurer le rejet de perturbation et à faire chuter le gain en haute fréquence pour des raisons de robustesse vis-à-vis des erreurs de modélisation.

3.5 Application au système d'inverseurs de poussée

3.5.1 Représentation d'état du système

Nous avons vu précédemment dans la partie modélisation les modèles non linéaires fournis par les équipes de l'Ecole centrale de Lille et de l'INSA de Toulouse. Nous avons choisis d'effectuer quelques modification dans la structure du modèle afin de le rendre plus adéquat à notre commande.

Nous avons dit auparavant que le modèle que nous avons considéré pour la commande est un modèle ajusté ou modifié structurellement, et ce au vu du type de commande que nous voulons appliquer. En l'occurrence, pour notre cas, dans ce chapitre, la commande robuste H_∞ Looshaping .

Le problème est que dans le modèle présenté ci-dessus (3.3), l'effort aérodynamique représenté par C_{aero} est considéré comme une entrée exogène au système (ie une perturbation).

Notre but est de rendre le système sous la forme :

$$\begin{cases} \dot{x}(t) = f(x(t)) + B.u(t) \\ y(t) = C.x(t) \end{cases} \quad (3.28)$$

Pour cela, deux méthodes (ou modifications) peuvent être envisagées :

- Considérer la perturbation comme deuxième entrée de commande .En d'autre termes, le vecteur B et la commande u vont changer de dimension.
- Inclure la perturbation C_{aero} dans la matrice A du modèle d'état, sa dimension sera ainsi augmenté à 5 états(4+1état dû à C_{aero}). Ceci est rendu possible par la connaissance à priori de la dynamique de la perturbation en question ,ainsi que du fait qu'elle soit mesurable.

Dans ce qui suit, nous avons choisi la seconde option car le calcul du transfert Entrée–Sortie sera beaucoup plus simple comparé à la première option.Cependant, la première méthode a un avantage qu'on ne peut omettre de citer. En effet, sur le long terme, si CERTIA à travers ses clients désire faire une étude plus approfondis, dans le sens ou on ne va plus considérer l'effort aérodynamique sur un seul atterrissage mais plutôt faire une étude sur un avion donné en prenant en compte tous les atterrissages effectué durant six (6) mois par exemple. Dans ce cas ,on devra avoir un fichier qui comportera les valeurs de la perturbation C_{aero} , et dans ce cas la première méthode sera beaucoup plus adéquate.

3.5.2 Le système à 5 états

Le nouveau système considéré à 5 états est sous la forme (3.3) avec :

$$A = \begin{pmatrix} -\frac{R_a}{2L_a} & -\frac{K_e m}{J_m} & 0 & 0 & 0 \\ \frac{K_e m}{2L_a} & -\frac{F_m + \frac{(F_{vis})}{N^2}}{J_m} & \frac{-PV K_{tube}}{N} & 0 & 0 \\ 0 & \frac{PV}{J_m N} & 0 & -\frac{BL}{J_{porte}} & 0 \\ 0 & 0 & BLK_{tube} & -\frac{F_{porte}}{J_{porte}} & 1 \\ 0 & \alpha_i & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & \frac{1}{J_m} & 0 & 0 & 0 \end{pmatrix} \text{ et } D = 0$$

Le paramètre α_i déterminera dans quelles phases de déploiement l'inverseur de poussée se trouve. Elles sont comme suit :

- Overstow : Correspond à un déploiement de l'actionneur électromécanique de la position -2mm à 405mm.(le signe moins (-) est dû au fait qu'il existe un espace servant à verrouiller la porte de l'inverseur lorsque l'avion est en vol).
- Flush : Correspond au déploiement de la valeur 0mm à 405mm.
- Deployed : Atteint lorsque l'actionneur est à son élongation maximale 405mm .
- Full reverse : Phase statique de l'actionneur à 405mm durant laquelle la force de contre poussée sera la plus importante.

Dans le schéma suivant figure 3.8, on peut visualiser quatre (4) différents modes à considérer :

- Normal deploy : Correspond à une ouverture normale des deux portes de l'inverseur de poussée lors de l'atterrissage .
- Emergency deploy : Correspond à une procédure d'urgence où le pilote va solliciter l'ouverture des portes de l'inverseur sachant que la vitesse du moteurs dépasse sa valeur nominale en phase de roulage.
- Rejected Take off : Correspond à une urgence en phase de décollage où ce dernier est annulé par le pilote en cas d'urgence.

- Normal restow : Correspond à la phase post-déploiement ,la porte reviendra à sa position initiale après la phase Full reverse définit plus haut.

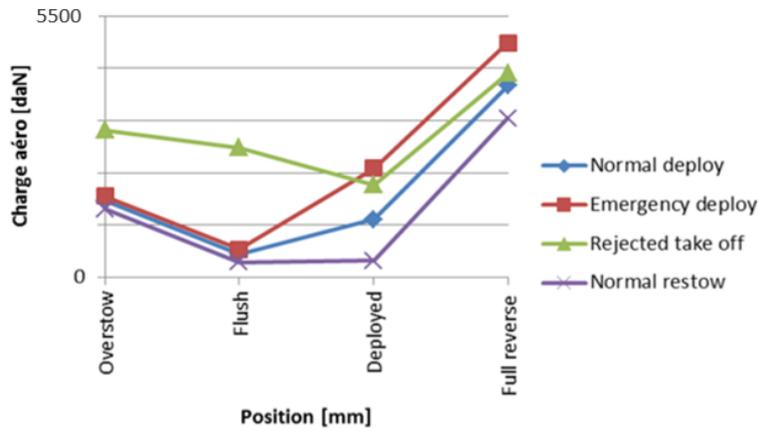


FIGURE 3.8 – Allures de la charge aérodynamique

Dans ce qui suit, nous avons considéré le cas du Normal deploy.

On adoptera les notations suivantes pour exprimer le comportement et la réponse du système dans les différentes phases de déploiement de l'actionneur :

1. Phase [overstow :Flush] : (i=1)

$$\begin{cases} \dot{x}(t) = A_{\alpha_1}x + B.u(t) \\ y(t) = Cx(t) \end{cases} \quad (3.29)$$

2. Phase [Flush : Deployed] : (i=2)

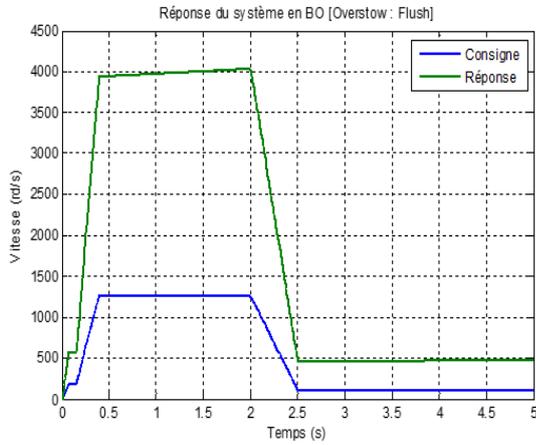
$$\begin{cases} \dot{x}(t) = A_{\alpha_2}x + B.u(t) \\ y(t) = Cx(t) \end{cases} \quad (3.30)$$

3. Phase [Deployed :Full reverse] : (i=3)

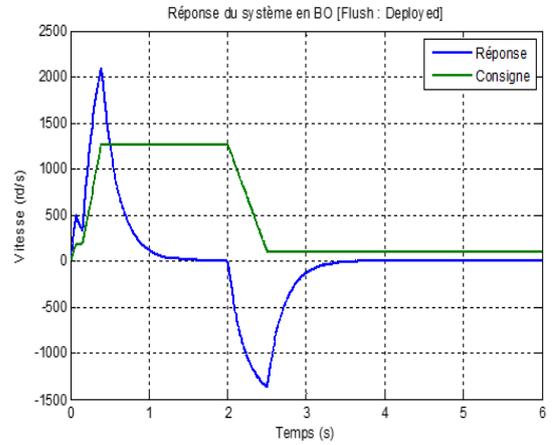
$$\begin{cases} \dot{x}(t) = A_{\alpha_3}x + B.u(t) \\ y(t) = Cx(t) \end{cases} \quad (3.31)$$

3.5.3 Résultat de simulation

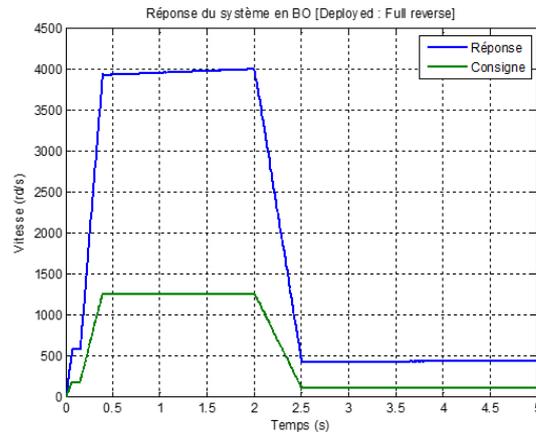
3.5.3.1 Simulation du système en boucle ouverte



– (a)



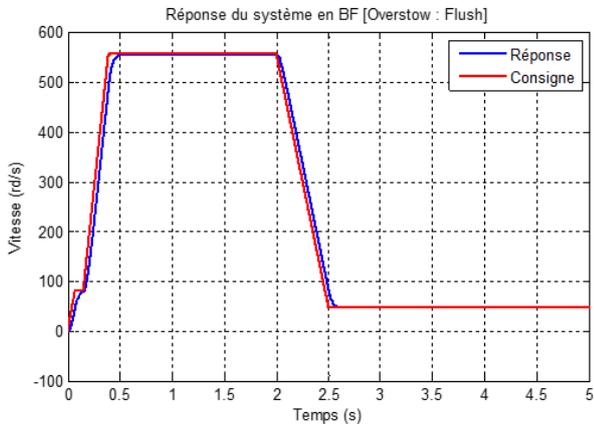
– (b)



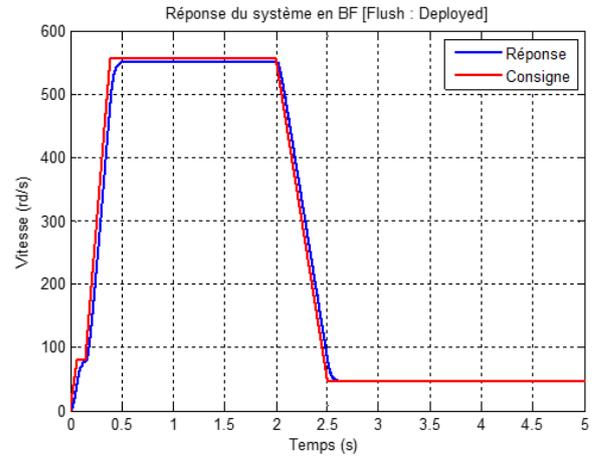
– (c)

FIGURE 3.9 – Réponse du système en BO : (a) [Overstow - Flush] (b) [Flush - Deployed]
(c) [Deployed - Full reverse]

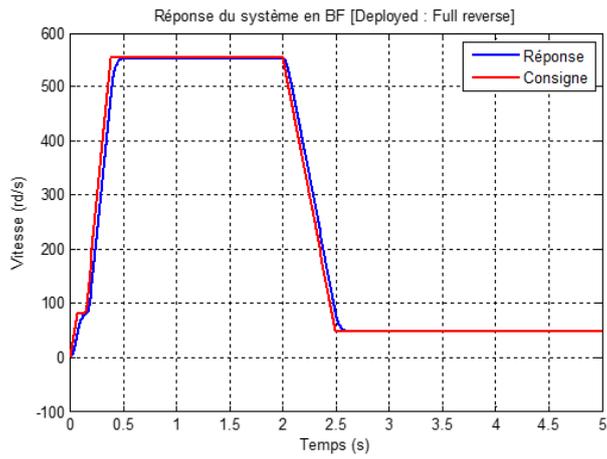
3.5.3.2 Simulation du système en BF sans perturbations en sortie



– (a)



– (b)



– (c)

FIGURE 3.10 – Réponse du système en BF : (a) [Overstow - Flush] (b) [Flush - Deployed]
(c) [Deployed - Full reverse]

3.5.3.3 Simulation du système en BF avec perturbations en sortie

La robustesse vis-à-vis des perturbations représente un atout majeur de la commande par H_∞ Loopshaping, pour démontrer cette dernière, nous avons considéré, en premier lieu, une perturbation de la forme 3.11 :

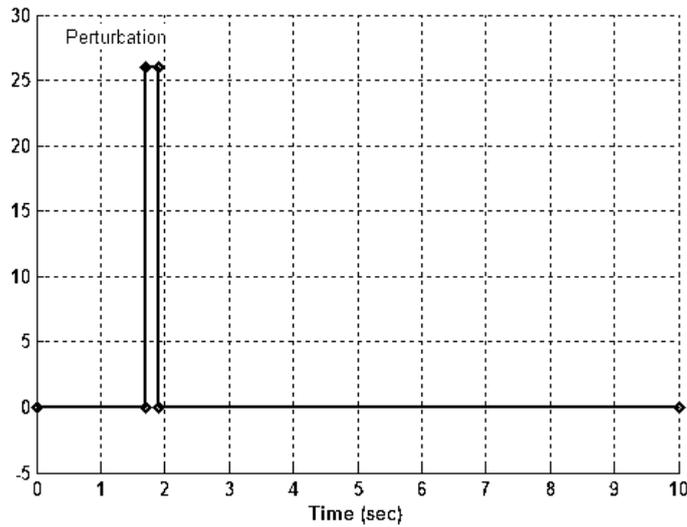


FIGURE 3.11 – La forme de la perturbation

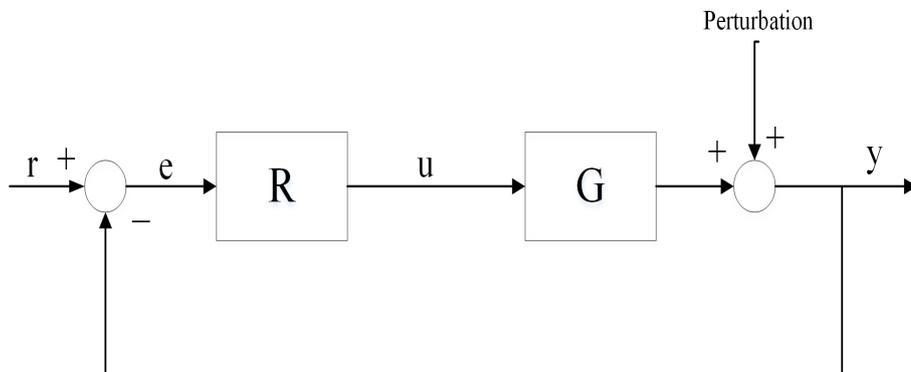
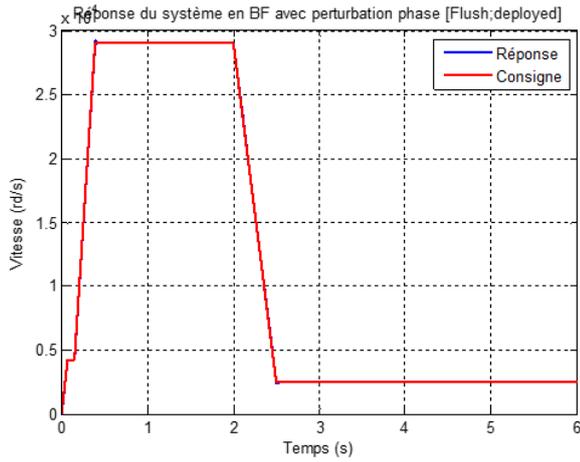
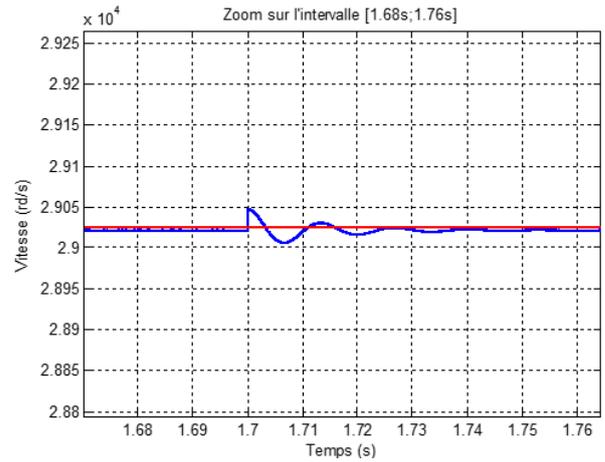


FIGURE 3.12 – Système avec erreurs de modélisation additives

Cette perturbation peut être matérialisé en pratique par un défaut capteur, où à l'instant $t = 1.7s$, la vitesse mesuré sera supérieure à sa valeur réelle.



