

1700
et



المدرسة الوطنية المتعددة التقنيات
المكتبة
BIBLIOTHEQUE
Ecole Nationale Polytechnique

Département: Génie-Électrique

Laboratoire de Commande des Processus

Thèse de Doctorat d'Etat

Option : Automatique

Présentée par

LINDA BARAZANE

Magister en Génie-Électrique de l'ENP

Option : Électrotechnique

Thème

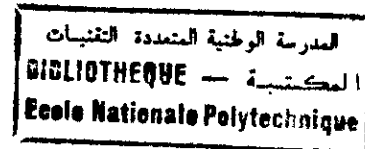
Application des Systèmes Emergents à la commande
d'un Moteur Asynchrone.

Soutenu publiquement le 23 novembre 2003, devant le jury composé de :

Président :	F. BOUDJEMA	Professeur à l'ENP
Rapporteurs :	M.S. BOUCHERIT	Maître de conférence à l'ENP
	E.M. BERKOUK	Maître de conférence à l'ENP
Examineurs :	R. OUIGUINI	Maître de recherche au CDTA
	M.O. MAHMOUDI	Maître de conférence à l'ENP
	M. TADJINE	Maître de conférence à l'ENP
Invités :	M.E. SIFI	Chef de Mission Audit Materiel et Traction SNTF
	C. LARBES	PhD , Chargée de cours à l'ENP



المدرسة الوطنية المتعددة التخصصات
Ecole Nationale Polytechnique
Département: Génie-Électrique



Laboratoire de Commande des Processus

Thèse de Doctorat d'Etat

Option : Automatique

Présentée par

LINDA BARAZANE

Magister en Génie-Électrique de l'ENP

Option : Électrotechnique

Thème

**Application des Systèmes Emergents à la commande
d'un Moteur Asynchrone.**

Soutenu publiquement le 23 novembre 2003, devant le jury composé de :

Président :	F. BOUDJEMA	Professeur à l'ENP
Rapporteurs :	M.S. BOUCHERIT	Maître de conférence à l'ENP
	E.M. BERKOUK	Maître de conférence à l'ENP
Examineurs :	R. OUIGUINI	Maître de recherche au CDTA
	M.O. MAHMOUDI	Maître de conférence à l'ENP
	M. TADJINE	Maître de conférence à l'ENP
Invités :	M.E. SIFI	Chef de Mission Audit Materiel et Traction SNTF
	C. LARBES	PhD, Chargée de cours à l'ENP

- لقد وجهنا عملنا حول تقنيات التحكم الحديثة وهي تلك التي تركز على شبكات الخلايا العصبية الاصطناعية والنماذج الغامضة:
- قصد تجاوز بعض مشاكل التحكم في المحرك اللامتزامن، مثل ربط متغيرات الوضعية، لديناميكيات غير الخطية، الإرتيانات الوسيطة وغير الوسيطة المتعلقة بالنظام.
 - بهدف تقليص ظاهرة "الشاترينغ" (chattering) في عملية لفتح ذات الهياكل المتغيرة.

وبالتالي تجسدت نتيجة جهدنا في هذا العمل من خلال مساهمتين هما:

1. مخطط تحكم موثوق قائم على شبكات الخلايا العصبية الاصطناعية، موجه لتقريب العبارات غير الخطية لقانون التحكم غير المستمر المقترح، والمنجز مسبقا بمعدلات الهياكل المتغيرة لعملية التحكم المختلط الواقع عليه الاختيار.
2. من جهة أخرى، ومن أجل تقادي بعض حدود التقنية السالفة، وجهنا عملنا نحو التشكيل الغامض للنماذج بغرض إعداد تحكم جديد ذات هياكل متغيرة غامضة تركز على نموذج غامض للمحرك اللامتزامن، والذي تم تطبيقه للمرة الأولى في تحكم المحركات الكهربائية. تسمح هذه التقنية بمحاكاة النظام عن طريق تمثيل غامض أخذا بعين الاعتبار الوصف التحليلي العام وأرتيانات النظام.

المفاتيح: محرك لامتزامن، تحكم شعاعي، تحكم تخطيطي، تحكم مختلط، تحكم ذات هياكل متغيرة، شاترينغ، تحكم موثوق مختلط بشبكات الخلايا العصبية الاصطناعية، تشكيل غامض للنماذج، تحكم غامض ذات هياكل متغيرة قائم على نموذج غامض.

Résumé :

Nous avons dirigé notre travail sur les techniques de commande émergentes à savoir celles basées sur les réseaux de neurones artificiels (RNA) et les modèles flous :

- Dans un effort visant à résoudre certains problèmes de commande du moteur asynchrone, tels que le couplage des variables d'état, les dynamiques non-linéaires, les incertitudes paramétriques et non paramétriques inhérents au système.
- et dans le but de réduire le phénomène de chattering dans le processus de commande à structures variables.

Ainsi, le fruit de notre travail de doctorat s'est concrétisé par deux contributions:

1. Un schéma de commande robuste basé sur les RNAs destinés à approximer les termes non-linéaires de la loi de commande discontinue proposée et qui est réalisée au préalable par les régulateurs à structures variables du processus de commande hybride adoptée.
2. D'autre part et pour contourner certaines limitations causées par la technique précédente, nous avons orienter notre travail vers la modélisation floue dans le but d'élaborer une nouvelle commande à structure variable floue basée sur un modèle flou de la machine asynchrone et qui est appliquée pour la première fois dans la commande des machines électriques. Cette technique sert à imiter le système par une représentation floue tout en tenant compte de la description analytique grossière et des incertitudes du système.

Mots-clés : Moteur asynchrone, control vectoriel, commande linéarisante, commande hybride, commande à structures variables, chattering, commande robuste hybrid par réseaux de neurones artificiels, modélisation floue, commande à structures variables floue basée sur un model flou.

Abstract :

This research work is based on the application of artificial neural networks (ANRs) and fuzzy models:

- In order to solve some control problems of the asynchrone motor such as, state variables coupling, dynamic nonlinearities, parametric and nonparametric uncertainties inherent to the system.
- And in the objective to reduce the chattering phenomenon in variable structure control system.

So, the result of this reaserch consists on:

1. a new robust control scheme based on ANRs aimed to approximate the nonlinear terms of the proposed discontinuous control law which were previously implemented by variable structure regulators in the adopted hybrid control.
2. in the other hand, and in order to overcome the limitations of the previous technique, fuzzy modelling was proposed to elaborate a new approach to fuzzy variable structure control based in this case on the fuzzy model of the asynchrone motor which is applied for the first time in electrical motors control. This technique help to imitate the system model by a linguistic fuzzy rule base representation, with taking into account the mathematical description and the boundary uncertainties of the system.

Key words : Induction motor, vector control, feedback linearisation control, hybrid control, variable structure control, chattering, robust hybrid control based on artificial neural networks, fuzzy modelling, fuzzy variable structure control based on fuzzy model.

Dédicaces

C' est,

À *Bon Dieu le tout puissant qui m'a aidé à concrétiser le rêve des gens que j'aime,*

À *mes très chers parents,*

À *la mémoire de mes grand-parents,*

À *tous ceux qui me sont et me seront chers,*

À *tous ceux qui donnent ou donneront un sens à ma vie,*

À *tous ceux dont la seule présence embellit mes journées,*

Et sont pour moi comme une source intarissable de joie,

Et de bonheur,

À *chaque âme inoubliable que j'ai pu croiser un jour au cours
de ma vie,*

*Et dont l'immensité du cœur se trouve être garnie de toutes sortes
de passions, noblesse, générosité et grandeur d'âme,*

À *toute personne qui m'a profondément marquée soit par une amitié*

Dévouée ou tout autre sentiment noble,

*Et qui a laissé au fond de mon âme un sentiment de bonheur intense,
de richesse et d'épanouissement.*

Et surtout, à tous ceux que j'estime ou estimerai plus que tout au monde

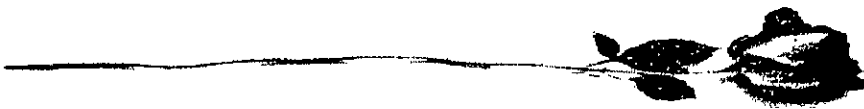
Et qui ont su ou sauront m'aimer pour ce que je suis.

Que je dédie cet humble travail,

Avec toute la sincérité dont un être humain est capable de ressentir.



Linda.



المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Avant Propos

Les travaux présentés dans cette thèse ont été effectués en Algérie, sous la direction de Messieurs M.S.Boucherit et E.M.Berkouk, Maîtres de Conférence au sein du Laboratoire de Commande des Processus (LCP) de l'Ecole Nationale Polytechnique (ENP), qu'ils trouvent ici le témoignage de ma profonde reconnaissance, pour leur encadrement, leurs directives, en particulier Mr Boucherit pour la confiance qu'il m'a accordé, ses conseils continus et surtout pour sa générosité.

Je tiens également à exprimer ma gratitude la plus absolue et mes remerciements les plus sincères à Mr R. Ouiguni, Maître de Recherche au Centre de Développement des Technologies Avancées (CDTA), qui a bien voulu superviser mon travail depuis le début, en me faisant profiter de ses connaissances et ses nombreuses expériences. Je n'oublie pas de le remercier pour sa disponibilité, ses encouragements, ainsi que pour toutes les discussions fructueuses que nous avons pu avoir. De même, pour l'honneur qu'il me fait en acceptant de participer au jury de soutenance.

Je ne saurais jamais comment exprimer ma profonde reconnaissance et gratitude à Mr A. Mezian du (CDTA) pour son aide, sa patience, ses perpétuels conseils et encouragements qui m'ont permis ne pas baisser les bras devant les difficultés et d'aller toujours de l'avant, faisant ainsi de mon rêve une réalité.

Par ailleurs, j'aimerais préciser que les travaux présentés dans cette thèse ont été entamés au départ, dans le laboratoire de Machines Électriques sous la direction de Messieurs R. Ibtouen, et O. Touhami, Professeurs à l'ENP que je tiens à remercier.

Au cours de son élaboration et dans le but de valider les résultats expérimentaux obtenus sur un banc d'essai avec Dspace, un stage a été effectué à la Chaire Hydro-Québec de l'Université du Québec à Trois Rivières (Canada) sur l'invitation et la direction de Messieurs P. Sicard, Professeur et A. Ba-Razzouk Chercheur Post-Doctoral qui ont beaucoup contribué à enrichir et à mener à bien ce travail. Qu'ils soient assurés de toute ma gratitude, en particulier, le Professeur D.S. Zinger de l'Université d'Akron (U.S.A) qui m'a soutenu, encourager et diriger depuis que j'ai entamé ma thèse de Magister et jusqu'à ce jour.

Il m'est particulièrement agréable de remercier Messieurs C. Larbes et Y. Sellami pour leur aide précieuse et la participation effective dans la réalisation de certains de ces travaux.

Je remercie très sincèrement Mr F. Boudjema, Professeur à l'ENP pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont également à Messieurs M.O. Mahmoudi et M. Tadjine, Maîtres de Conférence à l'ENP, pour leurs précieux conseils, directives lors de l'élaboration de ce travail et aussi pour avoir accepté de participer au jury de soutenance.

Je n'oublie de citer les conseils, le soutien moral et la gentillesse qui m'ont été prodigué de la part de Messieurs L. Nezli, B. Hemici et S. Ait-Cheikh, Chargés de cours à l'ENP, qu'ils trouvent ici le témoignage de toute ma reconnaissance.

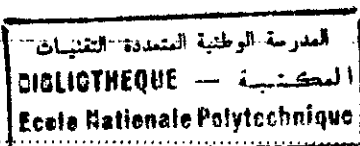
Que mes amis (es) soient aussi sincèrement remerciés pour leur encouragements afin de mener à terme les travaux de cette thèse.

A toutes les personnes, dont le nombre est très élevé pour pouvoir les citer ici, qui ont contribué de près ou de loin à l'aboutissement de ce travail, qu'ils trouvent ici le témoignage de ma profonde reconnaissance et je les remercie vivement pour leur aide efficace et soutien moral.

Sommaire

INTRODUCTION	1
CHAPITRE I. COMMANDES VECTORIELLE ET PAR RETOUR D'ÉTAT LINEARISANT DE LA MACHINE ASYNCHRONE	
I.1 Introduction	9
I.2 Modélisation de la Machine Asynchrone	10
I.3 Commande vectorielle.....	11
I.4 Commande par retour d'état linéarisant.....	17
I.4.1 Application à la machine asynchrone.....	17
Étape 1 : Choix des grandeurs de sortie.....	17
Étape 2 : Calcul du degré relatif et changement de coordonnées.....	18
Étape 3: Retour d'état non-linéaire.....	19
Etappe 4: Stabilité asymptotique et poursuite de référence.....	20
I.5 Commande hybride linéarisante et à orientation du flux rotorique.....	21
I.6 Conclusion.....	28
CHAPITRE II. COMMANDE HYBRIDE A STRUCTURES VARIABLES DE LA MACHINE ASYNCHRONE	
II.1 Introduction	29
II.2 Les systèmes à structures variables.....	30
II.3 Conception des régulateurs à structures variables	30
II.3.1 Choix de la surface de glissement.....	31
II.3.2 Condition d'existence du mode de glissement	31
II.3.3 Choix des grandeurs de commande	32
II.3.4 La commande équivalente.....	32

II.3.5 Cas des systèmes mis sous forme canonique.....	34
II.3.6 Réduction du phénomène de chattering	37
II.4 Application à la commande hybride de la machine asynchrone.....	39
II.4.1 Application de la commande u_n de type relais avec un degré relatif $r=1$	40
II.4.2 Application de la commande adoucie proposée	41
II.5 Les inconvénients et les contraintes liées à la commande classique.....	45
II.6 Conclusion.....	46
CHAPITRE III. COMMANDE HYBRIDE PAR RESEAUX DE NEURONES ARTIFICIELS DE LA MACHINE ASYNCHRONE	
III.1 Historique.....	47
III.2 Du neurone biologique au neurone artificiel.....	48
III.4 Réseaux de neurones artificiels	49
III.4.1 L'élément de traitement.....	49
III.4.2 L'architecture d'un réseau	49
III.4.2.1 Réseaux statiques ou multicouches.....	49
III.4.2.2 Réseaux dynamiques ou récurrents.....	50
III.4.2.3 Réseaux topologiques.....	51
III.4.3 Apprentissage et adaptation	52
III.4.3.1 Apprentissage supervisé	53
III.4.3.2 Apprentissage non supervisé	53
III.4.3.3 Différents algorithmes d'apprentissage des Réseaux de Neurones Artificiels	53
III.5 Certaines considérations pratiques pour le choix du réseaux de neurone adéquat.....	57
1. Choix et dimensionnement d'un réseau	57
2. Paramètre d'apprentissage ou pas de correction η	58
3. Poids initiaux.....	58
4. Minimums locaux	58
5. Saturation du réseau.....	58



III.6 Les réseaux de neurones et la commande des processus	58
III.6.1 Utilité de la commande supervisée :	59
III.7 Commande supervisée par réseaux de neurones de la commande hybride.....	60
III.7.1 Architecture des réseaux de neurones artificiels adoptée	60
III.7.2 Processus d'entraînement du réseau de neurone de la vitesse (RNA_1)	61
III.7.3 Processus d'entraînement du réseau de neurone du flux rotorique (RNA_2)	62
III.7.4 Validation de la régulation de vitesse avec les régulateurs robustes par réseaux de neurones (RNAs)	63
III.10 Conclusion	67
CHAPITRE IV : MODELISATION ET COMMANDE ROBUSTE FLOUES DE LA MACHINE	
ASYNCHRONE	
IV.1. Introduction.....	69
IV.2. Évaluation des systèmes réels et flous.....	69
IV.3. Les systèmes d'inférence flous	70
IV.3.1. Configuration de base d'un système d'inférence flou	71
IV.4. Les modèles flous à base de règles.....	75
IV.4.1. Partitionnement homogène	76
IV.4.2. Partitionnement par concaténation d'ensembles flous	77
IV.4.3. Partitionnement optimal.....	78
IV.4.4. Différents types de modèles flous	78
IV.4.4.1. Le modèle flou linguistique.....	79
IV.5. Élaboration d'un modèle flou- Modélisation floue.....	82
IV.6. Modélisation floue linguistique de Ben-Ghalia	83
IV.6.1. Condition d'existence de modèle flou selon la méthode adoptée.....	84
IV.6.2. Algorithme de la modélisation floue de Ben-Ghalia	85
IV.6.3. Évaluation de la base de règles du vecteur de commande	90
IV.6.4. Obtention du modèle flou	90

IV.7. Commande à Structure Variable Floue basée sur le modèle flou de Ben-Ghalia.....	91
IV.7.1. Algorithme de la commande à structure variable floue adopté.....	92
IV.7.2. Cas des systèmes mis sous forme canonique.....	98
IV.7.3. Modélisation floue du sous-système.....	99
IV.7.4. Élaboration de la Commande à structure variable floue.....	100
IV.7.5. Application à la commande du moteur asynchrone.....	103
IV.7.5.1. Représentation du système sous forme canonique.....	104
IV.7.5.2. Présentation du modèle flou du moteur asynchrone.....	108
IV.7.5.3. Résultats de simulation.....	109
IV.7.5.4. Commande par mode de glissement.....	111
IV.7.5.5. Résultats de simulations.....	112
IV.7.6. Commande robuste floue de Ben-Ghalia de la machine asynchrone.....	117
IV.7.6.1. Elaboration de la commande floue robuste.....	117
IV.7.6.2. Résultats de simulation.....	119
IV.8. Conclusion.....	125
CONCLUSION GENERALE	127
ANNEXE	
Annexe A. Paramètres du moteur asynchrone.....	A.1
Annexe B. Caractéristiques des ensembles flous.....	A.2
Annexe C. Démonstration du lemme (4.1).....	A.7
Les fonctions de Carathéodory.....	A.8

BIBLIOGRAPHIE

Nomenclature

- c, b, c : indices correspondant aux trois phases de la machine
- d, q : indices correspondant au référentiel lié au champ tournant
- α, β : indices correspondant au référentiel fixe (lié au stator)
- i : courant instantané
- v : tension instantané
- Φ : flux instantané
- R_s, R_r : résistances du stator et du rotor
- L_s, L_r : inductances (cycliques) du stator et du rotor
- L_m : inductance mutuelle cyclique entre le stator et le rotor
- L_{as}, L_{ar} : inductances d'une phase statorique et d'une phase rotorique
- T_s : constante de temps statorique (L_s / R_s)
- T_r : constante de temps rotorique (L_r / R_r)
- σ : coefficient de fuite ($\sigma = 1 - L_m^2 / (L_r L_s)$)
- p : nombre de paires de pôles
- Ω : vitesse mécanique
- ω_{sl} : vitesse de glissement
- θ_r : position du rotor ($\Omega = d\theta_r / dt$)
- θ : position du référentiel par rapport au stator
- ω_s : pulsation statorique
- ω_r : pulsation rotorique
- J : moment d'inertie
- f : coefficient de frottements visqueux
- C_{em} : couple électromagnétique
- C_r : couple résistant

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction Générale

Introduction générale

Dans diverses applications industrielles modernes, la machine asynchrone à cage bénéficie d'une attention particulière due essentiellement, à sa fiabilité, sa robustesse, son coût relativement faible et sa simplicité de construction. Toutefois, et contrairement aux simplifications offertes par le système balai - collecteur dans la machine à courant continu, la dynamique de ce type de machine se trouve être non-linéaire, multivariable et fortement couplée. En plus, non seulement certaines de ses variables d'état sont inaccessibles aux mesures directes (flux rotorique) mais aussi, les paramètres de la machine (en particulier, la résistance rotorique) sont affectés par l'effet thermique et les conditions de fonctionnement. Ce qui rend sa commande relativement difficile.