– (a)



– (b)

FIGURE 3.13 – Réponse du système en BF avec perturbation phase (a) [Flush - Deployed] (b) [Flush - Deployed] avec zoom sur l'intervalle [1.68 : 1.76]s

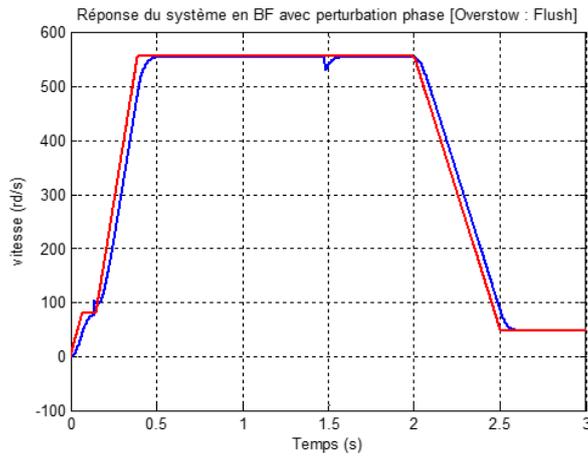


FIGURE 3.14 – Réponse du système en BF avec perturbation phase [Overstow : Flush]

Dans ce cas ,la perturbation considérée est de la forme :

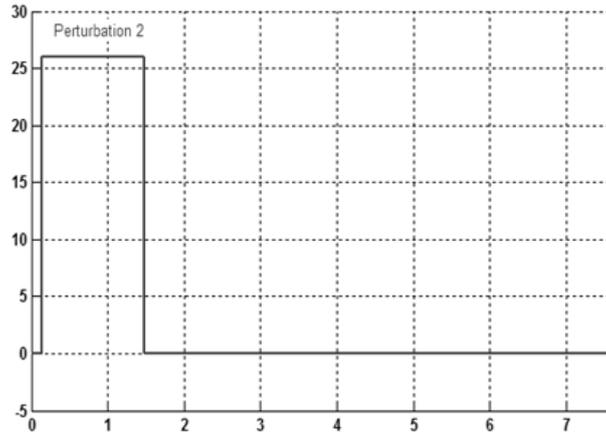


FIGURE 3.15 – La forme de la perturbation dans le 2ème essai

3.5.3.4 Simulation en présence d’erreurs de modélisation

Un problème récurrent survient lorsque l’on veut commander des systèmes complexes (exemple : aéronautiques). En effet, souvent, et malgré le fait que nous ayons pris le soin de modéliser le système en question (dans notre cas le système d’inverseurs de poussée ajouté au système de chargement hydraulique simulant l’effort aérodynamique) de la manière la plus fidèle possible, des erreurs de modélisation viennent ainsi se “ greffer ” au système étudié et altère ainsi la stratégie de commande adopté. En d’autres termes le système sera exprimé par :

$$G_{reel}(s) = G_N(s) + \Delta G(s) \quad (3.32)$$

$G_{reel}(s)$ au lieu de $G_N(s)$ comme considéré pour la synthèse. Par ailleurs, il existe dans la littérature (TAD11), plusieurs types d’erreurs de modélisation (additives, multiplicatives en entrée ou en sortie). Dans notre cas, nous avons choisi le premier type car mieux adapté pour représenter les variations paramétrique.

Ainsi et pour démontrer la robustesse de notre structure de commande $H_\infty(s)$, nous avons introduit ces erreurs de modélisation de tel sorte que :

$$| \Delta G(s) | < \sqrt{2} \quad (3.33)$$

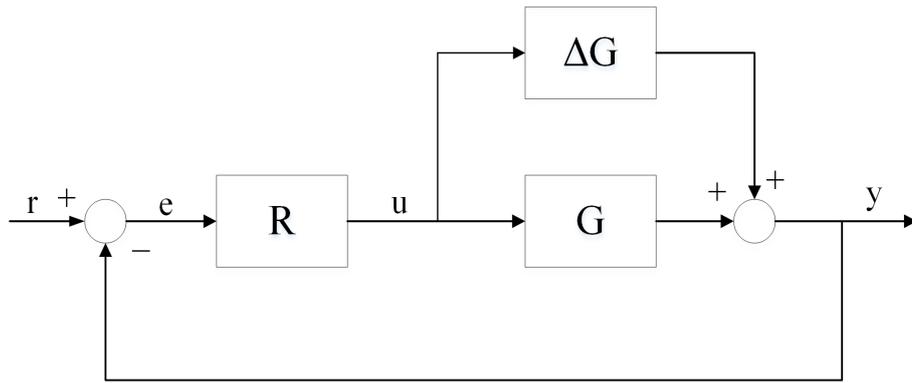


FIGURE 3.16 – Système avec erreurs de modélisation additives

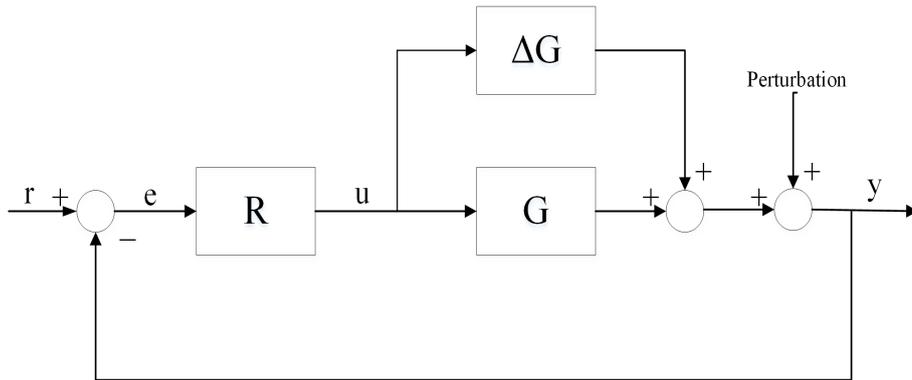
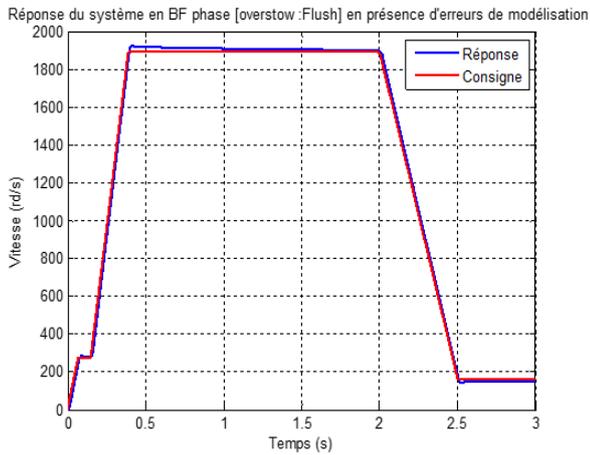
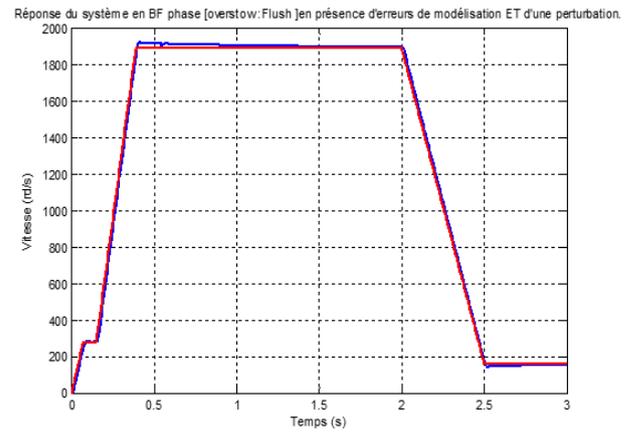


FIGURE 3.17 – Système avec erreurs de modélisation additives et une perturbation en sortie

Nous avons obtenu les résultats suivants :

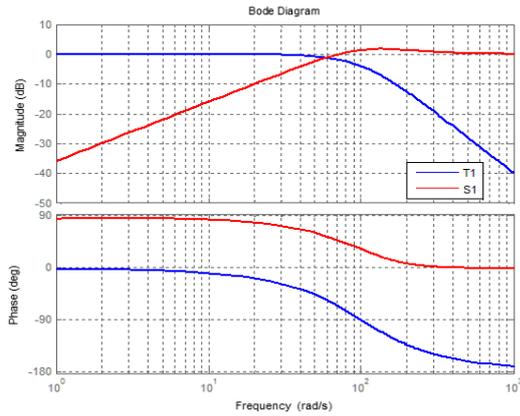


– (a)

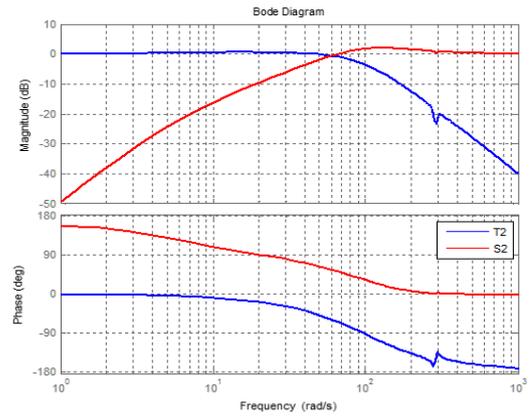


– (b)

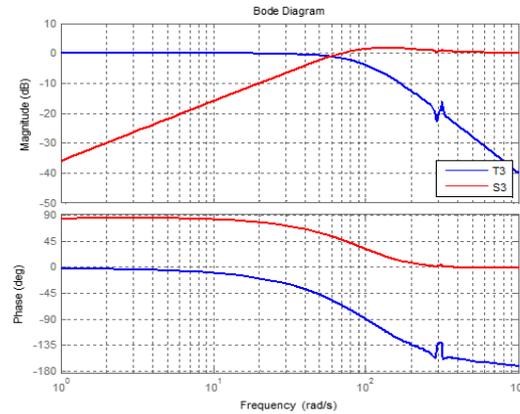
FIGURE 3.18 – Réponse du système en BF [Overtow - Flush] en présence : (a) d'erreurs de modélisation (b) d'erreurs de modélisation et d'une perturbation



– (a)



– (b)



– (c)

FIGURE 3.19 – Diagrammes de bode des fonctions de sensibilités et sensibilité complémentaire : (a) [Overstow - Flush] (b) [Flush - Deployed] (c) [Deployed - Full reverse]

3.6 Discussion

Bien que le système étudié soit stable 3.9 nous nous sommes heurtés à certaines difficultés lors de la synthèse de la loi de commande. En effet, un simple calcul de placement pôles s'est avéré impossible numériquement du fait que le modèle d'état développé (à 5 et même à 4 états) souffre d'un mauvais conditionnement, raison pour laquelle nous avons orienté nos recherches vers une autre commande : Le H_∞ Loopshaping.

On peut dire que la réponse du système en boucle fermée 3.10 répond aux attentes du cahier de charge imposé par le client de CERTIA. Pour pouvoir montrer la réelle plus-value qu'apporte la commande développée dans ce chapitre, nous avons imposé au système des contraintes assez importantes (erreurs de modélisation et perturbation en sortie). Les résultats ont été plus que concluants car en plus du fait que la perturbation inhérente au système (l'effort aérodynamique) est prise en charge par le système (ie inclut en 5^{ème} état dans la matrice A), ni des perturbations (inconnues mais bornées) en sortie ou même une erreur de modélisation importante n'arrive à désorienter le système de sa trajectoire référence.

Il est à signaler que les simulations effectuées ont été réalisées pour un "normal deploy", cependant, les résultats obtenus peuvent être extrapolés aux autres cas cités plus haut (Emergency deploy par exemple). En effet, la différence réside dans le profil de l'effort aérodynamique qui va changer 3.8.

Les trois dernières figures 3.19, montrent d'une part l'atténuation des erreurs de modélisation au moins $-40dB$ dans la plage fréquentielle $[1 : 100]rd/s$, et d'autre part le rejet de perturbation au-delà de $100rd/s$.

3.7 Conclusion

Dans ce chapitre, nous avons appliqué la commande par H_∞ Loopshaping sur le système d'inversion de poussée non sans modifier le modèle d'état considéré afin de rendre l'implémentation de la commande plus instinctive. Nous avons noté les avancées, en termes de performances que peut apporter ce type de commande notamment en termes de rejet de perturbation et de robustesse vis-à-vis des erreurs de modélisation.

Enfin, nous avons présenté les résultats donnés par la simulation des quantificateurs de performances, à savoir les fonctions de sensibilité et sensibilité complémentaire et ce afin de démontrer que dans la plage fréquentielle considérée, le système présente les performances attendues.

Commande par Backstepping

4.1 Introduction

L'objectif principal de ce chapitre est la conception d'une loi de commande stabilisante robuste. Pour cela nous allons faire appel à une technique de commande basée sur la théorie de Lyapunov "Backstepping".

La méthode de backstepping est une technique originale qui fournit un outil de conception très puissant (KRS95), elle permet la mise au point de lois de commande pour les systèmes non linéaires d'une manière récurrentes. Cette technique a été appliquée avec succès à une grande variété de systèmes linéaires et non linéaires, citons certains travaux portant sur la théorie de cette dernière (DAY85), (KAN91),(CHO11).

Pour réaliser ce travail, nous nous sommes fortement inspiré de l'application de la technique du Backstepping dans les systèmes d'actionnement électropneumatiques (SMA06), (SMA06) et(TUR10).

La première partie de ce chapitre, est dédiée essentiellement à des rappels théoriques. La deuxième partie est consacrée à la synthèse du contrôleur dans le cadre d'un suivi de trajectoire en vitesse de rotation. La dernière partie est destinée à la validation de la structure de commande.

4.2 Rappels et définitions d'automatique

4.2.1 Notion de stabilité au sens de Lyapunov

La théorie de la stabilité joue un rôle important en théorie des systèmes dynamiques. Nous entendons par stabilité du système, la stabilité des points d'équilibre (SAO06). La stabilité d'un point d'équilibre est généralement étudiée à l'aide du concept de stabilité au sens de Lyapunov, cette dernière est employée pour étudier le comportement des systèmes dynamiques soumis à des perturbations initiales autour du point d'équilibre.

Par définition, si un système est dans un état d'équilibre, il restera dans cet état pour t variant dans le temps. L'objet de la théorie de la stabilité est de tirer des conclusions quant au comportement du système sans calculer explicitement ses trajectoires. Le premier à avoir formulé mathématiquement cette idée est Lyapunov à la fin du 19e siècle. Son nom y est depuis associé, que ce soit pour s'y référer — “stable au sens de Lyapunov” — (YEG06). Pour permettre d'étudier mathématiquement la stabilité d'un point d'équilibre, Lyapunov généralise la notion d'énergie et regarde l'évolution de cette dernière. En 1963, J. Kurzweil a démontré l'équivalence entre l'existence des fonctions de Lyapunov et la stabilité des équilibres des équations différentielles ordinaires continues (SAO06).

Le paragraphe suivant rappelle brièvement les éléments de théorie utilisés dans ce mémoire, concernant la stabilité d'un point d'équilibre au sens de Lyapunov.

4.2.2 Éléments de théorie

Considérons le système autonome de dimension finie décrit par équation différentielle vectorielle non-linéaire du premier ordre :

$$\dot{x} = f(x), x \in \mathbb{R}^n \quad (4.1)$$

où le champ de vecteur $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est considéré *Lipschitzien*¹ sur un domaine $\Omega \subset \mathbb{R}^n$ incluant l'origine ou globalement *Lipschitzien* sur \mathbb{R}^n .

Commençons à l'état initial $x(0)$. Supposons que x_e est un point d'équilibre du système (c'est-à-dire $f(x_e) = 0$).

1. La définition exacte d'une application lipschitzienne sera donnée dans l'annexe B.

Nous disons que le point d'équilibre est :

1. Stable, si pour chaque $\varepsilon > 0$ existe $\delta(\varepsilon) > 0$, tel que :

$$\|x(0) - x_e\| < \delta \Rightarrow \|x(t) - x_e\| < \varepsilon \text{ pour tout } t \geq 0$$

2. Asymptotiquement stable, s'il est stable et que de plus il existe $r > 0$, tel que :

$$\|x(0) - x_e\| < r \Rightarrow x(t) \rightarrow x_e \text{ lorsque } t \rightarrow \infty$$

3. Globalement asymptotiquement stable, s'il est asymptotiquement stable pour tous les états initiaux.
4. Instable, s'il n'est pas stable.

Lyapunov a proposé l'idée d'incorporer le vecteur d'état $x(t)$ dans une fonction scalaire $V(x)$ afin de quantifier la distance entre $x(t)$ et le point d'équilibre du système. $V(x)$ peut être considérée comme une représentation de l'énergie contenue dans le système. S'il possible de montrer que $V(x)$ est continue et strictement décroissante selon t , alors le système lui-même doit se déplacer vers l'équilibre (ESH05).

Définition 1 *Fonction définie positive*

Une fonction scalaire $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite localement définie positive dans le domaine Ω si :

1. $V(x) = 0$
2. $V(x) > 0, \forall x \in \Omega$ et $x \neq 0$

Définition 2 *Fonction définie négative*

Une fonction scalaire $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite localement définie négative dans le domaine Ω si :

1. $V(x) = 0$
2. $V(x) < 0, \forall x \in \Omega$ et $x \neq 0$

Définition 3 *Fonction propre*

Une fonction scalaire $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite propre si :

$$V(x) = 0 \rightarrow +\infty \text{ quand } \|x\| \rightarrow +\infty \quad V(x) < 0, \forall x \in \Omega \text{ et } x \neq 0$$

Corollaire 1 Prenons $x = 0$, un point d'équilibre pour (B.4), $V(x)$ est une fonction scalaire continue et différentiable en x définie tel que :

1. $V(x)$ est définie positive.
2. $V(x)$ est propre.

3. $\dot{V}(x)$ est définie négative.

alors $x = 0$ est Globalement asymptotiquement stable.

Theoreme 1 (Stabilité Exponentielle(KHA01)) Soit $x = 0$ le point d'équilibre de $\dot{x} = f(x)$ et $\mathbb{D} \in \mathbb{R}^n$ un domaine contenant $x = 0$. Soient $V : \mathbb{D} \rightarrow \mathbb{R}$ une fonction C^n et a, k_1, k_2, k_3 , des constantes strictement positives telles que $\forall x \in \mathbb{D}$:

$$k_1 \|x\|^a \leq V(x) \leq k_2 \|x\|^a, \dot{V}(x) \leq -k_3 \|x\|^a \quad (4.2)$$

Alors $x = 0$ est exponentiellement stable. Si $D = \mathbb{R}^n$ alors $x = 0$ est globalement exponentiellement stable.

4.3 Théorie du Backstepping

La théorie de Lyapunov a été pendant longtemps un outil important aussi bien pour la commande des systèmes linéaires mais aussi pour les systèmes non linéaire. Cependant, son utilisation dans la commande non linéaire a été entravée par les difficultés pour trouver une fonction de Lyapunov pour un système donné(ESH05). L'invention des outils constructifs de la commande non linéaire basée sur la théorie de Lyapunov, comme le Backstepping, a été donc reçue à bras ouverts par la communauté scientifique.

Pour la classe des systèmes non linéaire en cascade, la commande par Backstepping permet systématiquement et de façon séquentielle de déterminer la loi de commande adéquate pour le système considéré(JAR08), cela est rendu possible par un bon choix de la fonction de Lyapunov.

Bien que la forme du Backstepping puisse s'apparenter à une linéarisation exacte, cependant ,elle s'en différencie en deux points. En effet, une linéarisation exacte ne garantit en rien la stabilité du système non linéaire, puisque la commande est construite uniquement à partir des dérivées de Lie jusqu'à l'apparition de la sortie. De plus, la commande par backstepping permet de garder des non-linéarités utiles c'est-à-dire les termes du modèle qui contribuent à la stabilité du système et qu'ils seraient dommage voire préjudiciable de compenser (ARC11).

4.3.1 Principe de la technique du Backstepping

Considérons le système suivant :

$$\begin{cases} \dot{x} = f(x) + g(x)\xi \\ \dot{\xi} = u \end{cases} \quad (4.3)$$

où $x \in \mathbb{R}^n$, $\xi \in \mathbb{R}$ est le vecteur d'état et $u \in \mathbb{R}$ est l'entrée du système, f et g sont des fonctions non linéaires connues avec $f(0) = 0$ et $g(x) \neq 0 \in \mathbb{R}^n$.

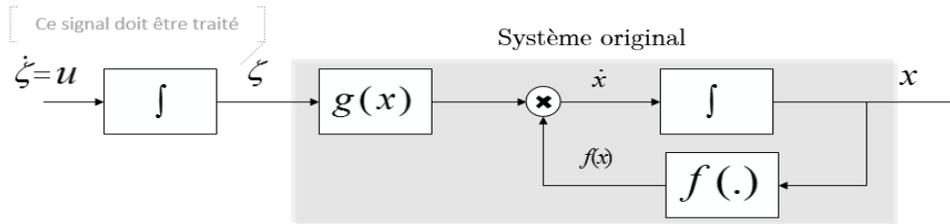


FIGURE 4.1 – Diagramme bloc du système (4.3)

Dans le système représenté ci-dessus, on peut contrôler x par ξ , si ξ était l'entrée actuelle du système, cependant, ce n'est pas le cas. De plus, il est possible de contrôler ξ par u , l'idée est donc d'utiliser le backstepping pour combiner les deux contrôleurs.

Pour stabiliser le système (4.3) On procède de la façon suivante, on ajoute et on retranche $g(x)\phi(x)$. On obtient le système équivalent :

$$\begin{cases} \dot{x} = f(x) + g(x)\phi(x) + g(x)(\xi - \phi(x)) \\ \dot{\xi} = u \end{cases} \quad (4.4)$$

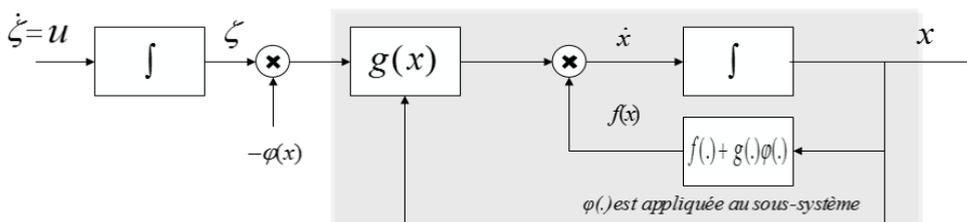


FIGURE 4.2 – Diagramme du système équivalent (4.4)

La diagramme ci-dessus montre clairement que si $\xi = \phi(x)$ alors l'erreur entre ξ et ϕ est nulle. Ainsi, avec une telle technique ϕ sera appliqué directement au sous-système en question. Soit le changement de variable :

$$z = \xi - \phi(x) \quad (4.5)$$

La dérivée de lie de z est donnée par :

$$\dot{z} = \dot{\xi} - \dot{\phi}(x) \quad (4.6)$$

On a par ailleurs

$$\dot{\xi} = u \quad (4.7)$$

et

$$\dot{\phi}(x) = \frac{\partial \phi}{\partial x} \dot{x} \quad (4.8)$$

En substituant l'expression (4.3) dans l'équation (4.8), on obtient alors :

$$\dot{\phi}(x) = \frac{\partial \phi}{\partial x} (f(x) + g(x)\xi) \quad (4.9)$$

IMAGE1

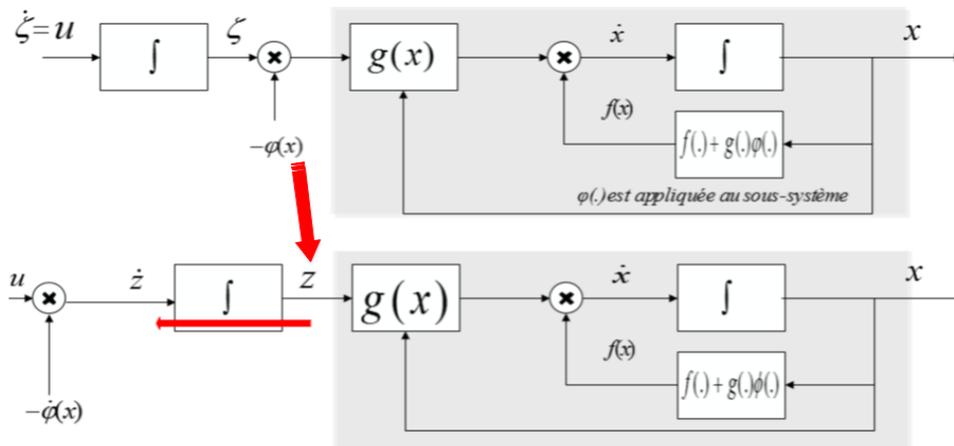


FIGURE 4.3 – Backstepping de $-\varphi(x)$ à travers l'intégrateur

La technique présentée ci-haut est en fait un Backstepping de ϕ à travers l'intégrateur.

En notant que :

$$v = \dot{z} \quad (4.10)$$

le système (4.4) s'exprime finalement comme suit :

$$\begin{cases} \dot{x} = f(x) + g(x)\phi(x) + g(x)z \\ \dot{z} = v \end{cases} \quad (4.11)$$

Le système (4.11) est équivalent au système (4.3).

A partir du système une fonction de Lyapunov est donnée par :

$$V = V(x, z) = V_1(x) + \frac{1}{2}z^2 \quad (4.12)$$

La dérivée de Lie de $V(x, z)$ le long du système s'exprime alors :

$$\dot{V} = \left(\frac{\partial V_1}{\partial x} (f(x) + g(x)\phi(x) + g(x)z) \right) + \dot{z}z \quad (4.13)$$

En remplaçant l'équation (4.10) dans l'expression (4.13), la dérivée de V s'exprime par :

$$\dot{V} = \frac{\partial V_1}{\partial x} f(x) + \frac{\partial V_1}{\partial x} g(x)\phi(x) + \frac{\partial V_1}{\partial x} g(x)z + zv \quad (4.14)$$

De même, pour assurer que \dot{V} est définie négative, nous posons :

$$v = - \left(\frac{\partial V_1}{\partial x} g(x) + kz \right), k \geq 0 \quad (4.15)$$

En substituant l'expression (4.15) dans l'équation (4.14), on obtient :

$$\dot{V} = \frac{\partial V_1}{\partial x} f(x) + \frac{\partial V_1}{\partial x} g(x)\phi(x) - kz^2 \quad (4.16)$$

Ce qui conduit au résultat suivant :

$$\dot{V} = \left(\frac{\partial V_1}{\partial x} f(x) + g(x)\phi(x) \right) - kz^2 \quad (4.17)$$

Il s'en suit que :

$$\dot{V} \leq -L(x) - kz^2 \quad (4.18)$$

Ce qui implique que l'origine du système (4.18) $\begin{pmatrix} x = 0 \\ z = 0 \end{pmatrix}$ est globalement asymptotiquement stable.

Finalement, à partir de (4.5), et on en déduit que l'origine de (4.3) $\begin{pmatrix} x = 0 \\ \xi = 0 \end{pmatrix}$ est asymptotiquement stable.

4.3.2 Système sous forme STRICT FEEDBACK

4.3.2.1 Système monodimensionnels sous forme STRICT FEEDBACK

Les modèles monodimensionnels sous forme STRICT FEEDBACK ont une structure récursive en cascade de taille croissante triangulaire inférieure. Ils sont appelés aussi boucle de retour stricte, c'est-à-dire la dérivée de chaque composante du vecteur d'état doit être une

fonction des composantes précédentes et dépendre additivement de la composante suivante (GUE12) :

$$\begin{cases} x_i(t) = f_i(x_1(t), \dots, x_i(t)) + g_i(x_1(t), \dots, x_i(t))x_{i+1}(t) \forall i \in \{1, \dots, n-1\} \\ x_n(t) = f_n(x_1(t), \dots, x_n(t)) + g_n(x_1(t), \dots, x_n(t))u(t) \\ y(t) = h(x_1(t)) \end{cases} \quad (4.19)$$

où $u(t)$ et $y \in \mathbb{R}$ représentent respectivement la commande et la sortie du système , $x(t) \in \mathbb{R}^n$ représentent les états du système.

f_i et g_i des fonctions non linéaires $\forall i \in \{1, \dots, n\}$, nous admettrons que ces fonctions ainsi que $h(x_1(t))$ sont suffisamment dérivable.