De nos jours et vu l'intérêt porté à ces actionneurs, les techniques de commande synthétisées sont de plus en plus complexes du fait qu'elles doivent répondre à des exigences de plus en plus sévères.

En effet, toute commande développée doit d'une part, avoir pour objectif la simplification du modèle mathématique de la machine asynchrone tout en assurant et le découplage entre ses deux principales dynamiques (la vitesse et le flux), et une certaine robustesse vis-à-vis de la variation des paramètres, de l'incertitude liée aux mesures et/ou estimations (observations) des variables d'état, d'autre part, permettre une mise en œuvre des lois de commande assez simple et assez souple pour une implémentation aisée et efficace. Dans ce contexte, avec la proposition de nouvelles théories de commande destinées à formuler des lois de pilotage automatique des machines asynchrones lesquelles, grâce à l'accroissement de la puissance de traitement des calculateurs numériques, l'apparition des DSP "Digital Signal Processing" et la disponibilité de sources d'alimentation réglables (convertisseurs de fréquence), ont permis la mise en œuvre de plusieurs schémas de commande performants. Ces techniques peuvent être classées en deux catégories à savoir: les techniques classiques et celles dites émergentes.

En ce qui concerne les techniques de réglage classiques, celles-ci sont de nature linéaire, et utilisent généralement des régulateurs standards ou à contre réaction d'état.

Malheureusement, si leur utilisation a permis d'assurer le contrôle de systèmes classiques plus ou moins compliqués, il se trouve qu'elles ont montrés leurs limites pour le cas de la machine asynchrone. En effet, la particularité du comportement dynamique de ce type de machine, caractérisée par une forte non-linéarité et un couplage interne étroit entre les diverses variables d'état, pose un problème important dans le cas de sa commande par ces techniques.

Ce n'est qu'au début des années soixante-dix que cette ambition qui consiste à faciliter la commande de la machine asynchrone est devenue réalisable et ce, avec la proposition de la commande vectorielle introduite par **Blaschke**, basée sur un changement de coordonnées et un retour d'état non-linéaire qui permet de réduire la complexité du modèle dynamique de la machine et assure un découplage asymptotique en régime établi des deux principales grandeurs "Couple - Flux".

La machine acquiert ainsi, un comportement similaire à celui de la machine à courant continu dont les propriétés de découplage sont réalisées naturellement par le biais de l'ensemble balai - collecteur.

D'autre part, en 1980, en ayant recours à des convertisseurs de fréquence, **W. Leonhard** [LEO85] a introduit les régulateurs PI dans le processus de commande vectorielle. Le but de sa technique réside dans l'amélioration des performances dynamiques en assurant une certaine robustesse vis-à-vis des variations des paramètres. Parmi d'autres travaux basés sur le principe de ce type de commande, nous pouvons citer ceux de **A. Bellini** [BEL86] qui a proposé une version de commande optimale et ceux de **C.M. Liaw** [LIA88], **C.C. Chan** [CHA90] qui ont utilisé la commande par modèle de référence.

Néanmoins, il est à signaler que les techniques de commande basées sur l'orientation du champ possèdent quelques inconvénients qui ont limité leurs performances. En effet, le découplage ciblé ne peut être assuré qu'asymptotiquement ou dans les régimes statiques (permanents) [LEO85], [SLO91]. D'un autre côté, si le module du flux est réduit pour faire fonctionner la machine asynchrone en dehors des limites de saturation du circuit magnétique et des tensions d'alimentation, la dynamique de la vitesse se dégrade considérablement. D'autre part, les algorithmes de commande utilisés dans la commande vectorielle étaient au début basés sur des régulateurs classiques linéaires dépendants du modèle mathématique simplifié de la machine et qui est déterminé pour ses paramètres nominaux. Une variation de ces derniers entraînerait inévitablement des incertitudes sur les coefficients de ce régulateur et lui font perdre les qualités de réglage exigées. De plus, le découplage "*Couple-Flux*" est très sensible aux variations des paramètres internes de la machine, en particulier à la constante de temps rotorique très influencée par l'effet thermique.

Pour pallier à ces problèmes, le recours aux techniques de commandes non-linéaires [ISI89], [SLO91], [BOD94], [KHA96] s'avérerait opportun et justifié. Parmi les techniques de commande non-linéaires largement utilisées, nous citons le bouclage linéarisant ou "the feedback linearisation" [ISI89], [SLO91], [BOD94], [KHA96], [SIR89]. L'idée de base de cette technique consiste en la transformation du modèle réel de la machine en un modèle totalement ou partiellement linéarisé, pour lequel une technique de commande linéaire est ensuite appliquée. Ce qui constitue un avantage de taille du point de vue synthèse de commande.

Grâce aux intéressants travaux développés par **A. Isidori et al.** [ISI89], **S.S. Sastry et al.** [SAS89], **R. Marino et al.** [MAR91], **I. Kanellakopoulos et al.** [KAN91] et autres, les théories de base pour le retour d'état linéarisant ont été mises en évidence. Les différentes conditions nécessaires et suffisantes pour qu'un système soit linéarisable par retour d'état ont été déterminées, ainsi que les lois de commande qui peuvent être engendrées.

La première rencontre de la machine asynchrone avec ce type de techniques a eu lieu avec les travaux de **A. De Luca et al.** [DEL87], [DEL89]. Les méthodes proposées sont basées essentiellement sur un modèle simplifié de la machine asynchrone qui ne tient compte que de la dynamique de la partie électromagnétique représentée par quatre variables d'état (les courants statoriques et les deux composantes du flux rotorique). Dans une première approche, une linéarisation complète était réalisée en considérant la vitesse rotorique comme un paramètre fixe [DEL87]. Dans une autre technique, le modèle obtenu est reconsidéré en supposant la vitesse comme un paramètre variant très peu et fonction des variables électriques. Par la suite, une linéarisation dynamique entrée/sortie par retour d'état avec un découplage exact entre la commande de

flux et celle du couple par une compensation dynamique de second ordre est obtenue, tout en vérifiant la stabilité de la dynamique d'ordre zéro (le modèle possède comme sorties, le module du flux rotorique et le couple électromagnétique et comme entrées, l'amplitude et la fréquence des tensions d'alimentation) [DEL89].

Bien que ces approches aient permis la simplification de la dynamique de la machine asynchrone en séparant ses deux comportements magnétique et mécanique l'un de l'autre, et facilité la mise en œuvre des schémas de commande, elles se sont avérées inefficaces devant la variation des paramètres et l'incertitude des valeurs des variables d'état (mesurées, estimées ou observées), en particulier, dans le cas des régimes transitoires et des régimes dégradés. D'autre part, il est à signaler que le recours aux méthodes d'adaptation classiques, pour compenser l'effet de ces variations et de ces incertitudes, est lui aussi confronté à plusieurs limitations telles que : la nécessité d'une modélisation précise, la complexité des schémas de commande résultants, la complexité des lois de commande générées, l'implémentation en temps réel, ainsi que la difficulté d'évaluer correctement la déviation des paramètres en temps réel. C'est pourquoi, il a fallu orienter les recherches vers de nouveaux axes, plus précisément depuis que les progrès technologiques en matière d'électronique de puissance, de la micro-électronique (VLSI) et des processeurs numériques DSP (Digital Signal Processor) a permis l'implémentation d'algorithmes de commande plus robustes et performants, parmi ces techniques nous pouvons citer : la commande à structures variables et les modes glissants associés qui est connue pour sa puissance du point de vue rapidité en temps de réponse de rejet de perturbations et insensibilité vis-à-vis des variations paramétriques. Cette technique possède deux objectifs successifs: contraindre le point représentatif du système à joindre une hypersurface le plus rapidement possible, puis de le faire glisser jusqu'à ce qu'il atteigne un certain point d'équilibre. Lors de cette dernière phase, le système est en régime glissant et adopte la dynamique de l'hypersurface considérée qui est indépendante des paramètres du système, ainsi que des perturbations, et par conséquent, insensible aux variations de ces derniers.

Les premiers travaux sur la commande des systèmes à structure variable furent élaborés lors des années soixante en ex-Union Soviétique par l'équipe du chercheur **Emelyanov** [EME65].

Cette technique fut d'abord développée pour un système linéaire du second ordre, et depuis, le spectre de son utilisation fut élargi à une plus grande classe de systèmes comprenant les systèmes linéaires [UTK77], non-linéaires [GAO93], [CAR88], discrets [YOU96], multivariables (MIMO) [GAO93], hiérarchisés et stochastiques. Par ailleurs, les concepts de la structure variable furent étendus du problème de stabilisation à celui de la poursuite de trajectoire [HAS85], ou de modèles [RAC96], à la commande adaptative et optimale [COS90], [HSU89], [UTK96], ou encore à la synthèse d'observateurs [UTK96]. **Sabanovic** et **Izosimov** [SAB81], [SAB89], furent les premiers à utiliser la commande à structures variables pour la commande des moteurs électriques.

De même que pour les commandes précédentes, bien qu'ayant divers avantages, cette technique de commande possède également un inconvénient qui a limité son utilisation. En effet, dans la pratique, les imperfections telles que, les retards de commutation et l'hystérésis engendrent inévitablement des oscillations autour de la surface de glissement pouvant exciter les dynamiques non-modélisées, soumettant

ainsi les organes de commande à une intense activité, néfaste pour les actionneurs. Ce phénomène est connu sous le nom de chattering.

Afin de le réduire, les recherches se sont orientées vers l'utilisation de techniques consistant à faire des approximations continues de la fonction discontinue au voisinage de la surface de commutation qui se fait malheureusement au prix d'une perte de précision [HUN91], [BAR98].

Lorsque le concepteur du système de commande cherche une meilleure robustesse vis-à-vis d'éventuelles variations paramétriques, d'autres types de commande intégrant les propriétés de robustesse sont alors nécessaires. En effet, l'utilisation des techniques de l'intelligence artificielle pourrait constituer une autre alternative pour contourner le problème du chattering tout en assurant une commande robuste, entre autres les réseaux de neurones artificiels, la théorie des ensembles flous et les modèles flous.

Dans ce contexte, et dans un effort visant à résoudre les problèmes posés par la reconstitution des dynamiques non-linéaires incertaines, des schémas de commande robustes basés sur les réseaux de neurones artificiels de types multicouches ont été proposés pour approcher ces termes non-linéaires. En effet, les possibilités d'approximation des réseaux de neurones ont été clairement vérifiées par plusieurs travaux tels que ceux de **S. Narendra** [NAR90]. Il a été démontré qu'un réseau de neurones de trois couches (d'entrée, cachée et de sortie) avec un nombre approprié de neurones dans la couche cachée peut approximer n'importe quelle type de fonction non-linéaire.

Ainsi, les réseaux multicouches ont été introduits dans plusieurs applications lorsque nous disposons d'une base d'apprentissage convenable. Pour la commande des machines électriques, **L.A. Cabrera et al.** ont utilisés ce type de réseaux dans la commande directe du couple pour identifier la résistance statorique et pour sélectionner les états de commutations pour les bras de l'onduleur qui alimente la machine [CAB97]. **A. Ba-Razzouk et al.** ont proposés un circuit de découplage à base de réseaux de neurones artificiels pour la commande vectorielle directe à flux rotorique [BAR97]. **M.G. Simões** et **B.K. Bose** ont proposés un réseau de neurones multicouches susceptibles d'estimer les valeurs du module et de la position du flux rotorique, ainsi que celle du couple électromagnétique afin d'être utilisés dans la commande vectorielle directe [SIM95].

Suite aux travaux sus-cités, les réseaux multicouches basés sur l'apprentissage en temps réel ont suscité un grand intérêt, en particulier pour l'identification et la commande des systèmes non-linéaires. En effet, ce type de réseaux est muni de règles d'apprentissage permettant d'ajuster continuellement et en temps réel les paramètres du réseau de neurones, ce qui les rend aptes à s'adapter aux différents régimes dynamiques du système étudié. En plus, la variation non-linéaire de leurs paramètres ainsi que leurs comportements non-linéaires leur permettent de s'intégrer aisément dans les systèmes de commande non-linéaires adaptatifs, ce qui constitue une caractéristique très recherché. Pour l'apprentissage de tels réseaux, il existe deux approches. La première approche, utilise l'algorithme de rétro-propagation en temps réel, tandis que la seconde, est basée sur des règles d'adaptation obtenues en combinant les propriétés des réseaux de neurones considérés et celle de la commande adaptative classique. Ces réseaux auto-adaptatifs ont été adoptés dans plusieurs domaines, en particulier, pour la commande des systèmes non-linéaires. Avec la première approche d'apprentissage, **Q. Song et al.**[SON98] ont proposé un algorithme robuste de rétro-propagation pour l'identification et la commande des systèmes non-linéaires. **F.J. Lin et al.**[LIN94]

ont utilisé ce type de réseau pour la commande d'un servomoteur. En ce qui concerne la deuxième approche, les méthodes et les principes de la commande adaptative classique des systèmes non-linéaires avec dynamique incertaine sont utilisées. A cet effet, quelques lois d'adaptation classiques telles que celles développées par P.A. Ioannou *et al.* [IOA83], par S. Narendra *et al.* [NAR87] et autres pour les systèmes classiques ont été empruntées pour générer des lois d'apprentissage pour les réseaux de neurones multicouches. Avec ces règles d'adaptation, les réseaux obtenus ont été exploités pour améliorer la commande non-linéaire adaptative, en particulier celle basée sur le retour d'état linéarisant. Dans ce cadre, A. Yazildirek *et al.* [YAZ95], T. Zhan *et al.* [ZHA99], S. Jaganathan *et al.* [JAG96] ont proposé des régulateurs adaptatifs pour des systèmes qui sont linéarisables par retour d'état. Ainsi, les réseaux de neurones utilisés ont permis la reconstitution des dynamiques incertaines avec une grande précision.

Parallèlement, aux travaux utilisant les réseaux de neurones, les notions de logique "multivaluée" sont apparues. En effet, avec la logique floue, l'intelligence artificielle a accompli une avancée spectaculaire dans les applications techniques. En plus de la notion de raisonnement conforme à l'intelligence artificielle symbolique, cette technique permet de faire le lien entre la modélisation numérique et la représentation symbolique.

En 1965, Lotfi Zadeh a introduit la notion d'ensembles flous qui constitue une généralisation de la théorie classique des ensembles et admet des situations intermédiaires entre le tout et le rien [ZAD65]. Ces ensembles flous partent de l'idée d'appartenance à une classe, de catégories aux limites mal définies, de passage graduel d'une situation à une autre. Par ailleurs, la théorie des possibilités constitue un cadre permettant d'exploiter dans un même formalisme imprécisions et incertitudes. Ils ont montré leurs aptitudes à manipuler des connaissances entachées d'erreurs, vagues ou imprécises ainsi qu'une, le développement d'interface entre des données décrites symboliquement (avec des mots) et numériquement avec des chiffres que Zadeh en 1978 appela théorie du raisonnement approché plus connue sous le nom de logique floue.

La plupart des approches linguistiques sont basées sur le modèle de Mamdani qui consiste à formuler un ensemble de règles de décision en termes linguistiques utilisant des ensembles flous pour décrire l'action à prendre en fonction des valeurs linguistiques des variables d'entrée.

Le véritable essor des systèmes flous en automatique ne remonte qu'aux dix dernières années. Cet essor a notamment été favorisé par le succès de l'application des systèmes flous dans divers secteurs de l'industrie, particulièrement grâce à l'effort de promotion des japonais [MEN99]. Devant cette passion des japonais pour les nouvelles technologies, et plus particulièrement pour la logique floue, des groupes de recherches spécialisés, ainsi que divers développements, qu'ils soient fondamentaux, tels que les théorèmes d'approximation universelle ou technique, tels que les méthodes d'apprentissage, ont vu le jour sur tous les continents. Ces apports ont eu un impact direct sur les applications des systèmes flous à l'automatique.

Par ailleurs et traditionnellement, un contrôleur flou est conçu sans la connaissance explicite du modèle du système. Les règles linguistiques sont conçues à partir des connaissances d'un expert, ce qui conduit à plusieurs inconvénients à savoir [MAM75]:

- l'analyse du modèle du système n'est pas prise en considération,

- difficultés dans l'analyse de la stabilité globale du contrôleur flou,
- manque de prédiction sur la performance du contrôleur.

Face à ces contraintes et restrictions, des algorithmes de commande basés sur des modèles flous sont apparus dans la littérature [SUG93], [GHA97], [TLE99]. Ces algorithmes tirent profit des résultats des travaux sur la modélisation floue des systèmes physiques [TAK85], [GHA96a].

Les différentes approches de modélisation floue proposées dans la littérature, permettent de manipuler et de traiter des variables vagues, imprécises à valeur subjective et d'utiliser un raisonnement approximatif proche de la démarche humaine pour induire une décision précise et pertinente. En plus, elles permettent de cumuler l'expérience et les connaissances acquises par les opérateurs humains.

A la lumière de l'analyse développée ci-dessus, notre travail de recherche consiste à introduire deux techniques émergentes dans les processus de commande de la machine asynchrone proposés, à savoir : les réseaux de neurones artificiels (RNA) et la modélisation par logique floue (LF) pour l'amélioration des performances, en assurant dans les deux cas une commande robuste par rapport aux variations paramétriques et à toute incertitude de modélisation de la machine asynchrone.

Principales Contributions de la thèse

La commande à flux orienté a montré son efficacité dans le découplage entre les variables (couple et flux) en régime permanent. Comme le retour d'état linéarisant est connu pour la réalisation d'un découplage indépendamment du régime de fonctionnement, nous avons présenté une commande hybride combinant la commande vectorielle et celle par bouclage d'état linéarisant. Cependant il s'est avéré que ce type de commande reste sensible aux incertitudes inhérentes au système, surtout lorsqu'il s'agit des variations paramétriques et des perturbations externes. Dans ce contexte, comme plusieurs approches de commande robustes intègrent, lors de leur élaboration, les fluctuations paramétriques et les perturbations de charge injectées dans le processus, nous avons opté pour la commande à structures variables, dont la robustesse a été déjà démontrée dans divers travaux de recherche. L'inconvénient majeur de cette technique de commande est le phénomène de chattering. La réduction de ce dernier est généralement effectuée par le choix d'une relation linéaire par partie du gain de la commande discontinue. De meilleures performances sont obtenues dans le cas d'une non-linéarité de cette fonction en utilisant une nouvelle approche de fonction discontinue adoucie que nous proposons. Néanmoins, cette réduction se fait au prix d'une difficulté dans sa conception et dans sa réalisation.

Pour palier aux inconvénients et aux restrictions citées ci-dessus, l'utilisation des avantages des réseaux de neurones artificiels RNA (la rapidité de fonctionnement, l'énorme potentiel pour approximer les dynamiques non-linéaires incertaines, la capacité de générer des lois de commande robustes et adaptatives et la tolérance de certaines incertitudes lors du fonctionnement) peut s'avérer une solution prometteuse.

Ce qui nous a conduit à introduire ces réseaux de neurones dans le processus de commande hybride afin d'approximer les lois de commande réalisées par les deux régulateurs de vitesse et de flux à structures

variables. En outre, cette structure de réglage s'est avérée gourmande vis-à-vis du temps de calcul pour effectuer l'entraînement des RNAs.

Par conséquent, et pour bénéficier des avantages que procure la logique floue, et vu la complexité, la non-linéarité et les incertitudes du modèle du moteur asynchrone, nous avons contribué à la modélisation floue du moteur, afin de concevoir un modèle flou susceptible d'approximer le modèle réel et de fournir une commande robuste floue de la structure de réglage adoptée.

Pour ce faire, une reformulation du modèle du moteur a été proposée permettant d'une part, une commande décentralisée, et d'autre part, une meilleure introduction des systèmes approximatifs, en particulier les modèles flous. Cette représentation consiste à simplifier au maximum la boucle externe du processus de commande, qui est l'origine du découplage entre les variables à contrôler, tout en gardant la forme canonique du système. Par la suite, une synthèse sur les différentes directions possibles de conception des régulateurs par logique floue et des modèles flous a été effectuée nous permettant de diriger notre travail sur l'application de la modélisation floue linguistique sur le système de commande conçu précédemment. Cette technique est basée sur une connaissance préalable de l'expression analytique du modèle, et en sa conversion en une équation différentielle floue, tout en tenant compte des incertitudes inhérentes au système.

Le fruit de notre travail de doctorat s'est concrétisé par une contribution à la modélisation floue afin d'élaborer une commande à structure variable floue des systèmes non-linéaires, technique qui a été proposée par **Ben-Ghalia** en 1995 et que nous introduisons pour la première fois dans la commande des machines, plus précisément dans la structure de réglage de la machine. Cette technique de modélisation floue sert à imiter le système par une représentation floue tout en tenant compte de la description analytique grossière et des incertitudes du système.

Une fois cette technique de modélisation floue appliquée, le système global est représenté par un modèle flou susceptible de fournir une commande robuste floue qui prend en charge toutes les incertitudes paramétriques et non paramétriques, sans avoir à considérer leur forme, leur structure ou leur répartition spectrale; ce qui n'est pas le cas dans plusieurs stratégies de commande des systèmes probabilistes.

L'approche proposée offre aussi l'avantage de conserver les performances de la commande indépendamment de la complexité du modèle en faisant seulement un bon choix de partitionnement et des formes des ensembles flous des entrées / sorties.

Organisation de la thèse

Le présent travail est structuré en quatre chapitres:

- Le premier chapitre est consacré en premier lieu à la modélisation de la machine asynchrone dans un repère lié au champ tournant. Par la suite, deux types de commande classique, à savoir : vectorielle et par retour d'état linéarisant, sont introduites dans le processus de commande. Une commande hybride combinant les deux commandes précédentes est ensuite présentée.

- Dans le deuxième chapitre, la commande robuste à structures variables est introduite pour la conception des régulateurs par mode de glissement du processus de commande hybride.
- Le troisième chapitre fait l'objet d'une commande par réseaux de neurones artificiels du schéma de commande hybride dans le but d'une éventuelle réduction du phénomène de chattering.
- Dans le quatrième chapitre, le concept de la commande robuste basée sur le modèle de Ben-Ghalia est introduit et est appliqué pour la première fois dans la commande des machines électriques et ce, pour pallier aux imperfections de la modélisation classique.

Une conclusion générale donnera une synthèse du travail effectué, et résumera les principaux résultats obtenus, ainsi que les perspectives envisagées pour d'éventuelles améliorations.

Chapitre I

Commandes Vectorielle et Par Retour d'État Linéarisant de la Machine Asynchrone

Chapitre I

Commandes Vectorielle et Par Retour d'État Linéarisant de la Machine Asynchrone

I.1 INTRODUCTION

Le moteur asynchrone bien qu'ayant divers avantages comparé au moteur à courant continu, il reste tout de même assez difficile à commander, en raison de la complexité de son modèle qui présente des non-linéarités et un fort couplage entre les différentes variables. De plus, certaines de ces variables d'état (par exemple : courants et flux rotoriques) ne sont pas disponibles à la mesure directe et les paramètres internes eux aussi sont affectés par l'effet thermique et les conditions de fonctionnement, ce qui influe sur la dynamique du système à commander [CHA83], [LEO85], [FU91].

Par le passé, diverses techniques ont été élaborées pour la commande de ces machines, dans le seul but d'atteindre de meilleures performances dynamiques en poursuite et en régulation.

En effet, plusieurs types de commande classiques ont été élaborés à titre d'exemple, la commande scalaire, et qui sont conçues sur la base de la boucle ouverte et les régimes permanents de la machine. Ces techniques se sont révélées relativement médiocres lorsque des performances élevées sont requises.

Pour pallier à ces contraintes, une nouvelle technique de commande appelée "commande vectorielle" fût introduite par *BLASCHKE* en 1971, et qui a fait l'objet d'une multitude de travaux de recherche [CHA83], [LEO85], [FU91], [HOL94], [HO95], [MAT95]. En effet, de nombreux schémas de commande vectorielle directe et indirecte furent proposées [FU91], [BAR93], [BAR91], [HOL94], [KIM95], [MAH99], [BAR99], [BAR00b], [BAR01].

Le principe de cette technique est basé sur un changement de coordonnées non-linéaires et une commande par retour d'état qui permet d'une part, de réduire la complexité du modèle dynamique de la machine, et d'autre part, d'aboutir à une relation linéaire entre la vitesse et la variable de commande correspondante et ce, en maintenant l'amplitude du flux constante. Cette technique a provoqué une réelle révolution dans le domaine de la commande des machines électriques leur permettant par conséquent d'épouser un comportement dynamique similaire à celui de la machine à courant continu à excitation séparée.

Pour ce type de contrôle, les correcteurs de la famille du PID (PI, PID, IP, PD avance et retard de phase), ont été largement utilisés en raison de leur simplicité de conception, leur facilité de réglage et d'implémentation, ainsi que leurs performances acceptables [TWA93], [LIA94].

Néanmoins, cette classe de correcteurs ne donne de bons résultats que dans la mesure où les paramètres du moteur et les perturbations externes sont connus avec précision et ne varient pas en cours de fonctionnement. C'est pourquoi, la commande basée sur l'orientation du champ possèdent quelques inconvénients qui ont limité leurs performances. En effet, l'efficacité de ce type de commande dépend essentiellement de la connaissance instantanée de la position du flux rotorique, laquelle dépend de la constante de temps rotorique de la machine. Une variation de la résistance rotorique affecterait indéniablement le découplage (Couple-Flux) et par conséquent, les performances requises du processus de commande s'en verront dégradés.

De plus, un autre inconvénient est à considérer et qui concerne l'établissement de ce type de commande basée sur une approximation linéaire de la dynamique de la machine autour du point de fonctionnement. Cet inconvénient ne permet pas une bonne compensation des non-linéarités de la machine en particulier lors des régimes perturbés et des régimes transitoires.

Pour pallier à cette déficience, les recherches se sont orientées vers un autre type de commande dites non-linéaires. Parmi ces techniques, nous pouvons citer le bouclage linéarisant ou "feedback linearisation" qui consiste à transformer la dynamique non-linéaire de la machine asynchrone en un système équivalent linéaire plus ou moins découplé. Ce qui représente un avantage de taille du point de vue synthèse de commande [ISI89], [SAS89], [SIR89], [SLO91], [BOD94], [KHA96].

Ce premier chapitre est consacré à l'étude de la machine asynchrone et de la mise en œuvre de deux techniques de commande qui lui sont associées. Dans ce contexte, la première partie étant réservée à la modélisation du système de commande. Dans la seconde partie le concept de la commande vectorielle et celle linéarisante sont présentées. Par la suite, nous présenterons un schéma de commande hybride qui combine entre les deux structures de commande : vectorielle et celle par retour d'état linéarisant.

1.2 MODELISATION DE LA MACHINE ASYNCHRONE

La machine asynchrone, avec la répartition de ses enroulements et sa propre géométrie est très complexe pour se prêter à une analyse tenant compte de sa configuration exacte. Ainsi, pour obtenir son modèle mathématique, nous supposons d'une part, que les trois phases statoriques et rotoriques sont symétriques et permettent une distribution spatiale sinusoïdale des forces magnétomotrices dans un circuit magnétique supposé non saturable. D'autre part, l'effet d'encoches, l'effet de peau, les courants de Foucault ainsi que les pertes ferromagnétiques sont négligées [CHA83].

En tenant compte de ces hypothèses, pour une commande en tension la représentation d'état de son modèle dans le repère lié au champ tournant en considérant les composantes de tension (v_{ds} , v_{qs}) comme grandeurs de commande, et les variables (i_{ds} , i_{qs} , Φ_{dr} , Φ_{qr} , Ω) comme variables d'état est donnée par [FU91] :

$$\dot{x} = f(x) + g(x) \cdot u \quad (1.1)$$

où :

$$x = (x_1, x_2, x_3, x_4, x_5)^t = (i_{ds}, i_{qs}, \Phi_{dr}, \Phi_{qr}, \Omega)^t;$$

$$u = (u_1, u_2)^t = (v_{ds}, v_{qs})^t$$

$$f(x) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} -\gamma x_1 + \omega_s x_2 + \frac{k}{T_r} x_3 + p k x_4 x_5 \\ -\gamma x_2 - \omega_s x_1 + \frac{k}{T_r} x_4 - p k x_3 x_5 \\ \frac{L_m}{T_r} x_1 - \frac{1}{T_r} x_3 + (\omega_s - p x_5) \cdot x_4 \\ \frac{L_m}{T_r} x_2 - \frac{1}{T_r} x_4 - (\omega_s - p x_5) \cdot x_3 \\ \frac{pL_m}{JL_r} (x_3 x_2 - x_1 x_4) - \frac{C_r}{J} \end{pmatrix}; \quad g(x) = (g_1(x) \quad g_2(x)) = \begin{pmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_r} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

avec :

$$T_r = \frac{L_r}{R_r}; \quad \sigma = 1 - \frac{L_m^2}{L_r L_s}; \quad k = \frac{L_m}{L_r L_s - L_m^2}; \quad \gamma = \frac{1}{\sigma L_s} \left(R_s - \frac{R_r L_m^2}{L_r^2} \right)$$

$\omega_{sl} = \omega_s - p x_5$: la vitesse du glissement.

D'après cette représentation nous pouvons remarquer, que la dynamique de la machine asynchrone est non- linéaire, multivariables et fortement couplée. En effet, le couple électromagnétique C_{em} est le résultat d'une interaction complexe entre les courants imposés dans les enroulements statoriques et les courants induits dans le rotor en court-circuit. Ainsi, toute variation de ce dernier par action sur les courants statoriques se traduit par une évolution du flux induit dans le rotor. Par conséquent, pour obtenir un contrôle dynamique performant de cette grandeur, il est nécessaire de réaliser un découplage *couple-flux* par l'adjonction d'une commande adéquate à titre d'exemple nous pouvons citer la "*commande vectorielle*" appelée aussi "*commande par flux orienté*" [BLA71], [BLA72], [FU91].

1.3 COMMANDE VECTORIELLE

Le comportement dynamique de la machine asynchrone devient analogue à celui d'une machine à courant continu, dans la mesure où elle est modélisée de manière appropriée pour réaliser un découplage entre les grandeurs "*couple-flux*", obtenu en ayant recours à la commande vectorielle introduite par Blachke en 1971 [BLA71]. Le principe de cette technique consiste à orienter le vecteur flux sur l'axe d ($\Phi_{dr} = \Phi_r$; $\Phi_{qr} = 0$) dans un repère lié au champ tournant. Ainsi, l'orthogonalité des deux vecteurs (i_{qs}, Φ_r) qui en résulte assure une production optimale du couple électromagnétique (Fig :1.1) [BLA71], [BLA72], [FU91], [BAR93], [BAR91].

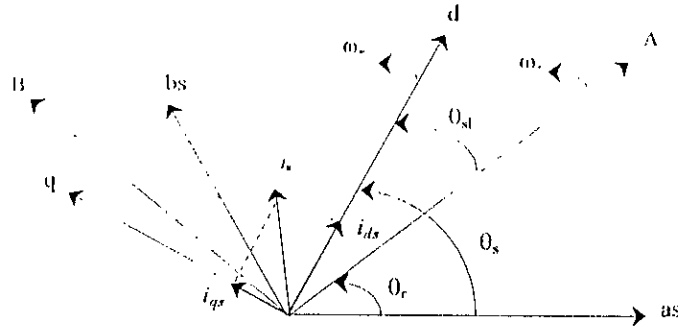


Figure I.1: Diagramme vectoriel de la machine asynchrone commandée par orientation du flux rotorique.

En substituant les conditions de l'orientation du flux rotorique ($\Phi_{qr} = 0$, $\Phi_{dr} = \Phi_r$) dans les équations du modèle (I.1), nous aboutissons au système d'équations suivant :

$$\begin{aligned}
 \frac{d}{dt} i_{ds} &= \frac{1}{\sigma \cdot L_s} \left[-R_s \cdot i_{ds} + \omega_s \cdot \sigma \cdot L_s \cdot i_{qs} - \left(\frac{L_m}{L_r} \right) \frac{d\Phi_{dr}}{dt} + v_{ds} \right] \\
 \frac{d}{dt} i_{qs} &= \frac{1}{\sigma \cdot L_s} \left[-R_s \cdot i_{qs} - \omega_s \cdot \sigma \cdot L_s \cdot i_{ds} - \frac{L_m}{T_r} \omega_s \cdot \Phi_{dr} + v_{qs} \right] \\
 T_r \frac{d}{dt} \Phi_r + \Phi_r &= L_m \cdot i_{ds} \\
 C_{em} &= \frac{p \cdot L_m}{L_r} \cdot \Phi_r \cdot i_{qs} \\
 \omega_s &= \omega_r + \frac{L_m}{T_r} \cdot \frac{i_{qs}}{\Phi_{dr}} = \omega_r + \omega_{sl}
 \end{aligned} \tag{I.2}$$

D'après le système (I.2), il est clair que si l'amplitude du flux rotorique est maintenue constante, son contrôle sera réalisé par la composante directe du courant statorique i_{ds} et le couple ne dépendra que de la composante en quadrature i_{qs} .

D'autre part, les deux premières équations de ce système présentent un couplage indésirable entre les diverses variables d'état, ce qui rend le recours à l'utilisation d'un circuit de découplage inévitable. A cet effet, il serait judicieux de découpler les deux composantes du courant statorique (i_{ds} , i_{qs}) en insérant des régulateurs de courants de type PI. Ces deux équations peuvent alors être réécrites sous la forme suivante :

$$\begin{aligned}
 v_{ds} &= \left(\frac{L_m}{L_r} \right) \frac{d\Phi_r}{dt} - \omega_s \cdot \sigma \cdot L_s \cdot i_{qs} + (R_s \cdot i_{ds} + \sigma \cdot L_s \frac{di_{ds}}{dt}) = V_{ds1} + e_q \\
 v_{qs} &= \left(\frac{L_m}{L_r} \omega_s \cdot \Phi_r + \omega_s \cdot \sigma \cdot L_s \cdot i_{ds} \right) + (R_s \cdot i_{qs} + \sigma \cdot L_s \frac{di_{qs}}{dt}) = V_{qs1} + e_d
 \end{aligned} \tag{I.3}$$

avec :

$$V_{ds1} = \left(\frac{L_M}{L_r} \right) \frac{d\Phi_r}{dt} + R_s \cdot i_{ds} + \sigma \cdot L_s \cdot \frac{di_{ds}}{dt}$$

$$V_{qs1} = R_s \cdot i_{qs} + \sigma \cdot L_s \cdot \frac{di_{qs}}{dt}$$

$$e_q = -\omega_s \cdot \sigma \cdot L_s \cdot i_{qs}$$

$$e_d = \left(\omega_r \frac{L_M}{L_r} \Phi_r + \omega_s \cdot \sigma \cdot L_s \cdot i_{ds} \right)$$

où e_d, e_q : représentent les f.e.m de perturbations partiellement liées aux courants et introduisant le couplage non linéaire.

V_{ds1} et V_{qs1} : deux nouvelles tensions totalement découplées et liées respectivement aux composantes du courant statorique (i_{ds}, i_{qs}) et qui représentent les variables à contrôler.

Ces deux nouvelles tensions sont ensuite corrigées afin de compenser l'erreur introduite par le découplage et peuvent être écrites sous la forme condensée suivante:

$$\begin{aligned} V_{ds1} &= (V_{ds} - e_q) \\ V_{qs1} &= (V_{qs} - e_d) \end{aligned} \tag{I.4}$$

Le schéma de la structure de réglage par orientation du flux est représenté à la figure (I.2).

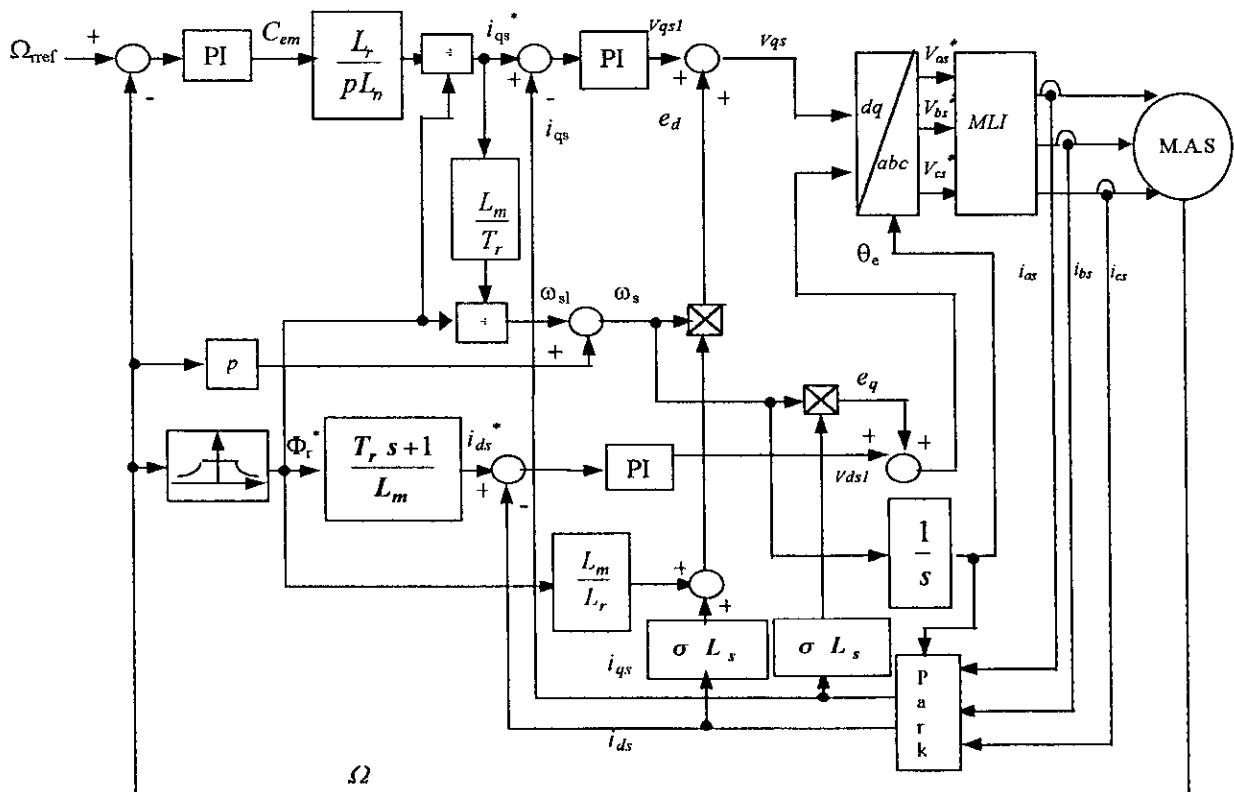


Figure I.2: Schéma bloc de la commande en tension avec régulation de vitesse et bloc de découplage.