4.3.2.2 Système multidimensionnels sous forme STRICT FEEDBACK

Les modèles multidimensionnels s'écrit sous la forme strict Feedback comme suit :

$$\begin{cases} x_i(t) = f_i(x_1(t), \dots, x_i(t)) + g_i(x_1(t), \dots, x_i(t))x_{i+1}(t) \quad \forall i \in \{1, \dots, n\} \\ x_{n+i}(t) = f_n(x_1(t), \dots, x_{n+m}(t)) + \sum_{j=1}^{m-i+1} g_{n+i,j}(x_1(t), \dots, x_{n+m}(t))u_j(t) \quad \forall i \in \{1, \dots, m\} \\ y_1(t) = h_1(x_1(t)) \\ y_i(t) = h_i(x_1(t), \dots, x_i(t)) \quad \forall i \in \{2, \dots, m\} \end{cases} \quad (4.20)$$

où $u(t)$ et $y \in \mathbb{R}^m$ représentent respectivement les commandes et les sorties du système , $x(t) \in \mathbb{R}^n$ représentent les états du système.

f_i et g_i des fonctions non linéaires $\forall i \in \{1, \dots, n\}$, nous admettrons que ces fonctions ainsi que $h_i(x(t)) \forall i \in \{1, \dots, m\}$ sont suffisamment dérivable.

4.3.3 Commande en poursuite de trajectoire référence par Backstepping

4.3.3.1 Modèle sans incertitudes

L'objectif principal de ce travail est d'exploiter la technique du Backstepping pour la synthèse et la mise en œuvre d'une commande du moteur électrique, dans le but d'assurer un bon suivi de trajectoire de la sortie y du système d'inverseur de poussée 4.4 .Une base de l'erreur $e_i, \forall i \in \{1, \dots, n\}$ est définie comme étant la différence entre tous les états du système et leurs états désirés x_{id} (TUR10).

$$e_i = x_i - x_{id}, \forall i \in \{1, \dots, n\} \quad (4.21)$$

Considérons $y_d(t)$, au moins C^n ², la trajectoire désirée, h fonction bijective et h^{-1} est C^n .

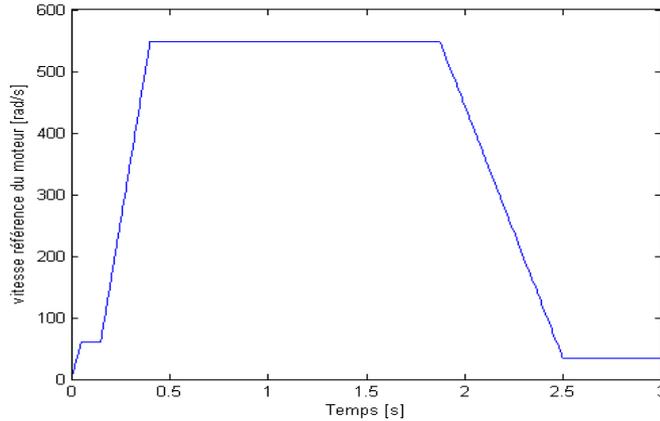


FIGURE 4.4 – Vitesse référence du moteur

Il est alors possible de construire un "Générateur de trajectoire" pour les états du système $x_i, \forall i \in \{2, \dots, n\}$ à partir de la trajectoire désirée. Le générateur de trajectoire peut alors se mettre sous la forme (TUR10) :

$$\begin{cases} x_{1d}(t) = h^{-1}(y_d) \\ x_{2d}(t) = \frac{1}{g_1}(-f_1 + \dot{x}_{1d} - k_1 e_1) \\ x_{(i+1)d}(t) = \frac{1}{g_i}(-f_i + \dot{x}_{id} - g_{i-1}e_{i-1} - k_i e_i) \quad \forall i \in \{2, \dots, n-1\} \end{cases} \quad (4.22)$$

La loi de commande stabilisante pour le système est alors donnée par :

$$u = \frac{1}{g_n}(-f_n + \dot{x}_{nd} - g_{n-1}e_{n-1} - k_n e_n) \quad (4.23)$$

Les expressions (4.23) et (4.22) imposent une forme particulière que doit prendre les erreurs³ :

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_i \\ \vdots \\ \dot{e}_n \end{pmatrix} = \begin{pmatrix} -k_1 & g_1 & 0 & \dots & 0 \\ -g_1 & -k_2 & g_2 & & \\ 0 & \ddots & \ddots & & \\ & & -g_{i-1} - k_i & g_i & \vdots \\ & \ddots & \ddots & & g_{n-1} \\ 0 & & \dots & -g_{(n-1)} & -k_n \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix} \quad (4.24)$$

2. Continuellement dérivable

3. L'annexe B présente une démonstration détaillée sur l'unicité du point d'équilibre à l'origine ainsi que la stabilité exponentielle globale du point d'équilibre.

4.3.3.2 Modèle avec incertitudes

L'objectif de ce paragraphe est l'élaboration d'une lois de commande robustes. Pour cela, des d'incertitudes qui peuvent affecter les fonctions non linéaires du système 4.19 sont prises en compte.

Ces incertitudes sont considérées sous la forme suivante :

$$\begin{cases} f_i = f_{iN} + \Delta f_i & \forall i \in 1, \dots, n \\ g_i = g_{iN} + \Delta g_i & \forall i \in 1, \dots, n \end{cases} \quad (4.25)$$

avec f_{iN} et g_{iN} sont les valeurs nominales connues $\forall i \in 1, \dots, n$ et les Δf_i et Δg_i sont des incertitudes considérées bornées peuvent être mises sous la forme :

$$\begin{cases} \Delta_i = \Delta f_i + \Delta g_i x_{i+1} & \forall i \in 1, \dots, n-1 \\ \Delta_n = \Delta f_n + \Delta g_n u \end{cases} \quad (4.26)$$

avec $|\Delta| \leq \Delta_{i_{max}} \forall i \in 1, \dots, n$ où les $\Delta_{i_{max}}$ sont les bornes connues des incertitudes $\forall i \in 1, \dots, n$.

Les expressions analytiques de générateur de trajectoire et de la commande dans ce cas, ne comporte que des dynamiques connues, sont données par :

$$\begin{cases} x_{1d}(t) = h^{-1}(y_d) \\ x_{2d}(t) = \frac{1}{g_1}(-f_1 + \dot{x}_{1d} - k_1 e_1) \\ x_{(i+1)d}(t) = \frac{1}{g_i}(-f_i + \dot{x}_{id} - g_{i-1} e_{i-1} - k_i e_i) \quad \forall i \in \{2, \dots, n-1\} \end{cases} \quad (4.27)$$

et

$$u = \frac{1}{g_{nN}}(-f_{nN} + \dot{x}_{nd} - g_{n-1,N} e_{n-1} - k_n e_n) \quad (4.28)$$

Dans ce cas, la base d'erreur B.2 devient (TUR10) :

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_i \\ \vdots \\ \dot{e}_n \end{pmatrix} = \begin{pmatrix} -k_1 & g_{1N} & 0 & \dots & 0 \\ -g_1 & -k_2 & g_{2N} & & \\ 0 & \ddots & \ddots & & \\ & & -g_{(i-1)N} - k_i & g_{iN} & \vdots \\ & & \ddots & \ddots & g_{(n-1)N} \\ 0 & & \dots & -g_{(n-1)N} & -k_n \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix} + \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_i \\ \vdots \\ \Delta_n \end{pmatrix} \quad (4.29)$$

Contrairement à ce qui a été fait dans la partie poursuite de trajectoire référence sans incertitudes, nous notons ici que le point $e_1 = e_2 = \dots e_i = \dots e_n = 0$ n'est pas un point d'équilibre. Cependant, une stabilité au sens entrée bornée /état borné peut être établie⁴.

4.4 Application au système d'inversion de poussée

4.4.1 Conception et application de la commande par backstepping

Avant de construire une loi de commande type Backstepping, il est nécessaire de transformer la représentation d'état issue du Bond Graph pour obtenir une forme triangulaire du système. Ainsi, deux méthodes peuvent être envisagées, la première consiste à considérer la charge aérodynamique comme deuxième entrée de commande, ce qui conduit à un système multidimensionnelle de la forme (4.20), la seconde méthode serait d'augmenter l'ordre du système en considérant la charge aérodynamique comme variable d'état, le système résultant est alors monodimensionnel de la forme (4.19). Dans cette partie nous avons choisi la 1^{ère} approche que nous avons jugé plus adapté pour la commande par Backstepping.

Après une inversion de l'ordre des variables d'états du modèle, la nouvelle représentation sous forme triangulaire (4.19) sera donnée par :

$$\begin{cases} \dot{x}(t) = f(x_b(t), l_{tube}(t)) + B \cdot u(t) \\ y(t) = Cx(t) \end{cases} \quad (4.30)$$

avec :

$$f(x_b(t), l_{tube}(t)) = \begin{pmatrix} -\frac{F_{porte}}{J_{porte}} x_{1b}(t) + BL(l_{tube})\psi(l_{tube}) x_{2b}(t) \\ -\frac{1}{J_{porte}} BL(l_{tube}) x_{1b}(t) + \frac{PV}{NJ_m} x_{3b}(t) \\ -\frac{PV}{N}\psi(l_{tube}) x_{2b}(t) - \frac{F_m + \frac{F_{vis}}{N^2}}{NJ_m} x_{3b}(t) + \frac{K_{em}}{2L_a} x_{4b}(t) \\ -\frac{K_{em}}{2L_a} x_{3b}(t) - \frac{R_a}{2L_a} x_{4b}(t) \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad C = \begin{pmatrix} 0 & 0 & \frac{1}{J_m} & 0 \end{pmatrix}$$

4. L'annexe B présente une démonstration détaillée sur stabilité entrée-état "ISS" Input-to-State Stability.

:

$$Et \quad \left\{ \begin{array}{l} f_1(x_1(t)) = -\frac{F_{porte}}{J_{portr}}x_1(t) \\ g_1(x_2(t)) = BL(l_{tube}(t))\psi(l_{tube}(t)) \\ f_2(x_1(t), x_2(t)) = -\frac{1}{J_{portr}}BL(l_{tube}(t))x_1(t) \\ g_2(x_1(t), x_2(t)) = \frac{PV}{NJ_m} \\ f_3(x_1(t), x_2(t), x_3(t)) = -\psi(t)x_2(t) - \frac{F_m + F_{vis}}{J_m}x_3(t) \\ g_3(x_1(t), x_2(t), x_3(t)) = \frac{K_{em}}{2L_a} \\ f_4(x_1(t), x_2(t), x_3(t), x_4(t)) = -\frac{K_{em}}{J_m}x_3(t) - \frac{R_a}{2L_a}x_4(t) \\ g_4(x_1(t), x_2(t), x_3(t), x_4(t)) = 1 \end{array} \right.$$

La loi de contrôle finale u est alors donnée par :

$$u = \frac{K_{em}}{J_m}x_3(t) + \frac{R_a}{2L_a}x_4(t) + \dot{x}_{4d}(t) - \frac{K_{em}}{2L_a}e_3 - k_4e_4 \quad (4.31)$$

4.4.2 Simulation et discussion

Cette partie est dédiée à la présentation, l'analyse ainsi que la comparaison des résultats de simulation. Dans un premier temps, nous illustrons la poursuite des états vers les trajectoires désirées construit par le générateur de trajectoire décrit par (4.22). Par la suite, une analyse des résultats donnés par la méthode du Backstepping sera présentée. Enfin une étude comparative entre la loi de commande classique et celle de backstepping sera effectuée.

L'implantation des lois de commande sur l'actionneur électromécanique nécessite le réglage de quatre gains. Les quatre gains correspondants à la loi de commande en position sont :

$$k_1 = 1$$

$$k_2 = 2$$

$$k_3 = 2$$

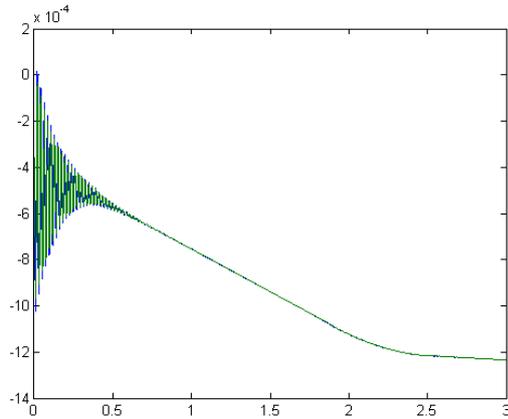
$$k_4 = 1$$

Satisfaisant les contraintes $k_1, k_2, k_3, k_4 > 0$

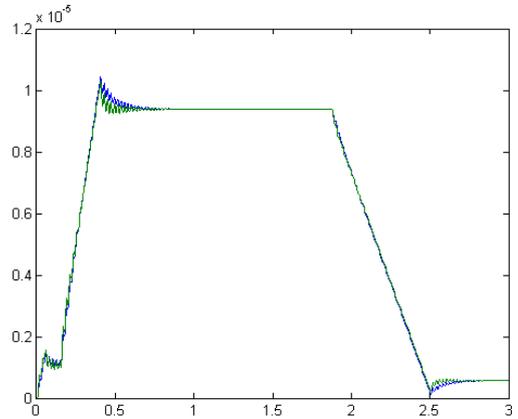
Toutes les simulations ont été réalisées à l'aide de Matlab/Simulink. Les essais sont effectués avec conditions initiales nulles. Ainsi, le profil de la charge aérodynamique correspond uniquement aux phases Flush et Deployed.

4.4.2.1 Modèle sans incertitudes

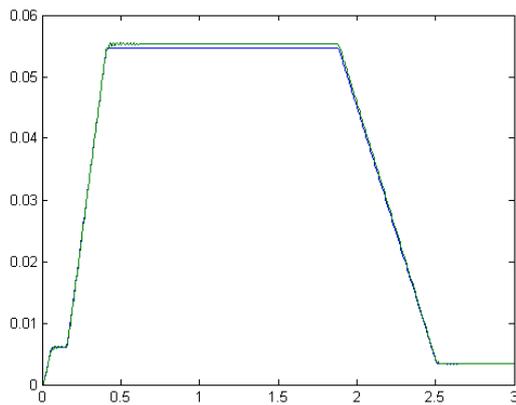
1. Poursuite des états



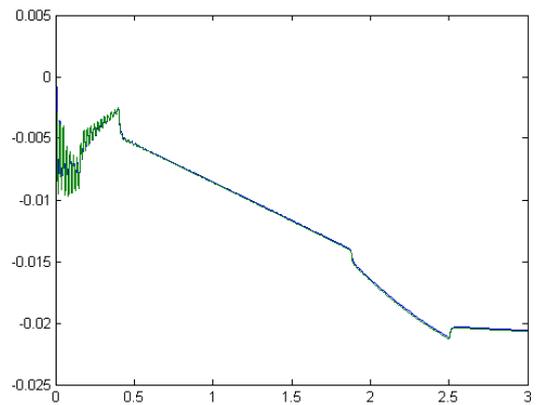
– (a)



– (b)



– (c)



– (d)

FIGURE 4.5 – Evolution des l'états :((a) x_1 et x_{1d} (b) x_2 et x_{2d} (c) x_3 et x_{3d} (d) x_4 et x_{4d}

Discussion Les figures ci-dessus 4.5 montrent une bonne convergence des valeurs mesurées vers les trajectoires désirées. Bien qu'il existe un léger écart, dû au choix des gains k_1, k_2, k_3, k_4 . Cependant, Les états du système suivent les trajectoires de références.