La dynamique du système avec une machine asynchrone dont les caractéristiques sont présentées en (annexe A) est simulée pour les conditions suivantes :

- démarrage à vide avec une consigne de vitesse de 100 rad/sec et une référence du flux rotorique de 1 Wb.
- application brusque d'un couple de charge de 10 Nm à l'instant $t = 0.5$ sec
- inversion de la consigne de vitesse de 100 rad/sec à -100 rad/sec à $t = 1.5$ sec

Les paramètres K_p et K_i des régulateurs PI sont calculés autour du point de fonctionnement, comme décrits dans [FU91],[BAR93].

Les résultats de simulations obtenus sont illustrés à la figure (I.3).

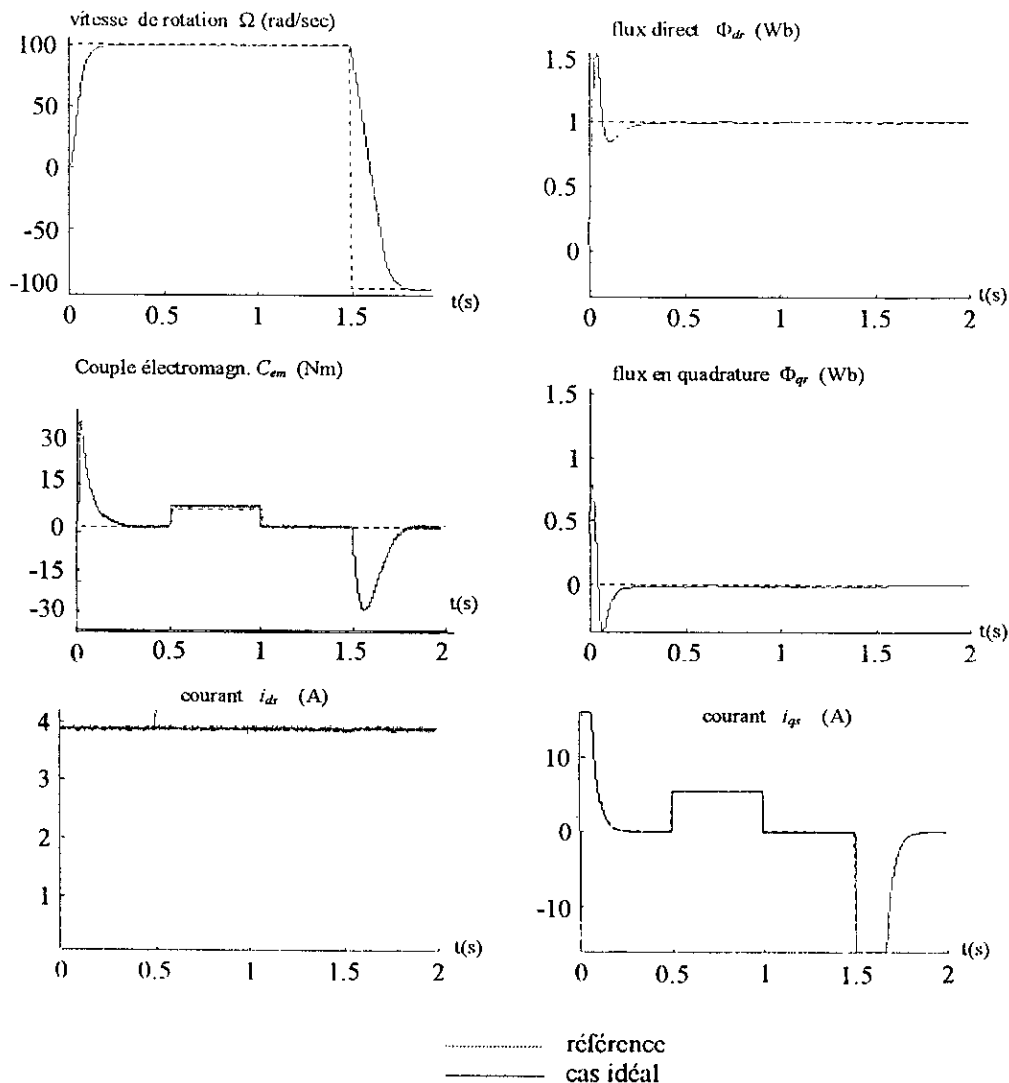


Figure I.3: Résultats de simulation pour la commande par flux orienté.

D'après ces résultats, nous remarquons que la vitesse suit sa valeur de référence sans dépassement, ni erreur statique.

La stratégie de commande vectorielle est mise en évidence par l'annulation de la composante en quadrature du flux rotorique ($\Phi_{qr} = 0$) et la poursuite par Φ_{dr} de la consigne de flux rotorique imposée à 1 Wb.

Nous constatons également que le découplage "Couple-flux" est réalisé étant donné que l'application d'un couple de charge à $t = 0.5\text{sec}$ n'affecte nullement le flux rotorique et que l'allure du courant i_{qr} est proportionnelle à celle du couple électromagnétique.

D'autre part, afin de voir l'influence des incertitudes et des variations paramétriques (en particulier, la résistance rotorique) sur le processus de commande, un test de robustesse qui se résume en deux essais est effectué:

- Le premier consiste à faire varier la résistance rotorique par rapport à la résistance nominale R_{r0} de 50% à 150% (Fig. I.4)
- Le second essai est effectué dans les mêmes conditions que précédemment avec cependant une variation des inductances de 5% (Fig. I.5).

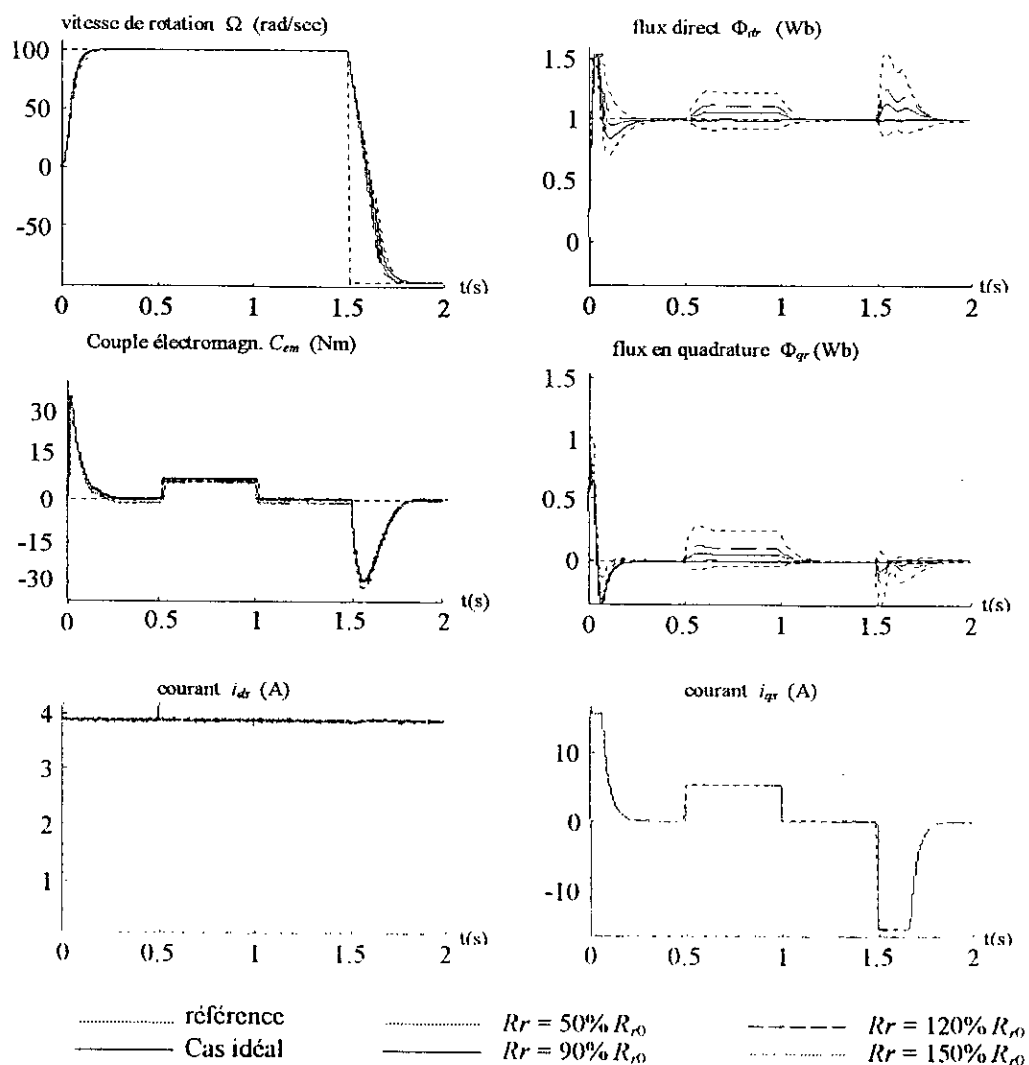


Figure I.4: Commande par flux orienté avec une variation de R_r de 50% à 150%.

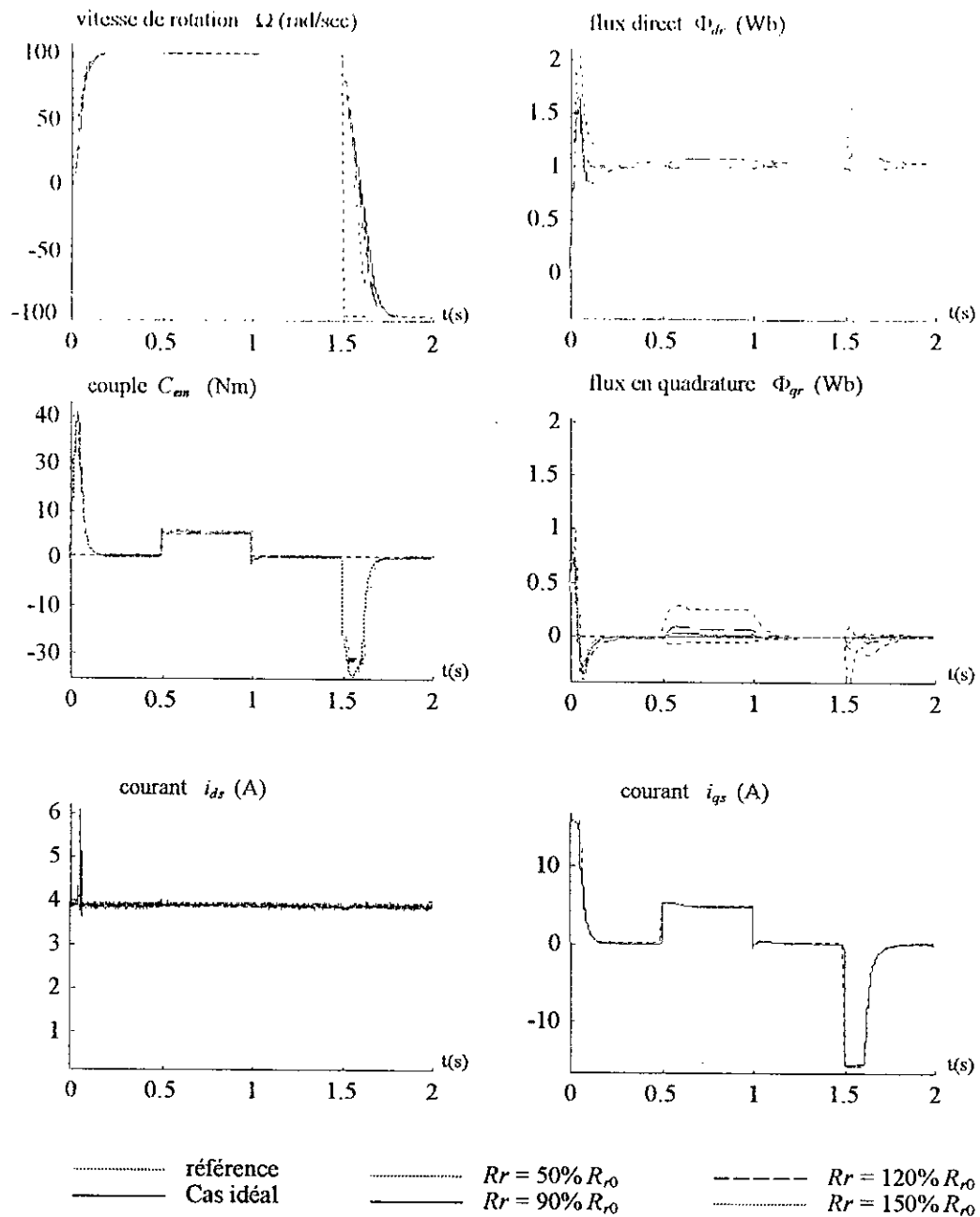


Figure I.5: Comportement de la machine lors d'une variation de R_r de 50% à 150% dans le cas où les inductances varient de 5%.

D'après les résultats de simulation obtenus, il est clair que le découplage est très sensible à la variation de la résistance rotorique. En effet, dans les deux cas la détérioration des performances dynamiques se traduit par une perte totale du contrôle vectoriel ($\Phi_{qr} \neq 0$), et est d'autant plus accentuée que la variation de R_r augmente.

Ainsi, nous pouvons dire que si la commande vectorielle permet d'avoir un découplage entre les deux grandeurs "couple-flux" de la machine et par conséquent des performances dynamiques similaires à ceux d'une machine à courant continu, son établissement basé sur une approximation linéaire de la dynamique de la machine autour du point de fonctionnement ne permet pas d'avoir une bonne compensation des non-linéarités de la machine, en particulier, lors des régimes perturbés et des régimes transitoires. Par conséquent, afin de trouver des solutions à certains problèmes de la commande vectorielle et conférer un découplage quel que soit le régime de fonctionnement de la machine le recours aux commandes non-linéaires s'avère approprié. Parmi ces techniques non-linéaires, nous pouvons citer le bouclage linéarisant ou "feedback linearisation" [ISI89], [SAB89], [SIR89], [SLO91], [BOD94], [KHA96].

I.4 COMMANDE PAR RETOUR D'ETAT LINEARISANT

Le bouclage linéarisant est une technique de commande non-linéaire dont la structure de réglage est constituée de deux boucles [ISI89], [SLO91]:

1. une première boucle interne nonlinéaire, ayant pour rôle la compensation des non-linéarités du système indépendamment des objectifs de la commande,
2. la seconde, externe, linéaire destinée à réaliser les objectifs de commande requis.

La mise en œuvre de ce type de commande s'effectue en 4 étapes successives à savoir:

1. choix des grandeurs de sortie du système,
2. calcul du degré relatif vectoriel et changement de coordonnées,
3. retour d'état non-linéaire,
4. stabilité asymptotique et poursuite de référence.

I.4.1 Application à la machine asynchrone

Étape 1 : Choix des grandeurs de sortie

Pour une commande en courant, la dynamique du stator peut être négligée et la machine est représentée par le modèle réduit du 3^{ème} ordre suivant [LAL99], [BAR03b]:

$$\dot{x} = f'(x) + g'(x) \cdot u \quad (1.6)$$

avec :

$$x = (x_3, x_4, x_5)^t = (\Phi_{dr}, \Phi_{qr}, \Omega)^t ;$$

$$u = (u_1, u_2, u_3)^t = (i_{dr}, i_{qr}, \omega_d)^t ;$$

$$f'(x) = \begin{bmatrix} f_1'(x) \\ f_2'(x) \\ f_3'(x) \end{bmatrix} = \begin{bmatrix} \frac{-x_3}{T_r} - p \cdot x_4 \cdot x_5 \\ \frac{-x_4}{T_r} + p \cdot x_3 \cdot x_5 \\ \frac{C_r}{J} \end{bmatrix}, \quad g'(x) = \begin{bmatrix} g_1'(x) & g_2'(x) & g_3'(x) \end{bmatrix} = \begin{bmatrix} \frac{L_m}{T_r} & 0 & -p \cdot x_4 \\ 0 & \frac{L_m}{T_r} & p \cdot x_3 \\ \frac{-pL_m}{JL_r} x_2 & \frac{pL_m}{JL_r} x_1 & 0 \end{bmatrix}$$

Afin de concevoir une structure de réglage des grandeurs (Φ_{dr} , Φ_{qr}) et de la vitesse Ω par bouclage linéarisant, les variables de sorties sont choisies comme [KAZ96], [BAR03b]:

$$\begin{aligned} h_1(x) &= x_3 = z_1 = \Phi_{dr} \\ h_2(x) &= x_4 = z_2 = \Phi_{qr} \\ h_3(x) &= x_5 = z_3 = \Omega \end{aligned} \quad (1.7)$$

Étape 2 : Calcul du degré relatif et changement de coordonnées

Le degré relatif r est défini comme étant le nombre de fois qu'il faudrait dériver chaque sortie $z_i = h(x)$ du système pour faire apparaître explicitement une des grandeurs de commande u .

– Sortie $h_1(x) = z_1$

Calculons la dérivée de z_1 .

$$\dot{z}_1 = L_f h_1(x) + L_{g_1} h_1(x) \cdot u_1 + L_{g_2} h_1(x) \cdot u_2 + L_{g_3} h_1(x) \cdot u_3 = \frac{-1}{T_r} \cdot x_3 - x_4 \cdot u_3 + \frac{L_m}{T_r} \cdot u_1 \quad (1.8)$$

Le degré relatif associé à la grandeur de sortie z_1 est : $r_1 = 1$

– Sortie $h_2(x) = z_2$

$$\dot{z}_2 = L_f h_2(x) + L_{g_1} h_2(x) \cdot u_1 + L_{g_2} h_2(x) \cdot u_2 + L_{g_3} h_2(x) \cdot u_3 = \frac{-1}{T_r} \cdot x_4 - x_3 \cdot u_3 + \frac{L_m}{T_r} \cdot u_2 \quad (1.9)$$

Le degré relatif correspondant à la grandeur de sortie z_2 est : $r_2 = 1$

– Sortie $h_3(x) = z_3$

$$\dot{z}_3 = L_f h_3(x) + L_{g_1} h_3(x) \cdot u_1 + L_{g_2} h_3(x) \cdot u_2 + L_{g_3} h_3(x) \cdot u_3 = \frac{pL_m}{J \cdot L_r} \cdot (x_3 \cdot u_2 - x_4 \cdot u_1) - \frac{C_r}{J} \quad (1.10)$$

Le degré relatif associé à la grandeur de sortie z_3 est : $r_3 = 1$

avec:

$$L_f h_1(x) = -\frac{x_1}{T_r} ; L_f h_2(x) = \frac{x_2}{T_r} ; L_f h_3(x) = -\frac{C_r}{J}$$

$$L_{g1} h_1(x) = \frac{L_m}{T_r} ; L_{g2} h_1(x) = 0 ; L_{g3} h_1(x) = -p \cdot x_3$$

$$L_{g1} h_2(x) = 0 ; L_{g2} h_2(x) = \frac{L_m}{T_r} ; L_{g3} h_2(x) = p \cdot x_3$$

$$L_{g1} h_3(x) = -\frac{pL_m}{J \cdot L_r} x_2 ; L_{g2} h_3(x) = \frac{pL_m}{J \cdot L_r} x_1 ; L_{g3} h_3(x) = 0$$

La somme des degrés relatifs correspondants aux sorties z_1 , z_2 et z_3 est donnée par:

$$\sum_{i=1}^3 r_i = n = 3 \quad (I.11)$$

Par conséquent, nous pouvons conclure que la linéarisation est exacte et que le système dans le nouvel espace d'état z qui en résulte s'écrit comme:

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} = \begin{pmatrix} L_f h_1(x) \\ L_f h_2(x) \\ L_f h_3(x) \end{pmatrix} + D(x) \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (I.12)$$

Où $D(x)$: est la matrice de découplage donnée par:

$$D(x) = \begin{pmatrix} L_{g1} h_1(x) & L_{g2} h_1(x) & L_{g3} h_1(x) \\ L_{g1} h_2(x) & L_{g2} h_2(x) & L_{g3} h_2(x) \\ L_{g1} h_3(x) & L_{g2} h_3(x) & L_{g3} h_3(x) \end{pmatrix} = \begin{pmatrix} L_{g1} h_1(z) & L_{g2} h_1(z) & L_{g3} h_1(z) \\ L_{g1} h_2(z) & L_{g2} h_2(z) & L_{g3} h_2(z) \\ L_{g1} h_3(z) & L_{g2} h_3(z) & L_{g3} h_3(z) \end{pmatrix}$$

Étape 3: Retour d'état non-linéaire

Pour que le retour d'état non-linéaire puisse être défini, il est impératif que la matrice de découplage soit non singulière. En effet, cette condition est vérifiée, étant donné que:

$$\det(D(x)) = \frac{L_m}{T_r} \frac{pL_m}{JL_r} (x_3^2 + x_4^2) = \frac{L_m}{T_r} \frac{pL_m}{JL_r} \Phi_r^2 \neq 0 \quad (I.13)$$

Ainsi, nous déduisons les nouvelles variable de commande (v_1 , v_2 , v_3) qui assurent le découplage et la linéarisation du système telle que :

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = D^{-1}(x) \left\{ \begin{pmatrix} -L_f h_1(x) \\ -L_f h_2(x) \\ -L_f h_3(x) \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \right\} = \frac{1}{\det(D(x))} \begin{pmatrix} \frac{pL_m}{JL_f} x_3^2 & \frac{pL_m}{JL_f} x_3 x_4 & -\frac{L_m}{L_f} x_4 \\ \frac{pL_m}{JL_f} x_3 x_4 & \frac{pL_m}{JL_f} x_4^2 & \frac{L_m}{L_f} x_3 \\ \frac{L_m L_m}{L_f T_r} x_3 & -\frac{L_m L_m}{L_f T_r} x_4 & \left(\frac{L_m}{T_r}\right)^2 \end{pmatrix} \begin{pmatrix} v_1 + \frac{1}{T_r} x_3 \\ v_2 + \frac{1}{T_r} x_4 \\ v_3 + \frac{C_r}{J} \end{pmatrix} \quad (I.14)$$

Le nouveau système d'état est par conséquent représenté par trois intégrateurs donnant trois sous-systèmes linéaires et entièrement découplés définie comme:

$$\begin{aligned} \dot{z}_1 &= \dot{x}_3 = v_1 \\ \dot{z}_2 &= \dot{x}_4 = v_2 \\ \dot{z}_3 &= \dot{x}_5 = v_3 \end{aligned} \quad (I.15)$$

Le schéma du processus de commande est illustré à la figure (I.6).

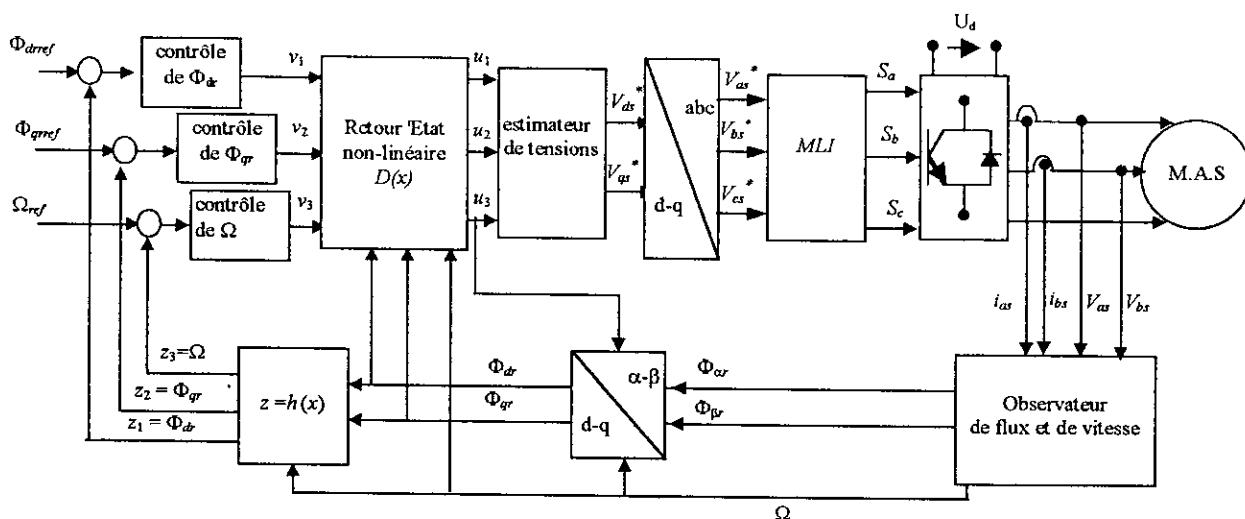


Figure I.6: Schéma de commande par retour d'état linéarisant de la machine asynchrone.

Etape 4: Stabilité asymptotique et poursuite de référence

Afin d'assurer une certaine stabilité du système, ainsi qu'une réponse rapide et bien amortie, nous avons opté pour la commande par placements de pôles. La synthèse des commandes v_1 , v_2 et v_3 est régie par les relations suivantes [LAL99], [BAR03b]:

$$\begin{aligned} v_1 &= k_{p1} \cdot (z_{1ref} - z_1) + k_{I1} \cdot \int_0^t (z_{1ref} - z_1) \cdot d\tau + \dot{z}_{1ref} \\ v_2 &= k_{p2} \cdot (z_{2ref} - z_2) + k_{I2} \cdot \int_0^t (z_{2ref} - z_2) \cdot d\tau + \dot{z}_{2ref} \\ v_3 &= k_{p3} \cdot (z_{3ref} - z_3) + k_{I3} \cdot \int_0^t (z_{3ref} - z_3) \cdot d\tau + \dot{z}_{3ref} \end{aligned} \quad (I.16)$$

avec z_{1ref} , z_{2ref} , z_{3ref} : les références des deux composantes du flux rotorique et de la vitesse respectivement.

Les coefficients k_{P1} , k_{I1} , k_{P2} , k_{I2} , k_{P3} , k_{I3} sont choisis par un placement de pôles.

Une simulation numérique a été effectuée en 4 étapes consécutives (Fig.1.7).

- un démarrage à vide du moteur sous une référence de la vitesse $\Omega_{ref} = 100\text{rd/sec}$ avec des références du flux $\Phi_{qr} = 0.3\text{ Wb}$ et $\Phi_{dr} = 0.8\text{Wb}$,
- diminution de cette référence à une valeur $\Phi_{dr} = 0.4\text{Wb}$ entre les instants $t = 0.5\text{sec}$ et $t = 1\text{sec}$,
- rebasculement de cette dernière référence vers sa valeur initiale $\Phi_{dr} = 0.8\text{Wb}$ à $t = 1\text{sec}$,
- application d'une inversion de vitesse (100 rd/sec , -100rd/sec) à $t = 1.5\text{sec}$.

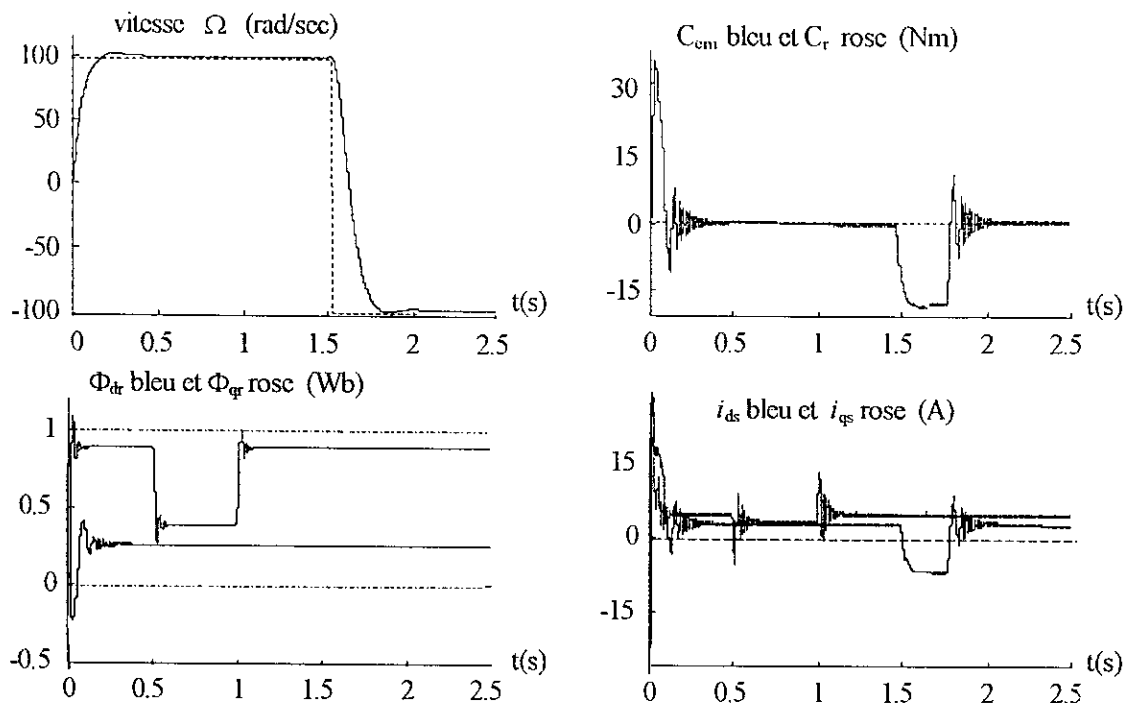


Figure 1.7: Résultats de simulation de la commande par placement de pôles dans le cas idéal

D'après ces résultats, il est clair que la brusque variation de l'une ou l'autre des trois références n'affecte nullement les deux autres. Ce qui nous permet d'affirmer qu'un découplage total entre les trois grandeurs de la machines (Φ_{dr} , Φ_{qr} , Ω) est réalisé.

1.5 COMMANDE HYBRIDE LINEARISANTE ET A ORIENTATION DU FLUX ROTORIQUE

L'introduction de la condition d'orientation du flux rotorique ($\Phi_{qr} = 0$ et $\Phi_{dr} = \Phi_r$) dans le modèle (I.12), nous permet d'obtenir une commande hybride associant la commande vectorielle et celle non-linéaire par retour d'état. L'avantage apporté par une telle technique réside d'une part, dans la réduction

de la complexité des deux schémas de commande représentés aux figures (I.2) et (I.6), tout en gardant des performances similaires à ceux de la machine à courant continu, et surtout un découplage *Couple-flux* non restreint à la plage de vitesses inférieures à celle nominale mais effectif sur toute la plage de variation de la vitesse de rotation du moteur [BAR03b]. Cette commande hybride est régit par le système d'équation suivant:

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_3 \end{pmatrix} = \begin{pmatrix} \frac{-1}{T_r} z_1 \\ \frac{-C_r}{J} \end{pmatrix} + \begin{pmatrix} \frac{L_m}{T_r} & 0 \\ 0 & \frac{pL_m}{J \cdot L_r} z_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_3 \end{pmatrix} = \begin{pmatrix} \frac{-1}{T_r} z_1 \\ \frac{-C_r}{J} \end{pmatrix} + D^*(x) \begin{pmatrix} u_1 \\ u_3 \end{pmatrix} \quad (I.17)$$

Le retour d'état linéarisant appliqué est défini par:

$$\begin{pmatrix} u_1 \\ u_3 \end{pmatrix} = D^*(x)^{-1} \left\{ \begin{pmatrix} -L_f h_1(x) \\ -L_f h_3(x) \end{pmatrix} + \begin{pmatrix} v_1 \\ v_3 \end{pmatrix} \right\} \quad (I.18)$$

Par conséquent, le système résultant est décrit par les équations suivantes:

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_3 &= v_3 \end{aligned} \quad (I.19)$$

Le schéma de commande de cette association entre les deux commandes, linéarisante et vectorielle, est illustré à la figure (I.8).

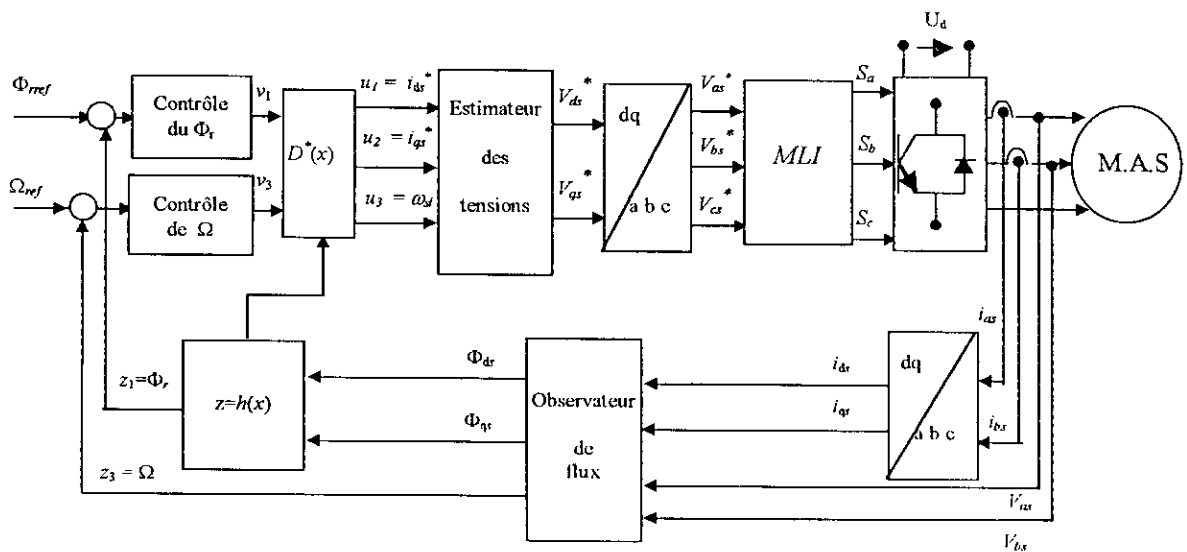


Figure I.8: Schéma de la commande hybride: linéarisante et vectorielle.

De même que pour la section (& I.4.1), la synthèse des commandes v_1 et v_3 est régie par:

$$\begin{aligned}
 v_1 &= k_{p1} \cdot (z_{1ref} - z_1) + k_{i1} \cdot \int_0^t (z_{1ref} - z_1) \cdot d\tau + \dot{z}_{1ref} \\
 v_3 &= k_{p3} \cdot (z_{3ref} - z_3) + k_{i3} \cdot \int_0^t (z_{3ref} - z_3) \cdot d\tau + \dot{z}_{3ref}
 \end{aligned}
 \tag{1.21}$$

avec : z_{1ref}, z_{3ref} : les références du flux rotorique et de la vitesse respectivement.

$k_{p1}, k_{i1}, k_{p3}, k_{i3}$: sont eux aussi choisis par un placement de pôles.

Une simulation numérique de la structure de réglage hybride a été effectuée dans les mêmes conditions que ceux de la section (I.3). Les résultats obtenus sont représentés aux figures (I.9 à I.10).

La figure (I.9) montre d'une part que le découplage entre les deux grandeurs couple et flux rotorique est réalisé et d'autre part, que l'orientation du flux rotorique sur l'axe "d" est mise en évidence par la poursuite de la référence du flux rotorique imposée à 1Wb par la composante directe Φ_{dr} et par l'annulation de Φ_{qr} .

La figure (I.10) montrent clairement que la commande par placements de pôles est sensible à la variation des paramètres de la machine (la résistance et l'inductance rotoriques). En effet, dans ces deux cas, une nette détérioration du flux rotorique est notée et est d'autant plus accentuée que la variation de R_r augmente. Par contre, la vitesse et le couple ne sont pas très affectés par les incertitudes paramétriques inhérentes à la machine.

Deux autres essais ont été également effectués dans les mêmes conditions précédentes avec cependant une non estimation du couple résistant dans le premier essai avec variation de R_r (50 à 150% R_{r0}) (Fig.I.11) et une estimation de ce dernier ($C_R = 60$ à 140% C_{R0}) dans le second avec variation de $R_r = 90\%$ R_{r0} (Fig.I.12).

Les performances du système obtenues sont eux aussi détériorés plus particulièrement ceux de la vitesse et du couple C_{em} dues à la dépendance de la vitesse par rapport au couple résistant.

En résumé, nous pouvons dire que cette technique de réglage par placements de pôles s'est avérée sensible à toute variation des paramètres internes de la machine et perturbations externes qui peuvent éventuellement survenir lors du fonctionnement du système.