2. Suivi du profil de vitesse

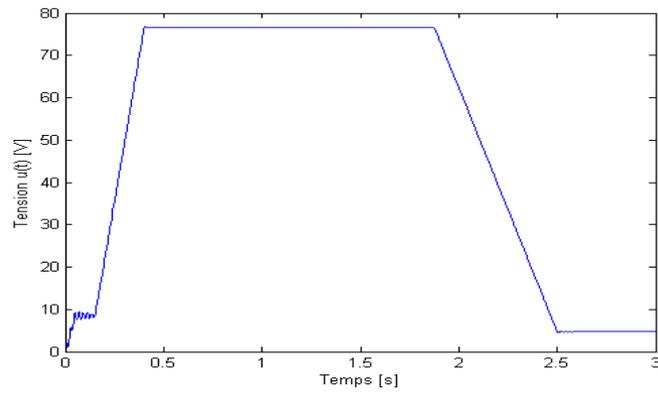


FIGURE 4.6 – Évolution du signal de commande

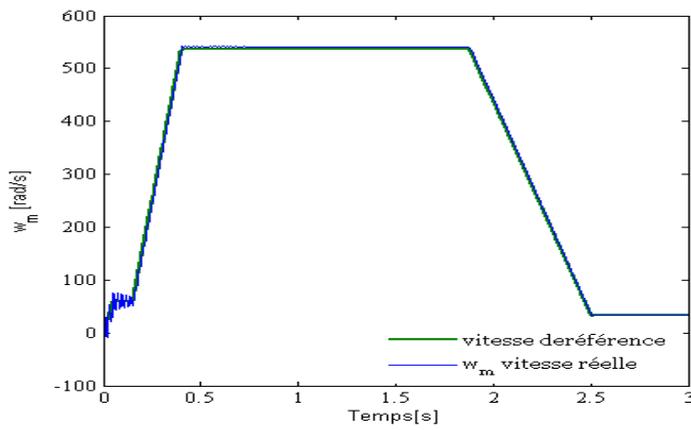


FIGURE 4.7 – Poursuite de vitesse de référence du moteur

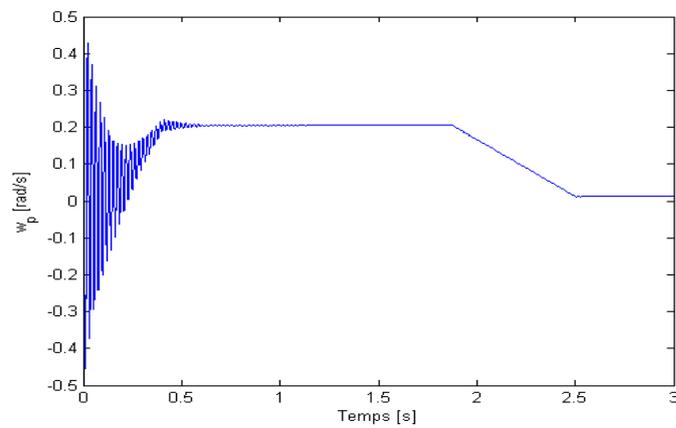


FIGURE 4.8 – La vitesse de la porte de l'inverseur de poussée

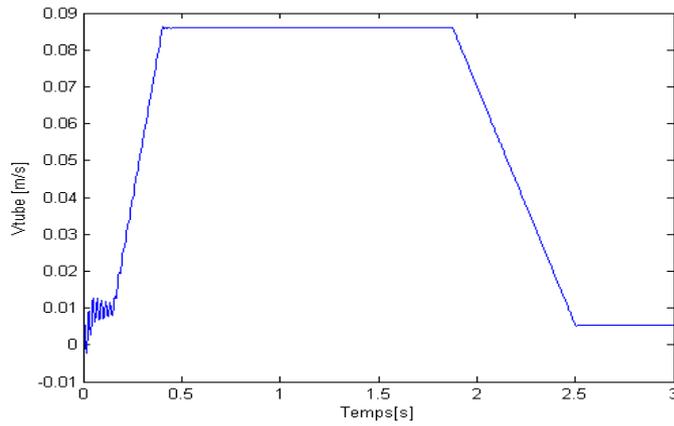


FIGURE 4.9 – La vitesse de l’allongement du tube

Discussion La commande par “backstepping en suivi de trajectoire sans incertitudes” nous a permis de garantir un suivi de trajectoire acceptable entre la référence et la vitesse réelle de moteur comme le montre la figure 4.7. D’autre part on remarque quelques oscillations lors du régime transitoire dans le profil de tension (commande) 4.6 et de vitesse du moteur 4.7, mais quand on atteint le régime permanent, ces oscillations disparaissent.

4.4.2.2 Modèle avec incertitudes

Dans le but de prouver la robustesse de la commande proposée, des incertitudes ont été ajoutées durant les tests. Les figures ci-dessous 4.10 et 4.11 présentent la vitesse mesurée du moteur et la vitesse de référence $w_{m_{ref}}$ en présence des incertitudes.

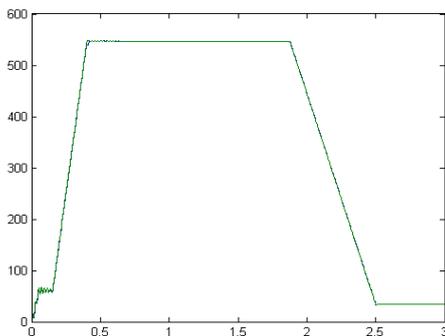
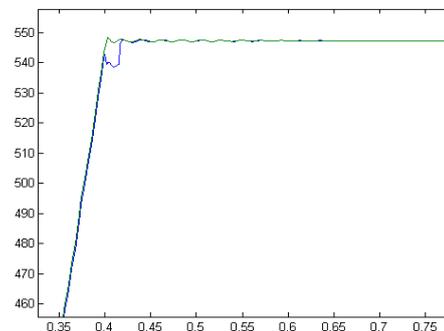


FIGURE 4.10 – Réponse du système en présence d’incertitudes

FIGURE 4.11 – Réponse du système en présence d’incertitudes à $t = 0.4$

et cela pour des incertitudes sur f_1 et f_3 de la forme :

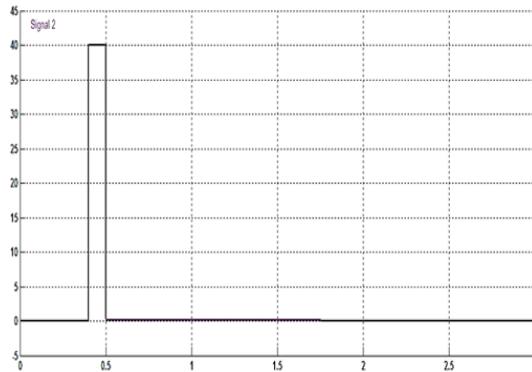


FIGURE 4.12 – les incertitudes sur les fonctions non linéaires f_1 et f_3

Discussion Les figure, présentées ci haut, montrent les résultats obtenus en présence d’erreurs de modèle. Cependant, malgré la présence d’incertitudes Δ_{f_1} et Δ_{f_3} au niveau de f_1 et f_3 respectivement, un bon suivi en vitesse a été obtenu.

4.5 Conclusion

Dans ce chapitre, nous avons traité la commande du système d’inversion de poussée dans le cas test “ PRESAGE ” par la technique du Backstepping en suivi de trajectoire. La synthèse de cette dernière a été construite grâce à l’utilisation d’un “générateur de trajectoire ”, garantissant la stabilité exponentielle du point d’équilibre du système dynamique d’erreurs en absence d’incertitudes. Pour ce qui est du cas où l’on prend en considération les incertitudes, il a été démontré que les écarts de trajectoires restent bornées à condition que les perturbations considérées le sont également.

Le chapitre suivant fera l’objet de la présentation d’une technique innovante dans le diagnostic des systèmes complexes, il s’agit de l’approche par Analyse en Composantes Principales.

Conclusion Générale

En abordant le travail qu'on nous a soumis, il fut clair que le but recherché fut une double contributions. En effet, d'une part aider au développement de la plateforme virtuelle de l'inverseur de poussée sous un progiciel (Dymola) et d'autre part apporter une réelle plus-value à la commande développée en amont et destinée à contrôler le système d'inversion de poussée dans le cas test " PRESAGE ". Cette dernière (commande) étant basée sur un PID simple, l'enjeu ne fut que plus important. En effet, il est connu que ce type de régulateur soit le plus (et le mieux) utilisée en industrie et ce pour sa grande efficacité prouvée jusque-là. Cependant, et comme dixit un grand journaliste et aviateur français Michel Polacco " Les enjeux de l'aéronautique demain sont : risque zéro ,moins de pollution avec un moindre cout ... ", c'est donc en toute logique que la recherche dans ce domaine se tourne vers des régulateurs ayant un plus fort potentiel de performance et de robustesse.

Dans le chapitre de l'élaboration de plateforme virtuelle sous Dymola, nous nous sommes attelées à adopter la méthodologie la plus optimale qui sera à même de nous garantir des résultats probants. En effet, tout en essayant d'apprendre à maîtriser le progiciel en question (Dymola/Modelica), nous avons travaillé en parallèle pour comprendre le fonctionnement du système multi-physique tout en s'inspirant de la modélisation effectuée sous AMESim. D'autre part, la validation des différents sous modèles a permis d'obtenir de très bon résultats pour la partie électromécanique et hydraulique. Ce dernier présente cependant sous Dymola quelques bémols qui seront à coup sûr effacés en approfondissant le développement et en ayant une meilleure maîtrise des différences existantes entre les deux progiciel (AMESim et Dymola). Nous ajouterons enfin que nous avons présentés les modèles développés sous Dymola à CERTIA ainsi qu'à DASSAULT SYSTEMES et ce pour avoir une vue d'ensemble de l'état d'avancement du projet " PRESAGE ⁵".

Pour ce qui concerne la partie commande, nous avons axé notre travail sur la recherche d'outils nous permettant d'aller vers plus de performances via des commandes présentant

5. Plateforme Réelle Et Simulée d'Actionnements Génériques et Evolutive

d'être robuste vis-à-vis des perturbation ainsi que des erreurs de modélisation. Pour ce qui est des résultats obtenus pour le Backstepping et le H_∞ Loopshaping et en les comparant aux précédent résultat obtenus avec un PID simple ,nous avons notés une nette amélioration des performances. En effet, bien que l'efficacité de cette dernière commande n'est en rien mise en cause, ses performances dans des environnements soumis à des incertitudes sont altérées. A fortiori, on peut envisager de pousser les recherches encore plus loin en intégrant un régulateur flou pour séquencer les différents régulateurs développées en H_∞ loopshaping en fonction des différentes phases et/ou configurations que peut rencontrer le système d'inversion de poussée. Aussi, pour ce qui est du Backstepping et la commande par trajectoire de référence, il serait judicieux comme perspective de développer d'autre commande du même type pour les différents cas que peut rencontrer le système (par exemple un décollage abandonnée).

En comparant les résultats obtenus, nous pouvons dire que le temps de réponse pour le Backstepping est légèrement meilleur que pour la méthode du H_∞ Loopshaping. Par contre, en termes de robustesse cette dernière s'est montré la plus adaptée au système considéré.

Pour ce qui de la partie où l'on a traité du diagnostic, nous nous sommes axés sur une méthode innovante, l'Analyse en Composantes Principales. Cette dernière, bien que jouissant d'une approche statistique à la base peut se montrer très intéressantes lors du diagnostic de notre système d'inversion de poussée. En effet, dans un environnement complexe ou l'éventualité même de considérer que le modèle est " certain " relève de l'hérésie intellectuelle, il est plus que judicieux de tenter une approche de diagnostic en se basant sur des données " saines " du système et non sur un modèle mathématique. Ce fut l'objet de la dernière partie de ce travail où après avoir introduit cette méthode dite " sans modèles ", nous avons expliqué la méthodologie à suivre lors du diagnostic de défauts sur notre système pour après présenter quelques résultats de simulation.

Comme perspectives, nous pouvons dire que cette dernière approche mérite plus d'attention et d'approfondissement car elle peut aux égards de son fort potentiel de robustesse, apporter une réelle plus-value dans le domaine du diagnostic des systèmes aéronautique.

Bibliographie

- [ABO03] H. Abou-Kandil, G. Freiling, G. Jank, et V. Ionescu, “*Matrix Riccati equations in control and systems theory*”, Birkhäuser, Suisse, 2003.
- [ABO04] H. Abou-Kandil et M. Jungers, “*Commande optimale multicritère : introduction à la théorie des jeux*, chapitre 5, Hermès, France, 2004.
- [AMM77] R.C. Ammer et H. Sowers, “*Thrust reverser design studies for an over-the-wing stol transport, General Electric company*”, NASA-CR-151958, NasaAmes Research Center, Mars 1977.
- [ARC11] L. Arcese. “*Modélisation et commande de microrobots magnétiquement guidés dans le système cardiovasculaire*”, Thèse de Doctorat, Discipline : Automatique et Traitement du Signal, Université d’Orléans, France, Novembre 2011.
- [ARZ02] D. Arzelier et D. Peacelle, “*An iterative method for mixed H_2/H synthesis via static output feedback* », *IEEE Conf Decision and Control*”, Las Vegas, USA, 2002.
- [ARZ04] D. Arzelier et D. Peacelle “*Multiobjective H/H impulse-to-peak synthesis : application to the control of an aerospace launcher*”, IFAC Symp Automatic Control in Aerospace (ACA), Saint-Petersbourg, Russie, 2004.
- [BEN11] O. Bennouna, N. Langlois, “*Commande tolérante aux fautes de l’inverseur de poussée d’une nacelle d’avion* ”, 4ème Workshop du Groupement d’Intérêt Scientifique ‘Surveillance, Sûreté ,Sécurité des Grands Systèmes’ GIS-3SGS’11, Valenciennes, Octobre 2011.
- [BLI99] L. Blin, “*Modélisation statistique et simulation des grandes échelles des écoulements turbulents application aux inverseurs de poussée*”, Thèse de Doctorat, Spécialité Mécanique des Fluides, Université De Rouen, France, Décembre 1999.
- [BOU10] M.Bouarroudj-Berkani, L.Dupont, “*Fatigue des composants électriques de puissance*”, Physique de défaillance, Technique de l’ingénieur d3126, Novembre 2010.