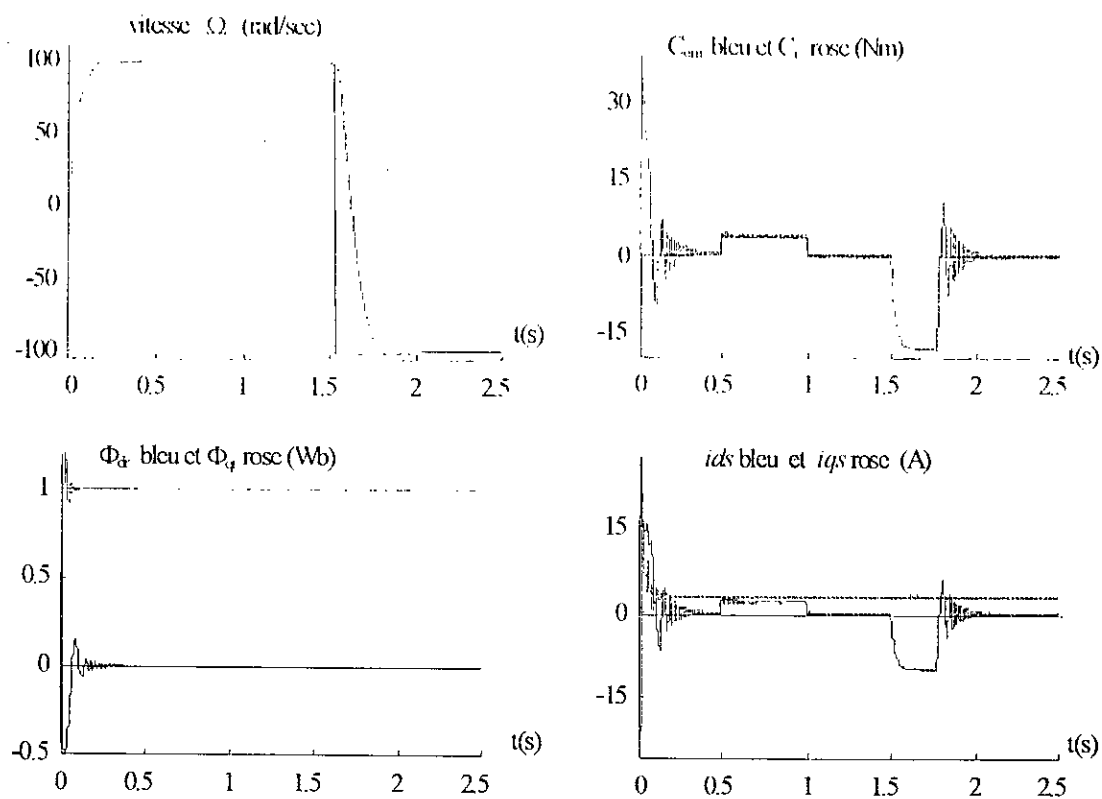
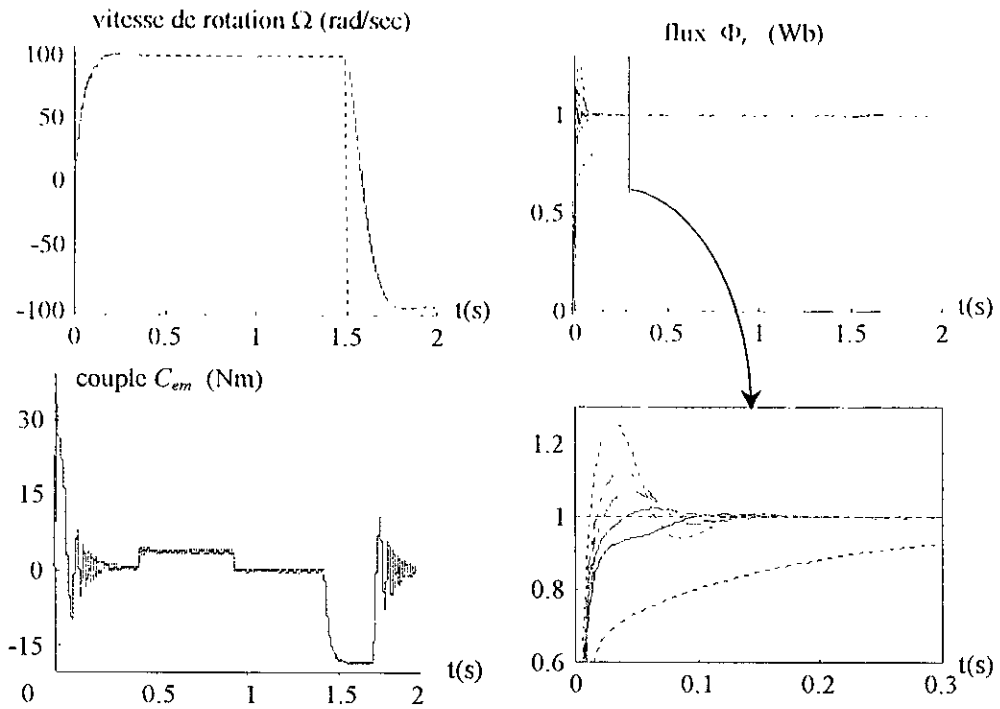
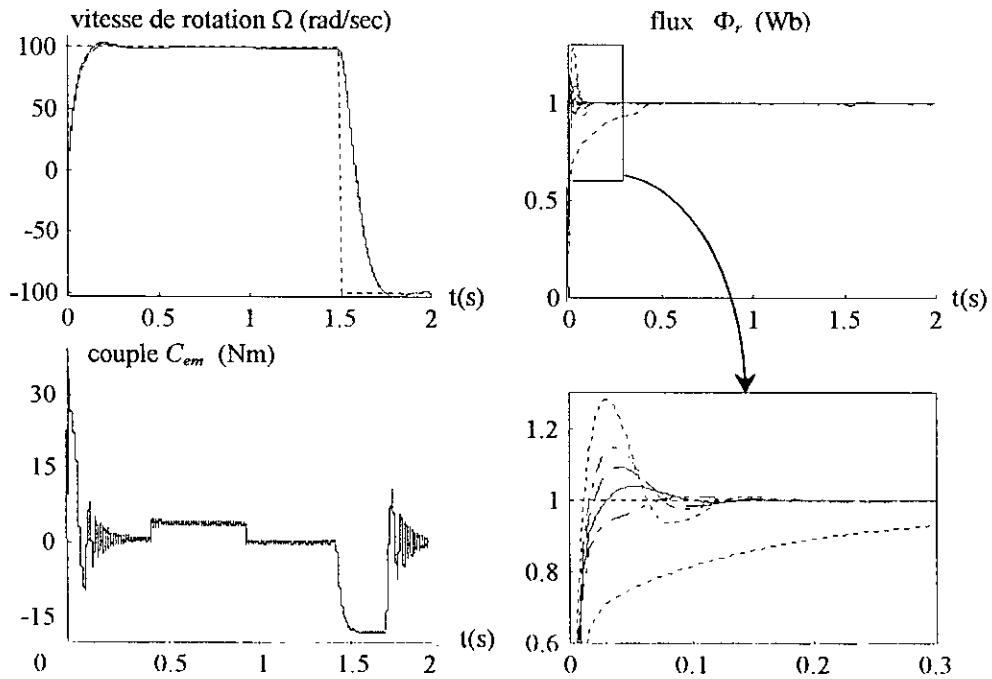


Figure I.9: Résultats de simulation de la commande par placement de pôles du moteur dans le cas idéal



a) Les inductances sont connues avec certitude



a) Les inductances sont connues à 5% près

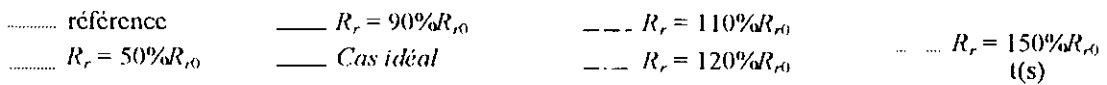


Figure I.10: Résultats de simulation pour une variation de la résistance R_r avec estimation de la charge C_r .

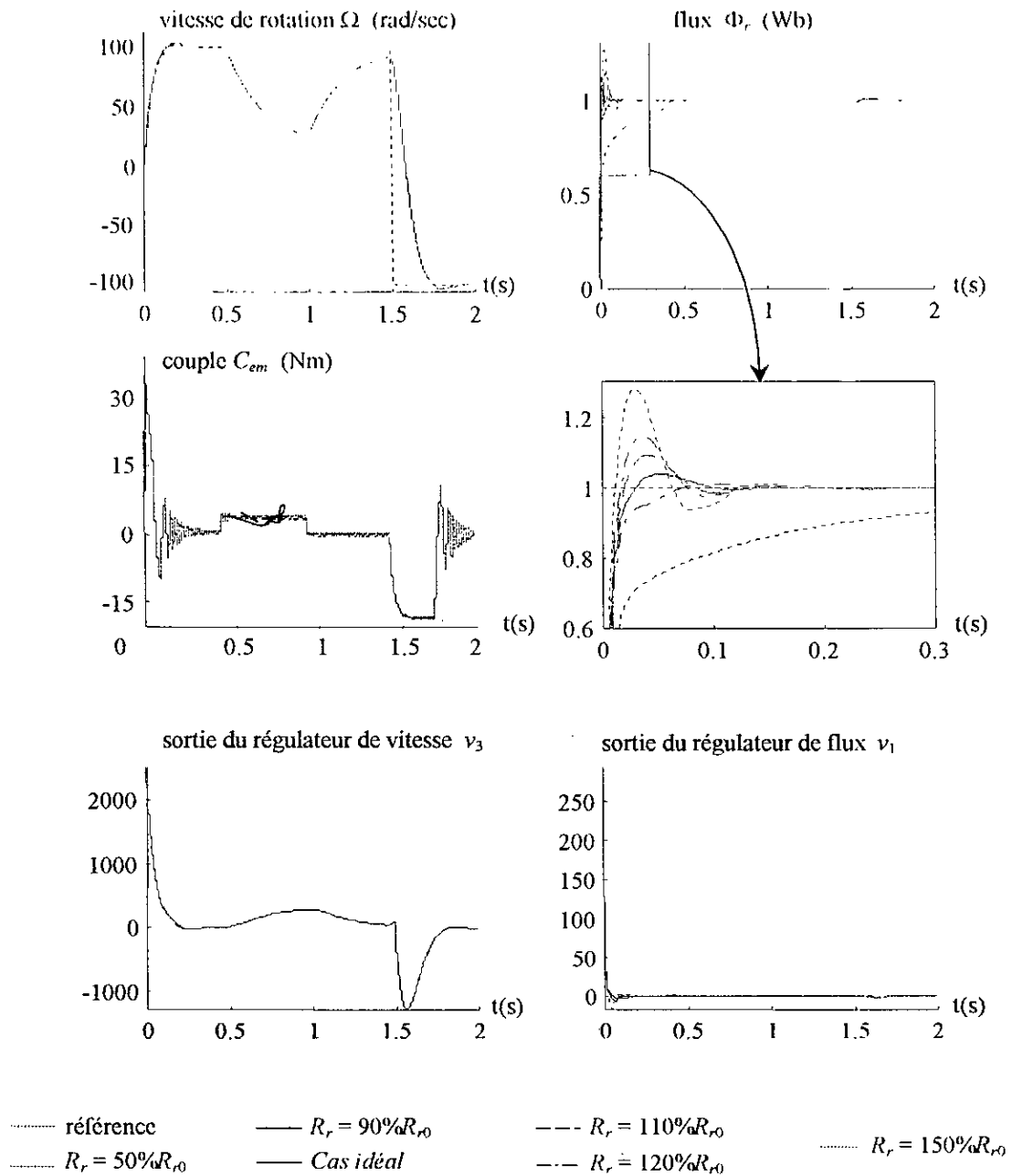


Figure I.11 : Résultats de simulation pour une variation de la résistance R_r sans estimation de la charge (les inductances sont connues à 5% près)

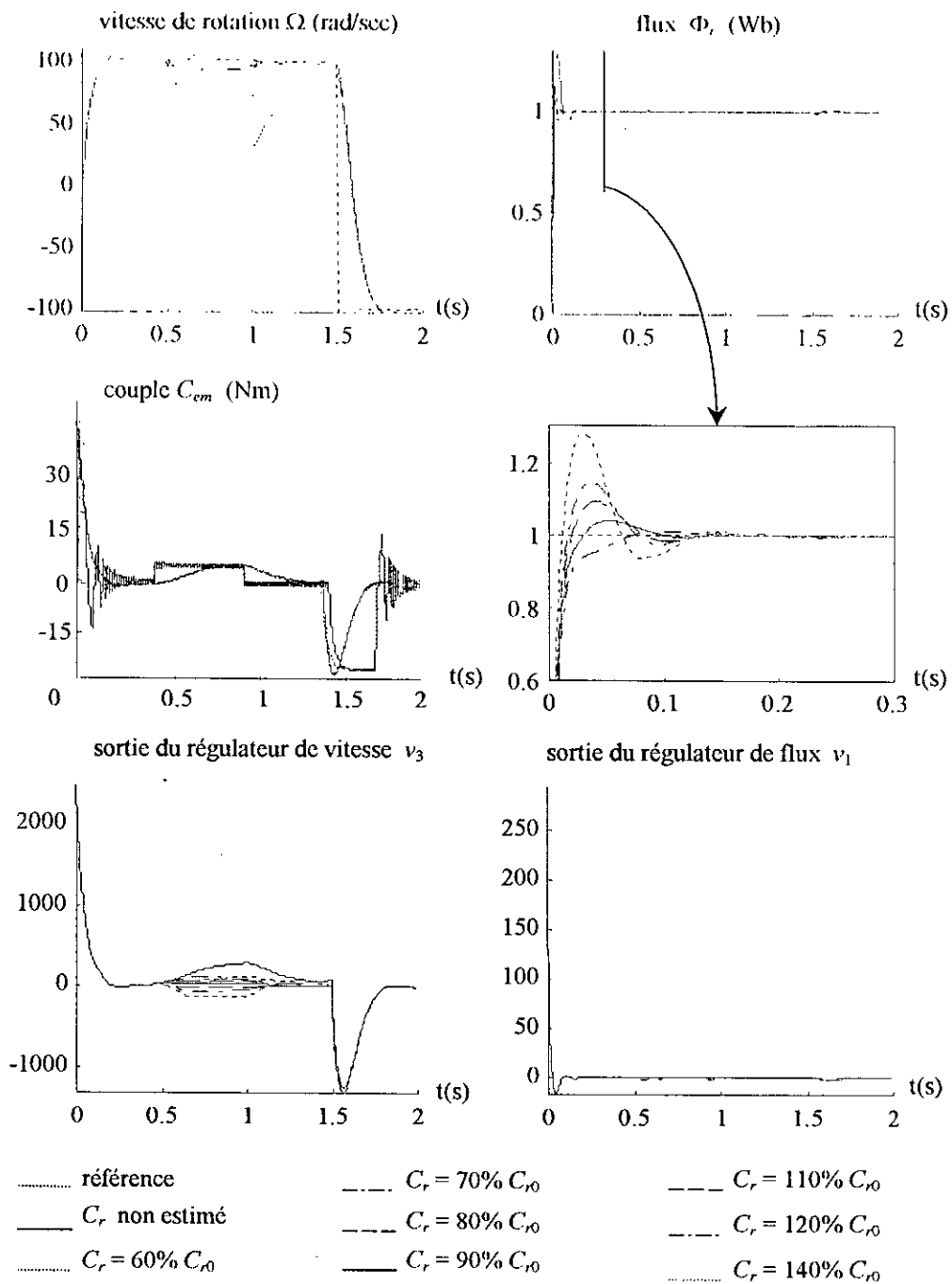


Figure I.12 : Résultats de simulation pour une variation de la résistance $R_r = 90\% R_{r0}$ et estimation de C_r (les inductances sont connues à 5% près).

1.6 CONCLUSION

Dans ce chapitre nous avons présenté le principe de la commande vectorielle, et ce après avoir modélisé la machine asynchrone en utilisant la transformation de Park dans le repère lié au champ tournant. Nous avons montré que la stratégie de commande par flux orienté présente l'inconvénient d'être très sensible aux variations des paramètres du moteur, en particulier la résistance et l'inductance rotorique qui sont souvent difficiles à identifier et varient avec l'état de fonctionnement du moteur, à cause de phénomènes divers tels que l'effet de peau, la variation de la température ou la saturation magnétique. Une erreur entre les paramètres utilisés dans l'établissement du modèle utilisé et les paramètres réels se traduit par une mauvaise orientation du flux rotorique sur le repère choisi. De plus, comme découplage total entre le couple et le flux rotorique n'est possible que pour un fonctionnement à flux constant, une commande par retour d'état linéaire a été appliquée.

Le bouclage linéarisant transforme le système original en un système équivalent linéaire et découplé à condition que la matrice de découplage soit non singulière. Si cette condition est violée, il faudra redéfinir les sorties, de sorte que la matrice soit inversible, ou redéfinir de nouvelles entrées qui ne sont rien d'autre que les dérivées des entrées du système. Néanmoins, par un souci d'orientation du flux sur l'axe d dans un repère lié au champ tournant, une commande hybride combinant la commande vectorielle et celle par bouclage linéarisant a été proposée. Cette technique concilie entre les performances des deux commandes d'origine en permettant en plus d'avoir un schéma de commande plus simple à implémenter. Néanmoins, elle nécessite la disponibilité de tous les états et ne garantit pas une robustesse face à des incertitudes paramétriques et perturbations externe. Pour pallier à ces inconvénients, nous faisons recours à une commande robuste dite à structure variables qui fera l'objet du chapitre suivant.

Chapitre II

Commande Hybride à Structures Variables de la Machine Asynchrone

*Chapitre II**Commande Hybride à Structures Variables de la Machine Asynchrone***II.1 INTRODUCTION**

La commande à structures variables (CSV) a été développée dans les années soixante par l'équipe du professeur Emelyanov [EME62] à partir des études théoriques du mathématicien Phillipov [FIL60].

Cette technique fut d'abord développée pour un système linéaire du second ordre, et depuis, le spectre de son utilisation fut élargi à une plus grande classe de systèmes comprenant les systèmes linéaires [UTK92], [UTK93], [UTK99], non-linéaires [CAR88], [GAO93], discrets [YOU96], multivariables [GAO93],...etc.

Les concepts de ce type de commande furent étendus, du problème de stabilisation à celui de la poursuite de trajectoires [HAS85], ou encore celui de modèles [RAC96] à la commande adaptative et optimale [HSU89], [COS90], [UTK92], [UTK93], [UTK99]. En raison des nombreux avantages de la commande à structures variables, tels que la robustesse, la rapidité de réponse, et la simplicité de mise en œuvre, ce type de commande semble tout à fait approprié pour les moteurs asynchrones très utilisés dans l'industrie pour lesquels de hautes performances sont requises. Par ailleurs, comme ces moteurs possèdent des paramètres pouvant varier en cours de fonctionnement et que les perturbations externes ne sont pas toujours connues, la commande devra rapidement les prendre en charge pour éviter une dégradation des performances. En effet, avec de tels systèmes de réglage, la commande commute d'une structure à une autre permettant d'amener le point figuratif de l'évolution du système vers une hypersurface de glissement et le contraindre à la suivre, ce qui constitue ce que nous appelons un régime glissant.

Malheureusement, bien qu'ayant divers avantages, cette commande possède également un inconvénient majeur qui a limité sa mise en œuvre au début de son apparition. En effet, si dans la théorie tout semble possible et parfait, dans la pratique, les imperfections telles que les retards de commutation et l'hystérésis qui en découlent engendrent des oscillations autour de la surface de glissement pouvant exciter des dynamiques non modélisées, soumettant alors les organes de commande à une intense activité, ce qui peut être néfaste pour les actionneurs particulièrement les composants électroniques qui constitue le système entre autres ceux de l'étage onduleur. Ce phénomène est connu sous le nom de chattering ou broutement.

Diverses techniques ont été proposées pour pallier à cet inconvénient, et dont les plus fréquemment utilisées consistent à approximer la fonction discontinue par une fonction continue au voisinage de la surface de commutation [HUN91]. Néanmoins, la réduction du chattering se fait toujours au prix d'une perte de précision.

Ce chapitre est entièrement consacré à la commande à structures variables. A cet effet, nous introduisons les principes fondamentaux de ce type de commande, que nous appliquons au processus de commande hybride de la machine asynchrone présentée au chapitre I. Au préalable, des correcteurs classiques de type relais sont d'abord utilisés. Par la suite et afin de réduire le phénomène de chattering engendré, nous faisons recours à l'utilisation d'une nouvelle approche de commande adoucie qui consiste à faire des approximations continues de la composante discontinue de la logique de commande.

II.2 LES SYSTEMES A STRUCTURES VARIABLES

Contrairement aux correcteurs par retour d'état où la commande est fixe, les systèmes à structures variables possèdent une commande susceptible de commuter à tout instant d'une structure à une autre. Le fait de changer de structure et de basculer de l'une à l'autre, permet de combiner entre les avantages de chacune d'elles, et mieux encore, de doter le système de nouvelles propriétés. Ainsi, un système asymptotiquement stable peut être obtenu à partir de structures instables.

Cependant, le problème de commande qui en résulte dans ce cas se résume à la manière de déterminer les paramètres de chacune de ses structures et à définir la logique de commutation.

Le schéma représentatif de ce type de systèmes à structures variables est représenté à la figure (II-1).

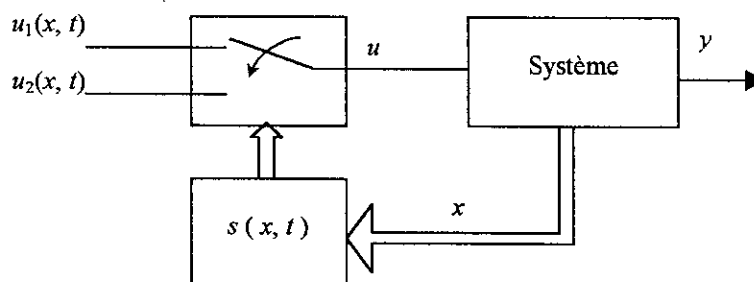


Figure II.1: Aspect général d'une commande à structures variables.

II.3 CONCEPTION DES REGULATEURS A STRUCTURES VARIABLES

Considérons un système multi-entrées / multi-sorties (MIMO) décrit par la forme matricielle suivante :

$$\dot{x} = f(x) + g(x) \cdot u \quad (\text{II.1})$$

où :

$$x = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} ; f(x) = \begin{pmatrix} f_1(x) \\ \dots \\ f_n(x) \end{pmatrix} ; g(x) = \begin{pmatrix} g_{11}(x) & \dots & g_{m1}(x) \\ \dots & & \dots \\ g_{1n}(x) & \dots & g_{mn}(x) \end{pmatrix} ; u = \begin{pmatrix} u_1(x) \\ \dots \\ u_m(x) \end{pmatrix}$$

La mise en œuvre des contrôleurs à structures variables d'un tel système s'effectue en trois étapes successives:

1. Choix de la surface de glissement,
2. établissement des conditions d'existence des modes glissants et de convergence vers la surface de glissement,
3. logique de commutation.

II.3.1 Choix de la surface de glissement

D'après le système d'équations (II.1), nous pouvons remarquer qu'il peut exister jusqu'à m surfaces de glissement relatives à chaque commande u_i . Leur choix doit être fait de manière à ce que leur intersection, qui représente en réalité la surface de glissement s permet d'avoir le comportement désiré du système ainsi qu'une bonne poursuite de cette surface.

La surface de glissement s peut être définie par:

$$s = \bigcap_{i=1}^m s_i \quad (\text{II.2})$$

$$s_i = \{ x \in \mathbb{R}^n / s_i(x) = 0 \} ; i = 1, \dots, m$$

avec :

$$s(x) = \begin{pmatrix} s_1(x) \\ \dots \\ s_m(x) \end{pmatrix} \quad (\text{II.3})$$

s_i : surface de glissement correspondant à l'entrée u_i

$s_i(x)$: fonction de commutation

$s(x)$: vecteur des fonctions de commutation

Le degré relatif de chaque surface $s_i(x)$ doit être égal à 1, ce qui signifie que la commande u_i doit apparaître dès la première dérivée de $s_i(x)$.

II.3.2 Condition d'existence du mode de glissement

Il est important de déterminer la condition dite d'atteignabilité, avec laquelle l'état x atteint et glisse sur la surface considérée. Pour satisfaire cette dernière, une fonction de Lyapunov est utilisée régie par la relation suivante:

$$V(x, t) = \frac{1}{2} s'(x) \cdot s(x) \quad (\text{II.4})$$

La condition d'atteignabilité de la surface de glissement s est définie comme:

$$\dot{V}(x,t) = [s(x)]^t \cdot \dot{s}(x) = \sum_{i=1}^m s_i(x) \cdot \dot{s}_i(x) < 0, \quad \text{pour tout } x \in \mathbb{R}^n - s \quad (\text{II.5})$$

En se basant sur cette condition, deux principales techniques de commande ont été élaborées, à savoir la technique de Filippov [FIL60], et celle de la commande équivalente proposée par V.I.Utkin [UTK93] qui fera l'objet du présent chapitre.

II.3.3 Choix des grandeurs de commande

En ce qui concerne la conception de la loi de commande à structures variables u_i ($i = 1, \dots, m$), elle est donnée par la relation suivante:

$$u_i(x,t) = \begin{cases} u_i^+(x,t) & \text{si } s_i(x) > 0 \\ u_i^-(x,t) & \text{si } s_i(x) < 0 \end{cases} \quad (\text{II.6})$$

Le rôle de ces lois de commande est de faire en sorte que chaque vecteur d'état x , se trouvant en dehors de la surface de glissement régie par la condition $s(x) = 0$, soit attiré vers cette même surface en un temps fini. Une fois cette surface atteinte, le mode de glissement a lieu, et les dynamiques désirées du système sont effectives. Par conséquent, le système devient insensible aux incertitudes et variations paramétriques ainsi qu'aux perturbations externes.

II.3.4 La commande équivalente

La loi de commande u qui correspond à ce type de technique est définie comme:

$$u = u_{eq} + u_n \quad (\text{II.7})$$

où:

u_{eq} : la commande équivalente,

u_n : la commande d'atteignabilité.

Ces deux commandes peuvent être obtenues à partir du développement de la dérivée de la loi de commutation $\dot{s}(x,t)$ comme suit [YOU78], [UTK93], [YOU96], [YOU98], [YOU99]:

$$\dot{s}(x,t) = \left(\frac{\partial s}{\partial x} \right)^t \cdot \dot{x} + \frac{\partial s}{\partial t} = \left(\frac{\partial s}{\partial x} \right)^t \cdot \{ f(x) + g(x) \cdot u \} + \frac{\partial s}{\partial t} \quad (\text{II.8})$$

Ainsi, la loi de commande u s'écrit comme:

$$u = \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \left\{ \dot{s}(x,t) - \left(\frac{\partial s}{\partial x} \right)^t f(x) - \frac{\partial s}{\partial t} \right\} \quad (\text{II.9})$$

avec la condition d'existence régit par:

$$\text{Det} \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\} \neq 0 \quad (\text{II.10})$$

En se basant sur l'équation (II.8), la commande équivalente qui correspond au régime glissant idéal et qui vérifie la condition $\dot{s}(x, t) = 0$ est donnée par :

$$u_{eq} = - \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \left\{ \left(\frac{\partial s}{\partial x} \right)^t f(x) + \frac{\partial s}{\partial t} \right\} \quad (\text{II.11})$$

Par conséquent, l'équation d'état du système en régime glissant est obtenue en remplaçant la commande u_{eq} définie par l'équation (II-11) dans le système (II.1) :

$$\dot{x} = \left\{ I - g(x) \cdot \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \left(\frac{\partial s}{\partial x} \right)^t \right\} f(x) - g(x) \cdot \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \left(\frac{\partial s}{\partial x} \right)^t \quad (\text{II.12})$$

Etant donné qu'en pratique le système réel présente des incertitudes paramétriques, non paramétriques et structurelles, et comme la phase d'atteignabilité représente une partie importante dans les dynamiques transitoires du système à contrôler, la commande équivalente se trouve être notoirement insuffisante à forcer la variable d'état à atteindre la surface de glissement. Il est de ce fait important de spécifier une loi de commande susceptible de contraindre le système à suivre une dynamique désirée. La loi de commande u , dans ce cas, peut être exprimée par [GHA95a], [GHA95b]:

$$u = u_{eq} + u_n = \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \left\{ \dot{s}_d(x, t) - \left(\frac{\partial s}{\partial x} \right)^t f(x) - \frac{\partial s}{\partial t} \right\} \quad (\text{II.13})$$

avec:

$$u_n = \left\{ \left(\frac{\partial s}{\partial x} \right)^t \cdot g(x) \right\}^{-1} \cdot \dot{s}_d(x, t) : \text{la loi de commande discontinue;}$$

$\dot{s}_d(x, t)$: la dynamique désirée de la fonction de commutation.

Le plus souvent, la loi de commande discontinue u_n est de type relais régi par la relation suivante:

$$\dot{s}_d(x) = -K \cdot \text{sgn}(s(x)) \quad (\text{II.14})$$

$K = \text{diag}[K_1, \dots, K_m]$: la matrice diagonale de gains positifs.

sgn : un champ de vecteurs contenant des fonctions signe

$$s(x) = \begin{pmatrix} s_1(x) \\ \dots \\ s_m(x) \end{pmatrix}$$

Il est à noter que, pour chaque fonction de commutation $s_i(x)$, ($i = 1, \dots, m$), la vitesse de convergence ainsi que la robustesse de la commande à structures variables sont fortement liées au gain K , [GHA95a], [GHA95b].

II.3.5 Cas des systèmes mis sous forme canonique

Le système ainsi étudié est mis sous forme canonique par blocs. Chaque bloc représente un sous système mono-entrée / mono-sortie défini par l'équation différentielle suivante:

$$\dot{x}^i = f^i(x) + g^i(x) \cdot u_i(x) \quad ; \quad i = 1, \dots, m \quad (\text{II.15})$$

avec :

$$x^i = \begin{pmatrix} x_1^i \\ \dots \\ x_{r_i}^i \end{pmatrix} \quad ; \quad f^i(x) = \begin{pmatrix} f_1^i(x) \\ \dots \\ f_{r_i}^i(x) \end{pmatrix} = \begin{pmatrix} x_2^i \\ \dots \\ f_{r_i}^i(x) \end{pmatrix} \quad ; \quad g^i(x) = \begin{pmatrix} g_1^i(x) \\ \dots \\ g_{r_i}^i(x) \end{pmatrix} = \begin{pmatrix} 0 \\ \dots \\ g_{r_i}^i(x) \end{pmatrix}$$

et $\sum_{i=1}^m r_i \leq n$

i : l'indice de l'entrée

r_i représente le degré relatif correspondant à l'entrée u_i

$$x^i \in \mathbb{R}^{r_i} \quad ;$$

$$f^i(x) \in \mathbb{R}^{r_i} \quad ; \quad g^i(x) \in \mathbb{R}^{r_i \times 1}$$

$x \in \mathbb{R}^n$ le vecteur d'état ; n : étant la dimension de tout le système.

Généralement, les surfaces de glissement sont choisies comme étant des hyperplans, et ce pour des raisons de stabilité et de simplicité. J.J.Slotine a proposé une équation de forme générale pour la détermination de la surface de glissement $s_i(x^i)$ qui assure la convergence donnée par [SLO84]:

$$s_i(x^i) = k_1 e_1 + k_2 e_2 + \dots + k_{r_i} e_{r_i} \quad (\text{II.16})$$

avec : $e_j = x_j^{*i} - x_j^i \quad ; \quad j = 1, \dots, r_i$

$$(x_1^{*i}, \dots, x_{r_i}^{*i})' : \text{constitue la trajectoire de poursuite du } i^{\text{ème}} \text{ sous-système.}$$

k_j étant les coefficients de l'hyperplan.

L'équation (II.16) peut être réécrite de la manière suivante :

$$s(x) = k_1 \cdot e + k_2 \cdot \dot{e} + \dots + k_{r_i} \frac{d^{(r_i-1)}}{dt} e \quad (\text{II.17})$$

Les coefficients k_j ($j = 1, \dots, r_i$) sont choisis de telle manière que le polynôme $\Gamma(p)$ défini comme:

$$\Gamma(p) = k^{r_i} + k_{r_i} p^{(r_i-1)} + \dots + k_1 \quad (\text{II.18})$$

soit un polynôme de Hurwitz. Par conséquent, une fois que le système est sur la surface, sa dynamique est asymptotiquement stable et converge de manière exponentielle.

En dehors de la surface, la condition d'atteignabilité (II.5) doit être vérifiée pour chaque sous système :

$$s_i(x) \cdot \dot{s}_i(x) < 0 \quad ; \quad i = 1, \dots, m \quad (\text{II.19})$$

En dérivant la surface $s(x)$, nous obtenons l'équation suivante :

$$\dot{s}_i(x^i) = k_1 \cdot \dot{e}_1 + k_2 \cdot \dot{e}_2 + \dots + k_{r_i} \cdot \dot{e}_{r_i} \quad (\text{II.20})$$

La substitution de la relation (II.1) dans l'équation (II.16) nous ramène à :

$$\dot{s}_i(x^i) = k_1 \cdot e_2 + k_2 \cdot e_3 + \dots + k_{r_i-1} \cdot e_{r_i} + \dot{x}^{*i} - (f(x) + g(x) \cdot u_i) \quad (\text{II.21})$$

avec $k_{r_i} = 1$

Ce qui nous permet d'aboutir à l'équation définissant la loi de commande suivante :

$$u_i = \left(k_1 \cdot e_2 + k_2 \cdot e_3 + \dots + k_{(r_i-1)} \cdot e_{r_i} + \dot{x}^{*i} - f(x) - \dot{s}_{id}(x^i) \right) \div g(x) \quad (\text{II.22})$$

Dans ce cas, la dynamique désirée assurant la condition d'atteignabilité (II.19) est choisie comme suit :

$$\dot{s}_{id}(x) = -K \cdot \text{sgn} (s_i (x^i)) \quad (\text{II.23})$$

Malgré les avantages de la commande à structures variables de type relais ainsi définie, celle-ci souffre de deux problèmes majeurs.

1. Le premier réside dans le fait que la propriété de robustesse n'est réalisée que si le système est dans la surface de glissement ou plus exactement dans une plage limitée d'incertitudes [UTK92], [UTK93], [UTK96]. Ainsi, le comportement des états se trouvant en dehors de cette zone devient sensible aux incertitudes paramétriques et aux perturbations externes. De plus, le système peut ne pas converger asymptotiquement, ou encore converger en un temps très grand, et la commande perdra par conséquent ses avantages [ISH93].
2. Le second problème concerne l'apparition d'un phénomène néfaste appelé : *phénomène du chattering*.

En effet, dans la conception et l'analyse de la commande à structures variables, il est supposé que la fréquence de commutation entre les structures est infinie. Ce qui est inconcevable en pratique pour plusieurs raisons, parmi lesquelles nous pouvons citer, l'existence du temps de retard lié au calcul de la loi de commande, et la limitation physique des organes de commande qui ne peuvent pas suivre la commutation théorique d'une structure à une autre. Comme conséquence de cet inconvénient, le système va commuter autour de la surface au lieu de glisser dessus comme le montre la figure (II.2). Ce phénomène est indésirable car il engendre l'excitation des dynamiques à hautes fréquences qui ne figurent pas dans la modélisation du système à contrôler [DEC88], [BAR00a], [BAR00c].

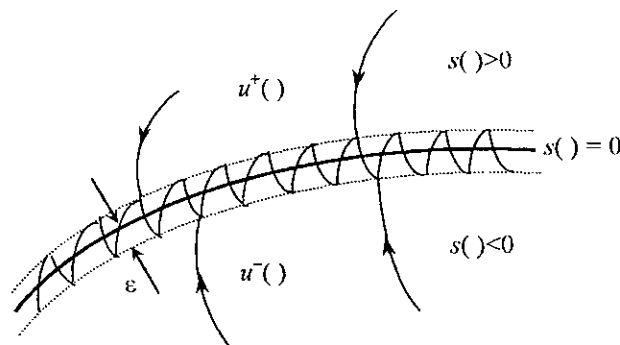


Figure II.2: Mise en évidence de l'écart de poursuite lors de l'implémentation pratique.

Pour remédier au premier inconvénient, par conséquent réduire le temps d'atteignabilité, plusieurs approches ont été développées dans la littérature parmi lesquelles nous pouvons citer :

1. l'utilisation d'un contrôleur à grand gain dans l'équation (II.23) suggérée dans [YOU78], et qui peut malheureusement conduire à la saturation du système et éventuellement accentuer le phénomène de chattering.
2. la condition de convergence (II.19) peut également prendre la forme suivante [SLO91]:

$$[s(x)]' \cdot \dot{s}(x) = \sum_{i=1}^m s_i(x) \cdot \dot{s}_i(x) \leq -\eta \cdot \|s(x)\| \quad (\text{II.24})$$

avec : η une constante positive, qui définit le temps de convergence vers la surface de glissement. La surface s sera atteinte en un temps t_r fini, déterminé par l'inégalité suivante:

$$t_r \leq \frac{s(t=0)}{\eta} \quad (\text{II.25})$$

3. une fonction de commutation variable dans le temps peut être éventuellement utilisée pour éliminer la phase d'atteignabilité où le vecteur d'erreur initial est supposé nul [SLO84]. Cette méthode est utilisée dans le cas d'une commande à structures variables avec un élément intégrateur pour les systèmes du premier ordre [CHU95a], [CHU95b], [MAH99].

II.3.6 Réduction du phénomène de chattering

Dans le présent travail, afin de pallier au problème du chattering, nous avons retenu la solution qui consiste à choisir une commande discontinue qui varie en fonction de la distance entre la variable d'état et la surface de glissement.

Dans ce cas, pour une commande à structures variables du deuxième ordre, la commande discontinue peut être régit par la relation suivante [HAR86]:

$$\Delta u = [\alpha_1 \cdot e_1 + \alpha_2 \cdot e_2 + k] \cdot \text{sgn}(s(x)) \quad (\text{II.26})$$

Sur la base de ces travaux, une expression générale de la commande discontinue pour les systèmes mis sous forme canonique a été présentée donnée par :

$$\Delta u_i = \left[\sum_{j=1}^{r_i} (\alpha_j \cdot |e_j| + k) \right] \cdot \text{sgn}(s_i(x^j)) ; \quad i = 1, \dots, m \quad (\text{II.27})$$

Avec:

α_k : est la pente de la fonction $K(s_i)$ suivant la $j^{\text{ème}}$ direction du plan de la surface s_i de dimension r_i .

k : est la valeur minimale de $K(s_i)$ qui assure la robustesse en mode de glissement, en tenant compte de la plage des incertitudes (Figure II.3).

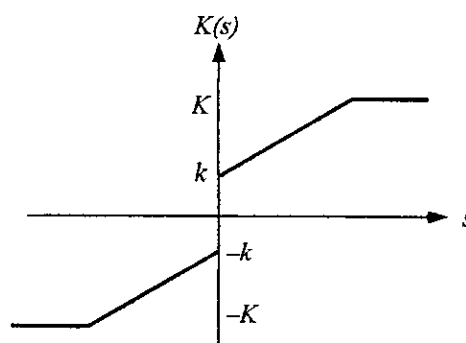


Figure II.3: Réduction du phénomène du chattering par une forme linéaire de la fonction discontinue

Il est clair que, l'augmentation du paramètre k a pour effet une augmentation de la vitesse d'atteignabilité, néanmoins il induit un chattering important se traduisant par un bruit ayant un spectre étendu et une amplitude élevée qui conduit à la présence de parasites indésirables dans les grandeurs de sortie.

L'absence de commutation ($k = 0$), par contre, a l'avantage de réduire ce phénomène au prix de la robustesse de la commande.