- [BOU11] M. S. Boucherit, “*Systèmes Asservis Linéaires Continus*”, Notes de cours de 3eme année Automatique, Ecole Nationale Polytechnique ,Alger ,2010-2011.
- [BRU02] D. Brück ,H. Elmqvist, S. Mattsson et H. Olsson, “*Dymola for Multi-Engineering Modeling and Simulation*”, 2nd International Modelica Conference, Proceedings, pp. 55-1 -55-8, Mars 2002.
- [BRU06] X. Brun, M. Smaoui et D. Thomasset, “*Systematic control of an electropneumatic system : integrator backstepping and sliding mode control*”, Transactions on Control System Technology, vol. 14, Issue 5, Sept.2006, p. 905- 913.
- [BUT04] J. Butterfield, H. Yao, M. Price, C. Armstrong, S. Raghunathan, E. Benard, R. Cooper et D. Monaghan , “*Enhancement of thrust reverser cascade performance using aerodynamic and structural integration*”, Queen’s University, Belfast, UK, Décembre 2004.
- [C6WP8] Projet : C6E2 WP2, “*État de l’art sur la conception des systèmes d’actionnement électromécaniques Cas test aéronautique MESSIER*”, Institut national des sciences appliquées de Toulouse et Le laboratoire de Génie Mécanique de Toulouse, Avril 2008 .
- [CAM10] L. Campbell, J. Chancelier et R.Nikoukhah, “*Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4*”, Springer, Springer New York Dordrecht Heidelberg London, Springer Science+Business Media, LLC 2010.
- [CHE11] A. Chebbi, “*Commande Backstepping d’une machine asynchrone sans capteur de vitesse*”, Thèse de Magister, Option : Commande, Université de Batna, Algérie, Février 2011.
- [CHI96] M. Chilali et P. Gahinet , “ *H_∞ design with pole placement constraint : an LMI approach* ». *IEEE Trans Automatic Control*”, Tome 41, n3 pages 358-367, 1996.
- [CHO06] J. Choley, “*Mécatronique : une nouvelle démarche de conception des systèmes complexes, LiSM Laboratoire d’Ingénierie des Systèmes Mécaniques et des matériaux*”, Revue : ‘Technologies Formations’ n 127 ,Paris ,France, 2006.
- [CHO11] A. Choukhou Braham, B. Cherki et M. Djemai, “*A backstepping procedure for a class of underactuated system with tree structure*”. In IEEE International Conference on Communications, Computing and Control Applications, CCCA’11. Hypersciences, Mars 2011.

- [CNR06] Le Centre national de ressources textuelles et lexicales (CNRTL), www.cnrtl.fr/definition/mach, France, février 2006.
- [COG03] S. Cogeze, “*Etude Logiciel CES4*”, mémoire de Licence en Technologie Mécanique, Spécialité : Mécanique, Université du Havre, France, Juin 2003.
- [DAS07] Dassault Systèmes, “*Bulletin des annonces légales*” obligatoires, Publications périodiques, Bulletin n 48, Décembre 2007.
- [DAS81] Dassault Systèmes, “www.3ds.com/fr/company”, éditeur de logiciels et filiale du Groupe industriel Marcel Dassault.
- [DAY85] W. Dayawansa, W. M. Boothby et D. L. Elliott, “*Global state and feedback equivalence of nonlinear systems*”, *Systems Control Letters*, vol. 6, pp.229 -234 ,1985.
- [DEL08] A. Delehelle, “*Étude d’un concept innovant d’un actionneur électromécanique linéaire à effet magnétique et piézoélectrique en vue d’applications dans le domaine des commandes de vol*”, Thèse De Doctorat, Spécialité : Génie Électrique, L’Institut National Polytechnique de Toulouse, Juin 2008.
- [DEM06] M. Dempsey, “*Dymola for Multi-Engineering Modeling and Simulation*”, In Proc. of IEEE-VPPC’06, 6p. Windsor (UK), September 2006.
- [DOY81] J.C Doyle et G. Stein, “*Multivariable Feedback Design : Concepts for a classical/modern Synthesis*”, *IEEE Trans Automatic Control* 26, pages 4-16, 1981.
- [DYN05] Dynasim, PRESS RELEASE, “*DaimlerChrysler, BMW, Audi and Volkswagen choose Dymola for modeling and simulation of air conditioning systems*”, Mars 2005.
- [EJJ10] K. Ejjabraoui, “*Contribution à la conception de systèmes mécatroniques automobiles :méthodologie de pré-dimensionnement multi-niveau multi-physique de convertisseurs statiques*”, Thèse de doctorat, Université Toulouse III - Paul Sabatier, France, Décembre 2010.
- [ELH08] B. El hefni et B. Bride, “*Modèles de fonctionnements thermo hydrauliques en régime permanent de centrales : Mira Fiori et biomasse Multi-configuration*”, présentation d’un projet d’EDF-RD (Électricité de France), La journée nationale des utilisateurs de logiciels dédiés la modélisation et au calcul scientifique (LMCS), France, Avril 2008.

- [ESH05] F. Eshbair, “*Modélisation et commande d’un système Multi-moteur par la technique de commande Backstepping*”, Mémoire Maîtrise en génie électrique, L’Université du Québec, Canada, Août 2005.
- [FIC04] P. Fichou, “*Bond graphs : une méthode pluridisciplinaire*”, Revues du CNDP, Technologie, Sciences et techniques industrielles, N133, Septembre-Octobre 2004.
- [FRI02] P. Fritzson, P. Aronsson, P. Bunuş, V. Engelson, L. Saldamli, H. Johanson et A. Karström, “*The open source Modelica project*”, pages 297–306, In Proc. of 2nd International Modelica Conference, 2002.
- [FRI03] P. Fritzson, “*Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*”, IEEE press, Wiley-interscience, 2003.
- [GAA12] S. Gaaloul Chouikh, “*Interopérabilité basée sur Les standards Modelica Et composant logiciel pour la simulation énergétique des systèmes de bâtiment*”, Thèse De Doctorat, Spécialité : Génie Electrique, L’Université De Grenoble, France, Octobre 2012.
- [GOO84] G.C Goodwin et K. Sang Sin, “*Adaptative Filtering Prediction and Control*”, Prentice-Hall, New Jersey, États-Unis, 1984.
- [GUE12] L. Guessas, “*Backstepping Backstepping adaptatif pour le contrôle la poursuite et la synchronisation des systèmes dynamiques non linéaires chaotiques*”, Spécialité : Electronique, Université Ferhat ABBAS, Algérie, 2012.
- [HAD08] H. Hadj-Amor, “*Contribution au prototypage virtuel de systèmes mécatroniques basé sur une architecture distribuée HLA. Expérimentation sous les environnements OpenModelica-OpenMASK*”, Spécialité : Sciences et techniques industrielles, Thèse De Doctorat, Université du Sud Toulon Var, France, Décembre 2008.
- [HAL04] S. Hall, E. Benard et S. Raghunathan, “*Progress in developing innovative flow control in a cascade thrust reverser*”, 24TH International Congress Of The Aeronautical Sciences, School of Aeronautical Engineering, Queen’s University Belfast, UK, 2004.
- [HAM12] M. Hammadi, “*Contribution à l’intégration de la modélisation et la simulation multi-physique pour la conception des systèmes mécatroniques*”, Thèse de doctorat de l’école centrale de Paris, Spécialité génie industrielle, Janvier 2012.
- [HAR03] M.F. Harkat, “*Détection et Localisation de Défauts par Analyse en Composantes Principales*”, Thèse de Doctorat, spécialité Automatique et Traitement numérique du signal, Institut National Polytechnique de Lorraine, France, Juin 2003.

- [JAN03] G. Jank et H. Abou-Kandil, “*Existence and uniqueness of open-loop Nash equilibria in linear-quadratic discrete time games*”, IEEE Trans. Autom. Control, tome 48, n 2, pages 267–271, 2003.
- [JAR08] Y. AI-Younes, M.A. Jarrah, “*Attitude stabilization of quadrotor UAV using backstepping fuzzy logic backstepping least-mean-square controllers*”, Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISMA08), Amman, Jordanie, Mai, 2008.
- [JAR10] A. Jardin, “*Contribution a une methodologie de dimensionnement des systemes mecatroniques : analyse structurelle et couplage à l’optimisation dynamique*”, Thèse de Doctorat, Spécialité : Énergie et Systèmes, l’Institut National des Sciences Appliquées de Lyon, France, Janvier, 2010.
- [JET08] C. Jettanasen, “*Modélisation par approche quadripolaire des courants de mode commun dans les associations convertisseurs-machines en aéronautique ; optimisation du filtrage*”, l’École Centrale De Lyon, France, Décembre 2008.
- [JMO09] “<http://www.jmodelica.org>” Copyright © 2009 Modelon AB.
- [KAN91] I. Kanellakopoulos, P. V. Kokotovic et R. Marino, “*An extended direct scheme for robust adaptive nonlinear control*”, Automatica, 1991.
- [KAR07] W. Karam, “*Générateurs de forces statiques et dynamiques à haute puissance en technologie électromécanique*”, Thèse De Doctorat, Spécialité : Mécanique, Université de Toulouse, France, 2007.
- [KHA01] H.K. Khalil, “*Nonlinear Systems*”, Systems, 3ème édition, Prentice Hall, 2001.
- [KRS95] M. Krstić, I. Kanellakopoulos, et P. Kokotovic. “*Nonlinear and Adaptive Control Design*”, Wiley-Interscience Publication, 1995.
- [LEC84] N.A. Lehtomaki, B.C. Levy et N.R. Sandell, “*Robustness and Modeling error characterization*”, IEEE Trans Automatic Control, 29(3), 1984.
- [LMS08] LMS FRANCE, “*LMS intègre la compatibilité avec Modelica dans sa plate-forme de simulation système LMS Imagine.Lab AMESim*”, Communiqués de presse du Syndicat de l’Instrumentation de Mesure, du Test et de la Conversion d’Energie dans le domaine de l’électronique – SIMTEC, Novembre 2008.
- [MAC89] J.M. Maciejowski, “*Multivariable Feedback Design*”, Anddisson-Wesley, Wokingham, U.K, 1989.

- [MAM10] D. Mami, “*Définition, conception et expérimentation de structures d’actionneurs électromécaniques innovants incluant par conception des fonctionnalités de sûreté et de sécurité de fonctionnement*”, Thèse De Doctorat, spécialité : Génie Électrique, L’Université de Toulouse, France, Janvier 2010.
- [MCF92] D.C. McFarlane et K.Glover , “ *A loopshaping design procedure using Hinf synthesis* ”, IEEE Trans Automatic Control, Tome 37, n6 pages 759-769, 1992.
- [MES12] M. Messaadi , “*Commande backstepping appliquée à la machine synchrone a aimants permanents*”, Mémoire de Magister, Discipline : Commande des systèmes électromagnétiques, Université Hadj Lakhdar, Batna,Algerie, Janvier 2012.
- [MES12] S. Narasimhan et L. Sirish, “*Model identification and error covariance matrix estimation from noisy data using PCA*”, In : IFAC ADCHEM. Hong Kong, Chine, Janvier 2004.
- [NFO06] G. Nfonguem, “*Contribution au développement d’actionneur plus électrique –Modélisation inverse et composants mécanique spécifiques a une application Aéronautique*”, Thèses de doctorat, Institut national des sciences appliquées de Toulouse, France, Janvier, 2006.
- [PAS11] A. Pascoal, “ *Loopshaping (SISO case)*”, Instituto Superior Tecnico, Portugaile, 2011.
- [PLA09] R. Plateauxa, O. Penasa, F. Mhennia, J-Y. Choleya, B. Roumizadeha et A. Rivièrea, “*Vers un environnement intégré pour le pré dimensionnement –Modelica 3D*”, 19ème Congrès Français de Mécanique, France, août 2009.
- [POS09] R. Postoyan, “*Commande et construction d’observateurs pour des systèmes non linéaires incertains a données échantillonnées et en réseau*”, Thèse De Doctorat, Spécialité : Physique, Université Paris-Sud 11, France, Novembre 2009.
- [QIN01] S.J. Qin et W. Li , “*Consistent dynamic pca based on errors-in-variables subspace identification*”, Journal of Process Control, 11(6) : 661–678, 2001.
- [RUE10] J.RUEL, “*Developpement de simulateurs des procédés pour la commande automatique et l’optimisation approche de conception hybride avec modelica et matlab/simulink*”, Mémoire dans le cadre du programme de maîtrise en génie électrique, Université Laval, Québec ,Canada, 2010.

- [SAC84] D. Sacquepey et D.Spenlé , “*Précis de construction mécanique*”, Tome3, Calculs technologie et normalisation, AFNOR, NATHAN, Paris, France, 1984.
- [SAF81] M.G Safonov, A.J Land et G.L Hartman, “*Feedback Properties of Multivariable Systems : The role and use of the return difference matrix*”, IEEE Trans Automatic Control, 26(1), 1981.
- [SAN07] H. Sande, T.A. Johansen, G.O. Kaasa, S.R.Snare et C.Bratli, “*Switched backstepping control of an electropneumatic clutch actuator using on/off valves*”, Proceedings of the 2007 American Control Conference, New York, USA, Juin 2007.
- [SAO06] H. Saoud , “*Étude des problèmes unilatéraux : Analyse de récession, stabilité de Lyapunov et applications en électronique et en mécanique*”, Thèse de Doctorat, Discipline : Mathématiques et ses applications, Département Maths-Info, Laboratoire XLIM, Université De Limoges, France, juin 2009.
- [SCH95] C. Scherer, “*Multi-objective output feedback control via LMI optimization*”, IEEE Trans Automatic Control, tome 40, n6, pages 1054-1062, 1995.
- [SID10] L. Sidhom, X. Brun, E. Bideaux et D. THomasset, “*Suivi de trajectoire en position d’un servo-vérin hydraulique via la technique de Backstepping*”, Université de Lyon – INSA de Lyon , CIFA 2010 Conférence Internationale Francophone d’Automatique, France, 2010.
- [SMA04] M. Smaoui, X. Brun et D. Thomasset, “*A robust multivariable control for an electropneumatic system using backstepping design*”, Symposium on Nonlinear Control Systems, NOLCOS 6th, IFAC symposium, Stuttgart, 1-3 Septembre 2004, p 1193-1198.
- [SMA06] M. Smaoui, X. Brun et D. Thomasset, “*A study on tracking position control of electropneumatic system using backstepping design*”. CEP, vol.14, pp 923-933, 2006.
- [STE87] G. Stein et M.Athaus , “*The LQG/LTR Procedure for Multivariable Feedback Control Design*”, IEEE Trans Automatic Control, Tome 32, n2, 1987.
- [TAD11] M. Tadjine, “*Commande des systèmes multivariables*”, notes de cours de 4eme année, Ecole Nationale Polytechnique, Alger, 2011.
- [THA08] Y. Tharrault, “*Diagnostic de fonctionnement par analyse en composantes principales : Application à une station de traitement des eaux usées*”, Thèse Doctorat, Spécialité Automatique et Traitement du signal, Institut National Polytechnique de Lorraine, Décembre 2008.

- [THA06] Y. Tharrault, G. Mourot et J. Ragot ,”Diagnostic de fonctionnement de capteurs d’un réseau de surveillance de la qualité de l’air par analyse en composantes principales itérative”, Centre de recherche en automatique de Nancy (CRAN), Avril 2006.
- [THM00] J. Thoma et B. Ould Bouamama, “*Modelling and simulation in thermal and chemical engineering : Bond graph Approach*”, Springer Verlag, 2000.
- [TUR10] K. Turki, “*Nouvelles approches pour la synthèse de lois de commande non linéaires robustes. Application à un actionneur électropneumatique et proposition d’une solution au problème du redécollage*”, Thèse De Doctorat, Formation Doctorale : Énergie des Systèmes, Institut National Des Sciences Appliquées De Lyon, France, Septembre 2010.
- [VER05] M. Verge, “*Modélisation pour l’ingénieur :approche par Bond Graph*”, 6 ème congrès Européen de Science des Systèmes, France, Septembre 2005.
- [YEG06] N. Yeganefar, “*Définitions et analyse de stabilités pour les systèmes à retard non linéaires*”, Thèse De Doctorat, Discipline : Automatique et Informatique Industrielle, École Centrale De Lille, France, Novembre 2006.
- [ZAU07] G. Zauner, D.l Leitner et F. Breitenecker, “*Modeling Structural - Dynamics Systems in MODELICA/Dymola, MODELICA/Mosilab and AnyLogic*”, 1st International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT07), pp : 99-110, Berlin-Germany, juillet 2007.

Modelica

A.1 Historique et généralités

Le langage Modelica a vu le jour en 1996, afin d'offrir à la communauté scientifique un langage conçu spécifiquement pour supporter la description du comportement de systèmes multi-physiques à l'aide d'équations différentielles et algébriques (RUE10).

Modelica est un langage orienté objet pour la modélisation des systèmes physiques complexes comprenant des composants électriques, mécaniques, hydrauliques et thermiques dans un but de simulation. Aussi, c'est un langage multi domaine et non causal (FRI02) dédié à des applications génériques. Le nombre de bibliothèques utilisant Modelica est en nette augmentation. Plusieurs outils commerciaux et Open source utilisent Modelica comme langage de prédilection pour le développement. De plus, il présente l'avantage de fédérer autour de lui une large communauté¹ scientifique très active. Cette communauté a permis la multiplication et la capitalisation des bibliothèques de modèles ainsi que des environnements de simulation. En outre, il offre la possibilité d'exploiter les concepts d'héritage et de profiter ainsi de la possibilité de réutilisation.

Ce langage est en voie de devenir une norme (RUE10) car il est supporté par un nombre grandissant d'environnements de développement :

1. **Libre :**

- OpenModelica project : Outil gratuit toujours en phase de développement par Linköping² University .

1. Modelica association

2. Linköping est le centre de l'industrie aéronautique suédoise

- SCICOS12 :Outil de modélisation et simulation des systèmes dynamiques développé par un groupe de chercheurs à INRIA ³ (CAM10),dans le cadre du projet METALAU ⁴ .
- JModelica :Une plateforme open source pour la modélisation, simulation et optimisation des Systèmes complexes en développement par Modelon AB (JMO09).

2. Commerciale :

- Dymola : Outil commercial développé par Dynasim ⁵,Dymola a été acquise par Dassault System. ⁶.
- MathModelica :Outil commercial développé par MathCore.
- MOSILAB : Outil commercial développé par Fraunhofer-Gesellschaft. ⁷
- LMS AMESim12 :Un logiciel de simulation pour la modélisation et l’analyse des systèmes multi-domaines.LMS ⁸ intègre progressivement le langage Modelica au logiciel LMS Imagine.Lab AMESim(?) .

A.2 Caractéristiques de Modelica

Les aspects les plus importants de Modelica sont :

1. Il est principalement basé sur des équations et non sur des instructions d’affectation. Ceci permet une modélisation qui favorise une meilleure réutilisation des classes puisque les équations ne spécifient pas de direction de flux de données.
2. Il a la capacité de fournir une modélisation multi domaine. Les modèles de composants correspondant aux objets physiques de plusieurs domaines différents, électrique, mécanique, thermodynamique, hydraulique, biologique et des applications de contrôle peuvent être décrits et connectés.
3. C’est un langage orienté objet avec un concept de classe générale qui englobe les classes. Ceci facilite la réutilisation des composants.
4. Il permet de créer des modèles de composants, avec des constructions pour connecter ces composants. Ainsi ce langage est adapté pour la description architectural des

3. Institut National de Recherche en Informatique et en Automatique ‘INRIA’ est un organisme de recherche français

4. METHodes, Algorithmes et Logiciels pour l’AUtomatique ‘METALAU’

5. Une société suédoise spécialisée dans les logiciels de modélisation et de simulation pour les systèmes embarqués.

6. Dassault Systèmes est une entreprise française, éditeur de logiciels.

7. Un organisme allemand spécialisé dans la recherche en sciences appliquées.

8. Société d’ingénierie qui travaille en partenariat avec des entreprises du secteur de l’automobile, de l’aéronautique et d’autres industries de fabrication de pointe

systèmes physiques complexes et dans une certaine mesure pour les systèmes logiciels.

A.3 Éléments du langage Modelica

Afin de comprendre les code Modelica présentés dans ce mémoire ,un certain nombre d'éléments du langage sont décrits ici :

Les classes

Comme tous les langages orienté objet, Modelica fournit la notion de classe et objet (HAD08). Une classe Modelica est structurée en blocs : bloc initial algorithm, bloc algorithm et bloc équation. Les membres d'une classe peuvent avoir deux niveaux de visibilité : publique (public) ou protégé (protected). Par défaut, un membre est public si rien n'est indiqué avant. Un membre public est accessible de partout et sans aucune restriction. L'autre niveau de visibilité est spécifié par le mot « protected ». Un membre protégé est accessible uniquement par le code au sein de la classe où il est déclaré aussi bien par le code dans les classes qui héritent de cette classe (HAD08).

Une classe définit les propriétés communes aux composants qui y sont rattachés :

i. Héritage

La modélisation orientée objet permet d'étendre le comportement et les propriétés d'une classe existante, c'est-à-dire la reprise de propriétés de classes existantes. Cette dernière est étendue pour obtenir une classe plus spécifique, appelée souvent classe fille ou sous-classe. Le comportement et les propriétés de la classe mère, décrit sous forme de variables, d'équations et d'autres contenus, sont hérités par la classe fille (HAD08). Pour qu'une classe fille puisse hériter des propriétés de la classe mère, il faut que les propriétés de la classe mère possèdent des attributs de visibilité.

Modelica permet de faire de l'héritage multiple, ce qui signifie qu'ils offre la possibilité de faire hériter une classe de deux superclasses.

Par ce moyen on crée une hiérarchie de classes de plus en plus spécialisées. De cette manière il est possible d'acheter dans le commerce des librairies de classes, qui constituent une base, pouvant être spécialisées à loisir.

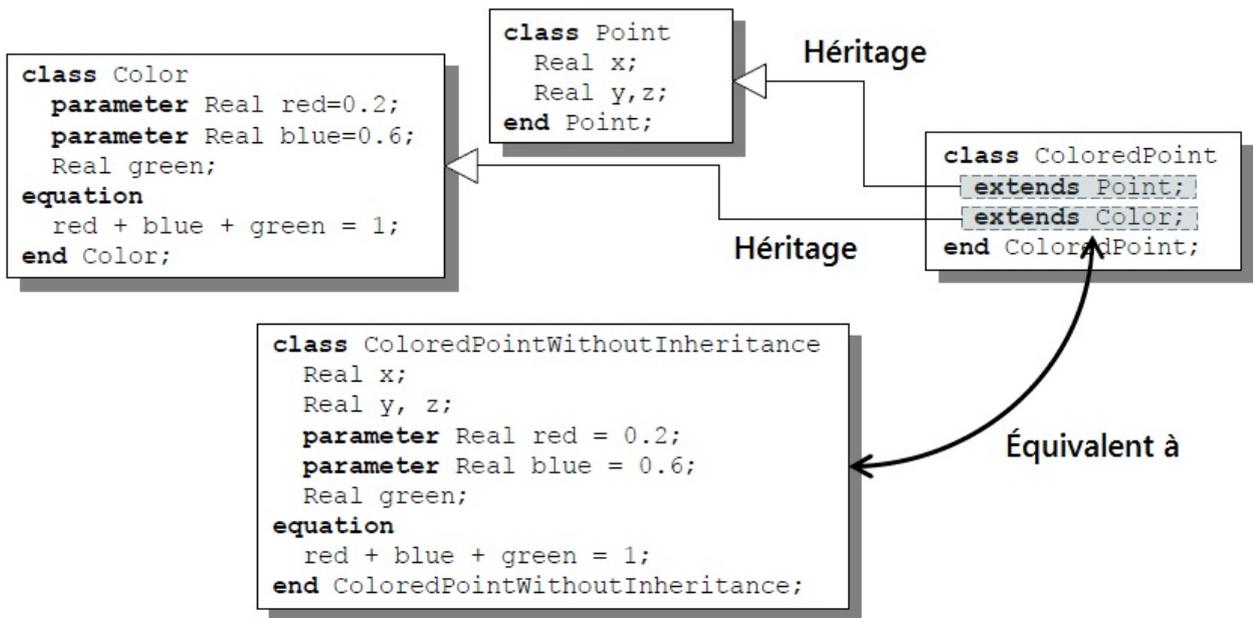


FIGURE A.1 – Principe d’héritage entre des classes de Modelica

ii. Variables d’état

Les variables sont habituellement des propriétés dont la valeur n’est pas connue d’emblée pour l’ensemble de la simulation. On peut catégoriser les différents types de variables de la manière suivante (RUE10) :

Variables à résoudre : Ce sont habituellement les variables différenciées. Ce sont les seules variables pouvant être initialisées. Les variables non-différenciées ne peuvent pas, quant à elles, être initialisées.

Variables de sortie : Les variables dont on désire communiquer la valeur à l’extérieur de l’objet et qui sont associées à une sortie de l’objet.

Variables d’entrée : Les variables d’entrées doivent être associées à une entrée de l’objet. Puisque la valeur initiale proviendra de l’extérieur, ces variables ne peuvent donc pas être initialisées.

Variables calculées : Données calculées à partir des autres variables et paramètres de l’objet.

iii. Paramètres

Les paramètres sont des éléments fixes pour la durée d’une simulation mais modifiables d’une simulation à l’autre. Ils sont habituellement utilisés pour emmagasiner

```

"classe 1 certia_rd"
class certia_rd
  Real x_rd;
  Real y_rd;
  Real z_rd;
equation
  z_rd = x_rd+y_rd ;
end certia_rd;
"classe 2 enp"
class enp
  Real xenp;
  Real yenp;
  Real zenp;
equation
  zenp = xenp-yenp ;
end enp;
"La classe Certia_ENP hérite des deux classes leurs propriétés"
class Certia_ENP
  Real z;
  extends certia_rd ;
  extends enp ;+
end Certia_ENP

```

FIGURE A.2 – Exemple d’héritage entre des classes de Modelica

les constantes de calibrage des modèles et autres entrées fixes du système. Au niveau de l’initialisation des paramètres, deux possibilités sont disponibles :

- **Initialisation directe** : La valeur du paramètre est fournie directement à l’initialisation.
- **Initialisation indirecte** : La valeur du paramètre provient d’un autre ou d’une combinaison d’autres paramètres .

iv. Équations

La section `equation` du code est l’endroit où apparaissent les lois décrivant le fonctionnement du système à représenter. Ces équations sont habituellement issues des lois de conservation de la matière et de l’énergie. C’est aussi l’endroit où l’on spécifie les règles de calcul des variables autres qu’à résoudre. Les énoncés d’interconnexion des composants du modèle font aussi partie de cette section. Afin d’avoir un système déterminé, il est important de spécifier le même nombre de variables à résoudre que d’équations. Autrement, la traduction du modèle à la compilation entraînera un message d’erreur de système singulier sur-spécifié ou sous-spécifié. Cette simple règle peut paraître triviale en soit mais peut devenir un véritable casse-tête lorsque le nombre de variables augmente. (RUE10)

Les équations sous Modelica peuvent être classées en quatre catégories : (HAD08)

- **Les équations normales**

Les équations normales sont utilisées dans ce bloc qui débute par le mot clé **équation**,

plusieurs sortes d'équations peuvent être présentes dans ce bloc des équations à égalité simple, des équations répétitives, des équations de connection, et des équations conditionnelles.

- **Les équations de déclaration.**
- **Les équations de modification.**
- **Les équations d'initialisation.**

Classes particulières

Les spécifications du langage Modelica définissent notamment les sept classes particulières suivantes ([JAR10](#)) :

1. Type : Définit un type, par extension d'un type prédéfini, il permet de définir les unités des variables.
2. Connector : Permet de décrire la manière dont les différentes classes de l'étude se connectent entre elles.
3. Model : Comme son nom l'indique, permet de décrire des modèles.
4. Block : Est utilisé pour la description de modèles sous le formalisme d'un bloc avec déclaration explicite des entrées et des sorties. La description reste acausale mais permet de restreindre le type de connexions possibles avec d'autres modèles.
5. Function : Classe block ne contenant pas d'équation et contenant au plus un algorithme.
6. Record : Classe sans équation, non exploitable comme connecteur.
7. Package : Autorise l'encapsulation de plusieurs classes dans une même et seule class et qui est ainsi utile pour la création de bibliothèques de modèles

Les composants, les connecteurs et les connexions

Les connecteurs

Un connecteur est une instance de la classe connector. Les connecteurs spécifient l'interface d'interaction entre les composants ([HAD08](#)). Les connecteurs peuvent être formulés pour plusieurs applications de différents domaines. Par exemple, le connecteur **FluidPort** est utilisé pour décrire les interfaces pour une interaction à une dimension entre deux composants fluidiques.

```

connector FluidPort
|
  replaceable package Medium = Modelica.Media.Interfaces.PartialMedium
    "Medium model" a;

  flow Medium.MassFlowRate m_flow
    "Mass flow rate from the connection point into the component";
  Medium.AbsolutePressure p "Thermodynamic pressure in the connection point";
  stream Medium.SpecificEnthalpy h_outflow
    "Specific thermodynamic enthalpy close to the connection point if m_flow < 0";
  stream Medium.MassFraction Xi_outflow[Medium.nXi]
    "Independent mixture mass fractions m_i/m close to the connection point if m_flow < 0";
  stream Medium.ExtraProperty C_outflow[Medium.nC]
    "Properties c_i/m close to the connection point if m_flow < 0";
  a;
end FluidPort;

```

FIGURE A.3 – Classe FluidPort

Le connecteur **Flange** est utilisé pour décrire les interfaces pour une interaction à une dimension entre deux composants mécaniques.

```

connector Flange_a
  "1-dim. rotational flange of a shaft (filled square icon)"
  SI.Angle phi "Absolute rotation angle of flange";
  flow SI.Torque tau "Cut torque in the flange";
  a;
end Flange_a;

```

FIGURE A.4 – Classe Flange

Deux sortes de variables sont utilisées dans les connecteurs :

i. **Variables potentiels** :

Lorsque deux connecteurs sont reliés, la connexion impose que les variables potentiels sont égales.

ii. **Variables flux** :

Lorsque deux connecteurs sont reliés, la connexion impose que les variables flux ont leur somme égale à 0.

Les composants

Un modèle Modelica est typiquement constitué de composants issus de domaines divers interagissant entre eux à travers des connexions qui les lient. Un composant est une classe qui doit être défini indépendamment de l'environnement où elle est utilisée. Ceci est essentiel pour la réutilisabilité. Dans la définition des équations d'un composant, seules les variables locales et les variables connecteurs doivent être utilisées. Le seul moyen pour la communication entre plusieurs composants est le passage par les connecteurs.

Connexion

La primitive `connect` est utilisée pour effectuer la connexion entre deux connecteurs ou plus et doit être spécifiée dans la section `equation` du code Modelica. Les connexions peuvent être soit causales ou acausales dépendamment du type de connecteur impliqué dans la connexion. Lorsqu'on utilise un environnement de développement avec interface graphique, les énoncés de connexion sont générés automatiquement au fur et à mesure que l'on procède à l'interconnexion de composants dans un modèle.

La connexion entre deux connecteurs se fait suivant la syntaxe :

```
connect(connector1,connector2)
```

A.4 Les étapes de traduction du compilateur Modelica

Le code source Modelica en entrée est transformé en un modèle plat. Dans cette étape, le compilateur vérifie les types des variables, génère les équations de connexion, substitue les paramètres fixés par leurs valeurs dans les équations, effectue les opérations liées à la notion orientée objet tel que l'héritage et fixe l'inclusion des packages. Le modèle plat contient un ensemble d'équations de déclarations et de fonctions. Les deux étapes suivantes sont l'analyse des équations puis leur optimisation. Elles sont nécessaires pour compiler les modèles contenant des équations (HAD08). En dernière étape, un Code C est généré, il permettra aux applications clientes de récupérer de l'information structurelle sur les modèles à analyser ou du code C compilable dont l'exécution dans un environnement de simulation permet d'obtenir l'état initial et les courbes d'évolution temporelles d'un système dynamique hybride. est introduit au compilateur C pour produire un fichier exécutable.

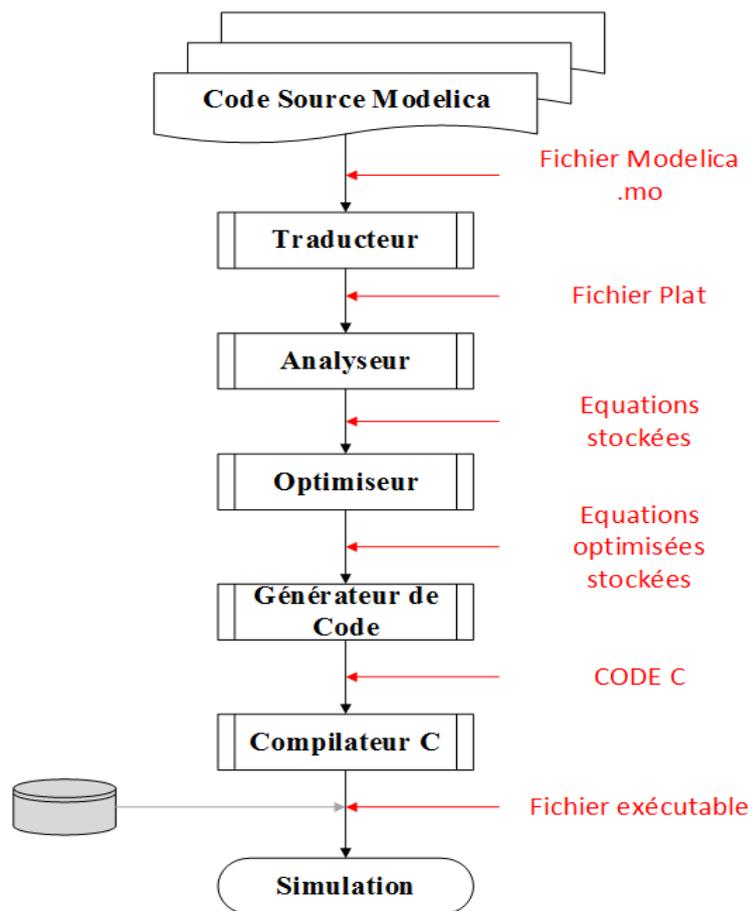


FIGURE A.5 – Les différentes étapes de traduction du code Modelica

Annexe B

B.1 Applications Lipschitziennes

Définition 4 (Applications Lipschitziennes (KHA01)) Soit f une application définie sur une partie Ω de E , à valeur dans F . Soit $\lambda > 0$. L'application f est dite λ -lipschitzienne ou encore lipschitzienne de rapport λ si :

$$\forall (x, y) \in \Omega^2, \| f(x) - f(y) \| \leq \lambda \| x - y \| . \quad (\text{B.1})$$

B.2 Preuve de la stabilité du modèle sans incertitudes

Preuve de l'unicité du point d'équilibre à l'origine

L'étude du point d'équilibre du système :

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_i \\ \vdots \\ \dot{e}_n \end{pmatrix} = \begin{pmatrix} -k_1 & g_1 & 0 & & \dots & & 0 \\ -g_1 & -k_2 & g_2 & & & & \\ 0 & \ddots & \ddots & \ddots & & & \\ & & & -g_{i-1} - k_i & g_i & \ddots & \\ & & & \ddots & \ddots & \ddots & \\ 0 & & & \dots & 0 & -g_{n-1} & -k_n \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix} \quad (\text{B.2})$$

revient à résoudre le système suivant (TUR10) :

$$e_1 = \frac{g_1}{k_1} e_2$$

$$\begin{aligned}
 e_2 &= \frac{g_{n-1}}{k_n} e_{n-1} \\
 e_i &= \frac{g_i}{k_i + \frac{g_{i-1}^2}{k_{i-1} + \frac{g_{i-2}^2}{k_{i-2} + \frac{g_{i-3}^2}{\ddots + \frac{g_1^2}{k_2 + \frac{g_1^2}{k_1}}}}} e_{i+1} \quad \forall i \in \{2, \dots, n-2\} \quad (\text{B.3})
 \end{aligned}$$

La dérivée de l'erreur est donnée par :

$$\dot{e}_i = \dot{x}_i - \dot{x}_{id} = f_i + g_i x_{i+1} - \dot{x}_{id} \quad \forall i \in \{1, \dots, n-1\} \quad (\text{B.4})$$

$$\dot{e}_n = \dot{x}_n - \dot{x}_{nd} = f_n + g_n u - \dot{x}_{nd} \quad (\text{B.5})$$

D'après l'équation (B.4), on a pour la $(n-1)^{\text{ème}}$ erreur :

$$\dot{e}_{n-1} = \dot{x}_{n-1} - \dot{x}_{(n-1)d} = f_{n-1} + g_{n-1} x_n - \dot{x}_{(n-1)d} \quad (\text{B.6})$$

En reportant (B.5) dans (B.3) avec $\dot{e}_{n-1} = 0$, on obtient :

$$e_{n-1} \left(\frac{g_{n-1}^2}{k_n} + k_{n-1} + \frac{g_{n-2}}{k_{n-2} + \frac{g_{n-3}^2}{k_{n-3} + \frac{g_{n-4}^2}{k_{n-4} + \frac{g_{n-5}^2}{\ddots + \frac{g_1^2}{k_2 + \frac{g_1^2}{k_1}}}}} \right) = 0 \quad (\text{B.7})$$

Sous les hypothèses déjà effectuées : $k_i > 0, \forall i \in \{1, \dots, n\}$ et $g_i \neq 0, \forall i \in \{1, \dots, n\}$

L'unique solution du système (B.3) est : $e_1 = e_2 = \dots e_i = \dots e_n = 0$

Preuve de la stabilité exponentielle globale du point d'équilibre

Considérons la fonction de Lyapunov suivante :

$$V = \frac{1}{2} \sum_{i=1}^n e_i^2 \quad (\text{B.8})$$

La dérivée de Lie de V s'exprime par :

$$\dot{V} = \sum_{i=1}^n e_i \dot{e}_i \quad (\text{B.9})$$

Ce qui conduit à :

$$\dot{V} = g_1 e_1 e_2 - k_1 e_{21} + \sum_{i=2}^n e_{n-1} (g_i e_i e_{i+1} - g_{i-1} e_{i-1} e_i - k_i e_i^2) - g_{n-1} e_{n-1} e_n - k_n e_n^2 \quad (\text{B.10})$$

Après développement on a :

$$\dot{V} = - \sum_{i=1}^n k_i e_i \text{ avec } k_i > 0, \forall i \in \{1, \dots, n\} \quad (\text{B.11})$$

Ainsi le point d'équilibre $e_1 = e_2 = \dots = e_i = \dots = e_n = 0$ est globalement exponentiellement stable .(??)

B.3 Preuve de la stabilité du modèle avec incertitudes

Lemme 1 (Stabilité entrée/état (TUR10)) Soit le système

$$\dot{x} = f(x, u) (x \in \mathbb{R}^x, u \in \mathbb{R}^u, f(0, 0) = 0) \quad (\text{B.12})$$

tel que :

- f est globalement Lipschitzien en x et u .
- $\dot{x} = f(x, u = 0)$ est globalement exponentiellement stable à l'origine.

Alors le système B.12 est entrée/état stable.

Preuve de la stabilité entrée/état

Le système sans les incertitudes B.2 étant exponentiellement stable à l'origine et les composantes du vecteur des incertitudes du vecteur des incertitudes étant bornées, alors d'après le lemme 1, le système est entrée/état stable.

Annexe C

C.1 Les expressions des fonctions non-linéaires

Les expressions des fonctions non-linéaires sont données par :

	<i>Les non – linéarités</i>	<i>Expression mathématique</i>
<i>Bras de Levier</i>	BL_{tube}	$a.l_{tube}(t) + b$
<i>Raideur du tube</i>	ψ_{MR}	$K_0 + K_1 l_{tube}(t)$
<i>Frottement de la vis</i>	ψ_{tube}	$F_{vis} w_{vis}(t)$

TABLE C.1 – Expressions mathématiques des non-linéarités

Les valeurs numériques des paramètres rencontrés dans les non linéarités sont :

a	0.75
b	0.35
K_0	2e7
K_1	-3e6

TABLE C.2 – Valeurs numériques des paramètres des non linéarités

C.2 Les expressions des régulateurs par H_∞ dans les 3 phases

Régulateur phase [Overstow - Flush]

$$G_1(s) = \frac{7.99e5s^6 + 8.02e10s^5 + 3.77e13s^4 + 3.19e16s^3 + 3.39e10s + 2.05e15}{s^7 + 5.91e06s^6 + 1.7e13s^5 + 2.92e15s^4 + 1.48e18s^3 + 2.30e20s^2 + 2.30e19s + 1.721e13}$$

Régulateur phase [Flush - Deployed]

$$G_2(s) = \frac{9.96e04s^6 + 1.00e10s^5 + 4.71e12s^4 + 3.94e15s^3 + 4.05e17s^2 + 2.47e20s + 1.16e21}{s^7 + 2.09e06s^6 + 2.18e12s^5 + 3.65e14s^4 + 1.77e17s^3 + 2.74e19s^2 + 5.758e18}$$

Régulateur phase [Deployed - Full reverse]

$$G_3(s) = \frac{7.99e05s^6 + 8.02e10s^5 + 3.77e13s^4 + 3.32e16s^3 + 4.00e18s^2 + 2.47e21}{s^7 + 5.91e06s^6 + 1.75e13s^5 + 2.92e15s^4 + 1.77e18s^3 + 2.76e20s^2 + 2.76e19s + 1.47e13}$$

ملخص:

هذا العمل يدخل في إطار مشروع واسع للبحث والتطوير تحت اسم "PRESAGE" منصة حقيقية ومحاكاة مشغلات ، عامة وقابلة للتطوير" ، يسعى هذا الأخير إلى إنشاء لوحات للتجارب الافتراضية لتحسين أداء النظم وخفض تكلفة التطوير والصيانة. ولتصميم هذه اللوحات استعنا ببرامج مكرسة لهذا النوع من النظم أولها Dymola/Modelica لتصميم الفيزيائي ، و MATLAB/Simulink لمحاكاة نظم التحكم.

النظام الذي تمت دراسته هو عاكس التوجه في طائرات ، واعتمدنا في كل من H_{∞} و Backstepping بتتبع مسار مرجعي. في آخر هذا التقرير ، قمنا بتطبيق تقنية لتشخيص الأخطاء في النظام المدروسة ، هذه التقنية تتركز على قواعد إحصائية تتمركز حول تحليل المركبات الرئيسية ACP.

الكلمات المفتاحية: عاكس التوجه ، Dymola ، التحكم بالتراجع ، تقنية H_{∞} ، التحليل بالمركبات الرئيسية ، المشغلات الكهرو-ميكانيكية.

Résumé :

Ce travail rentre dans le cadre d'un vaste projet de recherche appelé PRESAGE « Plateforme Réelle Et Simulée d'Actionnement ,Générique et Évolutive » , il vise à franchir une nouvelle étape dans la réalisation de bancs d'essais pour l'aéronautique en vitalisant le processus pour optimiser les performances recherchée. Le développement de la plateforme virtuelle s'est effectué sous Dymola/Modelica pour la modélisation et Matlab/Simulink pour la commande. Le système en question est l'inverseur de poussée, ce dernier revêt d'une importance capitale au sein de l'aéronef .Les commandes développés sont au nombre de deux: le H_{∞} Loopshaping et le Backstepping par poursuite de trajectoire référence. En fin de mémoire, nous avons développé une technique pour le diagnostic des défauts basée sur méthode statistique, l'Analyse en Composantes Principales (ACP).

Mots clés : inverseur de poussée, Dymola, Modelica, Backstepping, H_{∞} , Analyse en Composantes Principales, actionneur électromécanique.

Abstract :

This work is a part of a colossal scientific research project called "PRESAGE" the acronym of " Plateforme Réelle Et Simulée d'Actionnement ,Générique et Évolutive" (Real Stimulated Generic and Evaluative Platform of Actuation). Its aim is to pass a new stage in the realization of test beds in the aeronautic world by revitalizing the whole process in order to optimize the sought results. The virtual platform has been developed in Dymola/Modelica, while the modeling of the command has been done in Matlab/Simulink. The system is a thrust reverser, which is of a big importance in the aircraft domain. The are two developed commands: a H_{∞} Loopshaping and a Backstepping command with tracking the reference trajectory. Finally, we developed a defects diagnosis technique based on a statistic method Principal component analysis (PCA).

Key words : thrust reversal, Dymola, Modelica, Backstepping, H_{∞} , Principal component analysis, electromechanical actuator .