D'une manière plus générale, nous proposons une autre formulation dans laquelle la commande discontinue u_n est plus adoucie en étant une fonction non-linéaire de la distance entre l'état du système et la surface de glissement, comme le montre la figure (II.4) [BAR00a], [BAR00c], [BAR01], [BAR02a], [BAR02b].

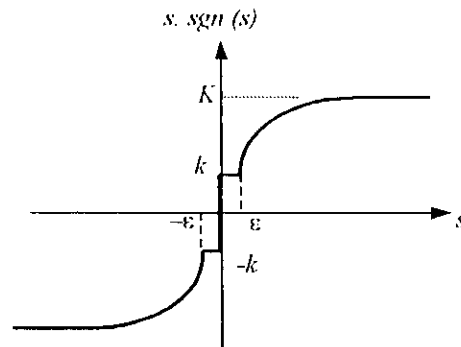


Figure.II.4 : Forme générale de la commande discontinue réduisant le chattering

K : correspond à la valeur maximale de l'organe de commande que peut supporter le système.

ϵ : doit être pris en compte dans l'implémentation pratique du processus car il influe directement sur la fréquence de commutation des semi-conducteurs et sur la poursuite de la trajectoire désirée. Un compromis est fait lors de la conception ; quand ϵ est très petit, la fréquence de commutation est très élevée et l'écart de poursuite autour de la surface est réduit, et vis-versa [SLO91], [UTK92], [UTK93].

Sur cette base, et pour éviter la discontinuité autour de $s = 0$, nous proposons une nouvelle formulation de u_n , définie comme [BAR03a], [BAR03b]:

$$U_n = -M \cdot \text{sgn}(s(x)) \tag{II.28}$$

tel que :

$$M = \begin{cases} K - (K - k) \cdot \exp\left(-\frac{|S(x) - \epsilon|}{\sigma}\right) ; & s(x) > \epsilon \\ \frac{k}{\epsilon} ; & 0 \leq s(x) \leq \epsilon \end{cases}$$

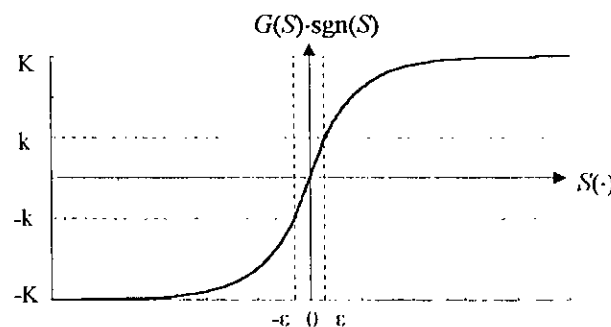


Figure II.5: Forme générale de la commande discontinue réduisant le chattering

K : correspond à la valeur maximale de l'organe de commande que peut supporter le système.

ε : doit être pris en compte dans l'implémentation pratique du processus étant donné qu'il influe directement sur la fréquence de commutation des semi-conducteurs et sur la poursuite de la trajectoire désirée. Lors de la conception, un compromis doit être fait; quand ε est très petit, la fréquence de commutation est très élevée et l'écart de poursuite autour de la surface est réduit, et vice versa [UTK92], [UTK93].

II.4 APPLICATION A LA COMMANDE HYBRIDE DE LA MACHINE ASYNCHRONE

Dans ce paragraphe, nous introduisons le concept de la commande à structures variables (CSV) dans le processus de commande hybride. Les deux régulateurs de vitesse et de flux rotorique de la figure (I.8) sont remplacés par des régulateurs à structures variables. Ce qui impose un choix de surfaces sur chaque axe (d, q) (Fig.II.6) [BAR03a].

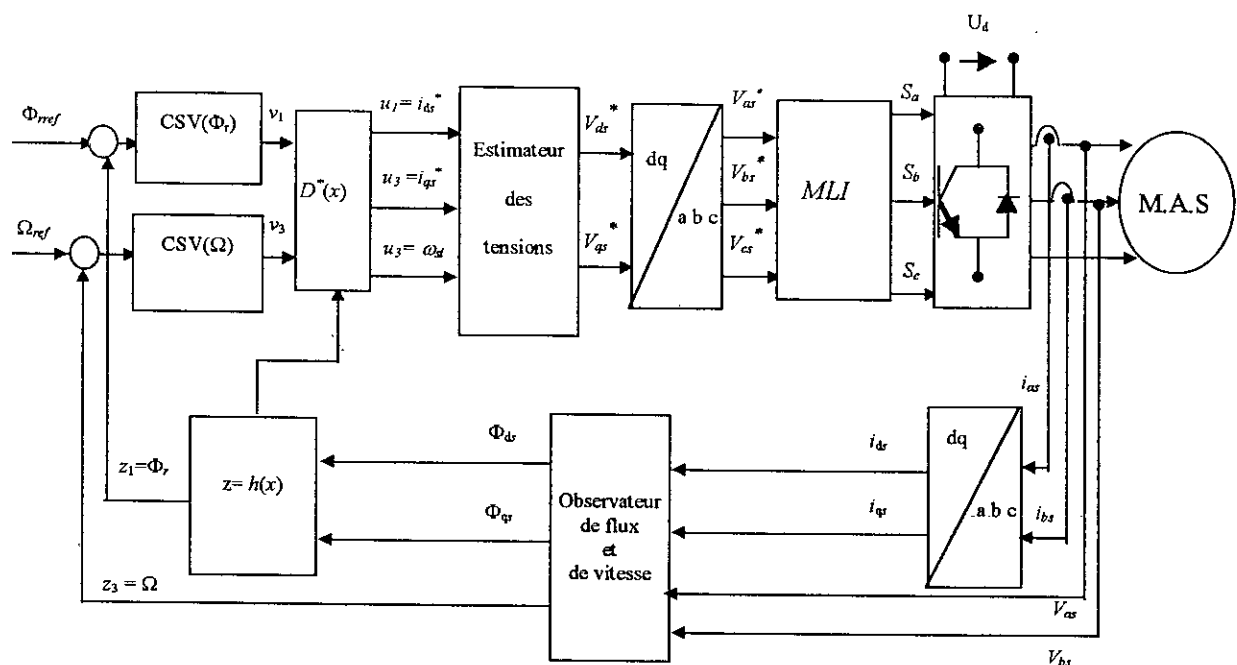


Figure II.6: Schéma du processus de commande par mode de glissement.

Dans le but de prévoir la conception de la commande avec les systèmes intelligents qui sera explicitée ultérieurement, une étude des systèmes à structures variables en continue est effectuée. Par conséquent et dans un premier temps, le système ne tient pas compte de l'ensemble convertisseur statique avec sa commande "MLI - stratégie triangulo sinusoïdale".

L'application de cette stratégie de commande nécessite la détermination du degré relatif de chaque variable à réguler.

II.4.1 Application de la commande u_n de type relais avec un degré relatif $r=1$

Les surfaces choisies pour chacun des axes (d, q) sont données par :

Axes d :

$$s(\Phi_r) = \Phi_{r\text{ref}} - \Phi_r \quad (\text{II.29})$$

Axes q :

$$s(\Omega_r) = \Omega_{r\text{ref}} - \Omega \quad (\text{II.30})$$

Le calcul des lois de commande se fait à partir du système d'équations (I.17), et de la définition de commande discontinue proposée (II.28), comme suit:

- Pour le contrôle du flux :

$$s(\Phi_r) = \Phi_{r\text{ref}} - \Phi_r \quad (\text{II.31})$$

$$\dot{s}(\Phi_r) = 0 \quad \Rightarrow \quad i_{ds\text{eq}} = \frac{T_r \cdot \dot{\Phi}_{r\text{ref}} + \Phi_r}{L_m} \quad (\text{II.31a})$$

$$i_{ds} = i_{ds\text{eq}} + i_{dsn} \quad (\text{II.31.b})$$

Ainsi, nous obtenons:

$$s(\Phi_r) \cdot \dot{s}(\Phi_r) < 0 \quad \Rightarrow \quad \begin{cases} i_{dsn} = K_f \operatorname{sgn}(s(\Phi_r)) \\ i_{ds\text{eq}} = \frac{T_r \cdot \dot{\Phi}_{r\text{ref}} + \Phi_r}{L_m} \end{cases} \quad (\text{II.31.c})$$

- Pour le contrôle de la vitesse :

$$s(\Omega) = \Omega_{r\text{ref}} - \Omega \quad (\text{II.32})$$

$$\dot{s}(\Omega) = 0 \quad \Rightarrow \quad i_{qs\text{eq}} = \frac{J \cdot \dot{\Omega}_{r\text{ref}} + C_r + f \cdot \Omega}{P \left(\frac{L_m}{L_r} \right) \Phi_r}$$

$$i_{qs} = i_{qs\text{eq}} + i_{qsn} \quad (\text{II.32.b})$$

Par conséquent :

$$s(\Omega) \cdot \dot{s}(\Omega) < 0 \quad \Rightarrow \quad \begin{cases} i_{qsn} = K_w \operatorname{sgn}(s(\Omega)) \\ i_{qs\text{eq}} = \frac{J \cdot \dot{\Omega}_{r\text{ref}} + C_r + f \cdot \Omega}{P \left(\frac{L_m}{L_r} \right) \Phi_r} \end{cases} \quad (\text{II.32.c})$$

Pour satisfaire le critère de convergence, les gains (K_f , K_w) doivent toujours être positifs. De plus, leur choix est effectué de manière à imposer la valeur désirée aux sorties des régulateurs respectifs [MAH99].

$$\Rightarrow \begin{cases} K_w = 2 \cdot i_{qs \max} \\ K_f = i_{ds \max} \end{cases}$$

$$\text{Nous avons : } \sqrt{i_{ds \max}^2 + i_{qs \max}^2} \leq i_{s \max}$$

$$\text{tel que : } i_{ds \max} = \frac{\Phi_{r \text{ ref}}}{L_m} \quad \text{et} \quad \sqrt{V_{dr \max}^2 + V_{qr \max}^2} \leq V_{s \max}$$

II.4.2 Application de la commande adoucie proposée

Dans ce paragraphe, nous reprenons les équations obtenues pour le cas avec relais pour les deux régulateurs du processus de commande, en considérant que les commandes u_n sont maintenant régies par l'équation (II.28).

Une simulation numérique a été effectuée dans les mêmes conditions d'essai que ceux de la section (II.1.5) pour le cas idéal avec commande à relais et celle adoucie (Fig. II.7 et II.8) respectivement. Concernant le test de robustesse il a été réalisé uniquement pour le cas de la commande adoucie avec une variation de la résistance rotorique $R_r = 90\% R_{r0}$ (les inductances rotoriques sont connues avec certitude) (Fig.II.9).

Les résultats de la figure (II.7) montrent clairement que comme pour le cas de la commande hybride, l'orientation du flux sur l'axe "d", ainsi que le découplage entre les deux grandeurs "couple-flux" sont réalisés.

Les composantes en quadratures du courant statorique (i_{qs} , i_{ds}) sont proportionnelles au couple électromagnétique C_{em} et au flux rotorique Φ_r , respectivement.

La figure (II.8) montre la nette réduction des oscillations dans les courants (i_{as} , i_{qs}), ainsi que sur le couple relativement au cas avec relais.

Le zoom réalisé sur le courant i_{as} montre clairement l'allure sinusoïdale de ce dernier autour d'une référence dans une petite bande due au phénomène de chattering (Fig.II.7 et II.8).

Pour ce qui est de la tension statorique V_{as} , le zoom effectué montre qu'effectivement l'allure de celle-ci est par modulation de largeur d'impulsion (Fig.II.7 et II.8).

Le test de robustesse met en évidence l'avantage apporté par ce type de commande par rapport à celle par placement de pôles et par flux orienté concernant son insensibilité vis à vis des variations paramétriques et des perturbations externes.

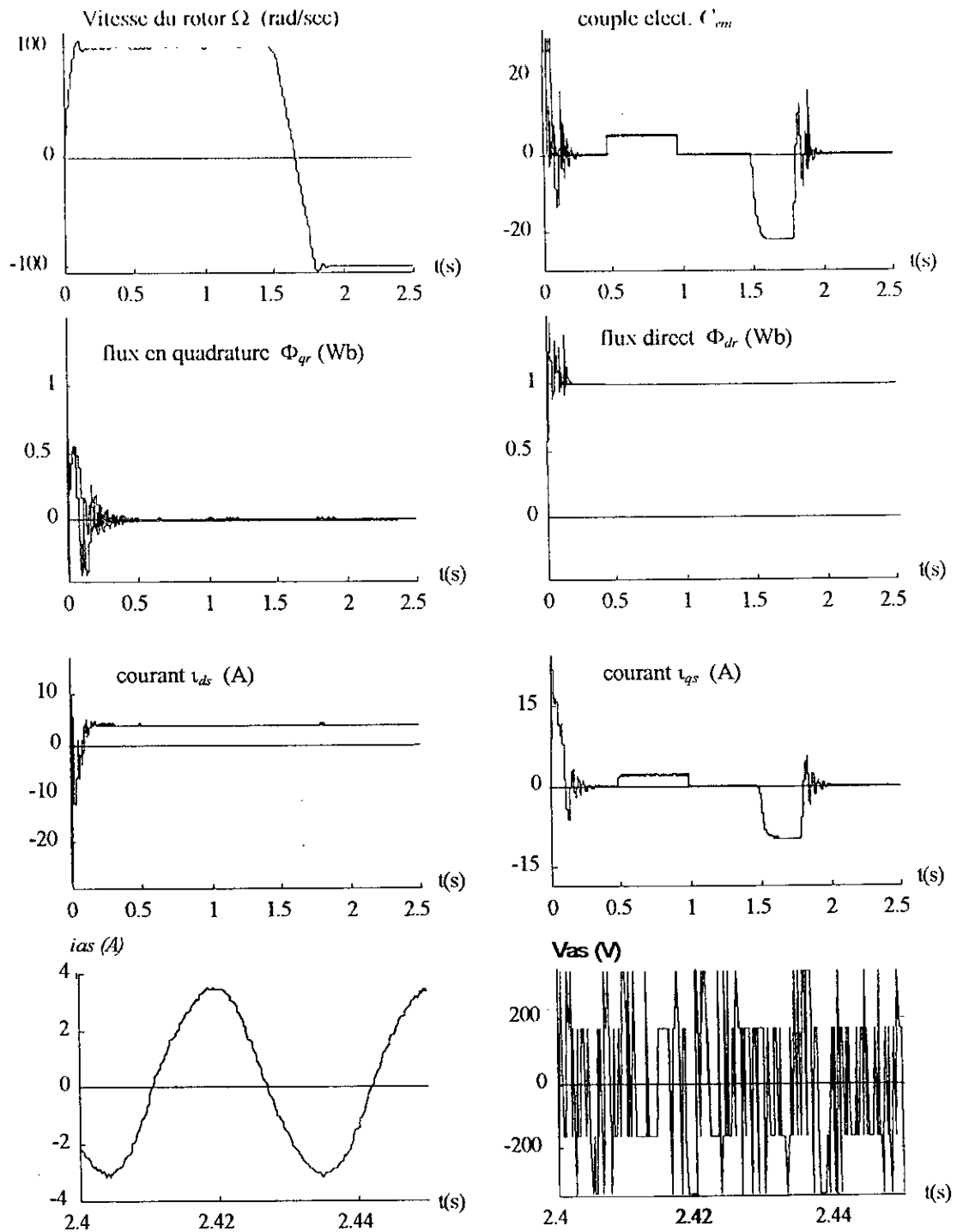


Figure II.7: Résultats de simulation de la commande à structures variables à relais dans le cas idéal.

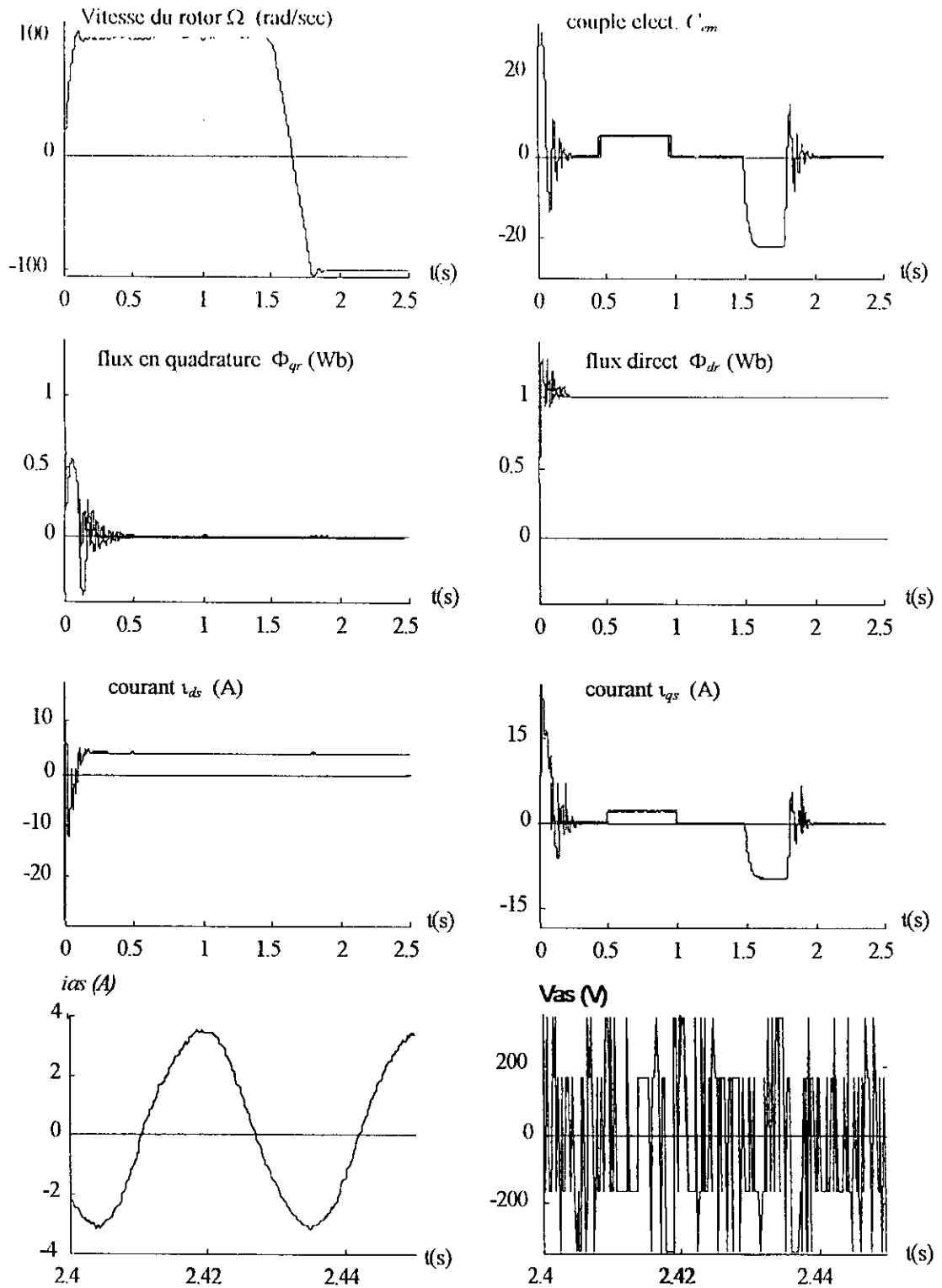


Figure II.8: Résultats de simulation de la commande à structures variables adoucie dans le cas idéal.

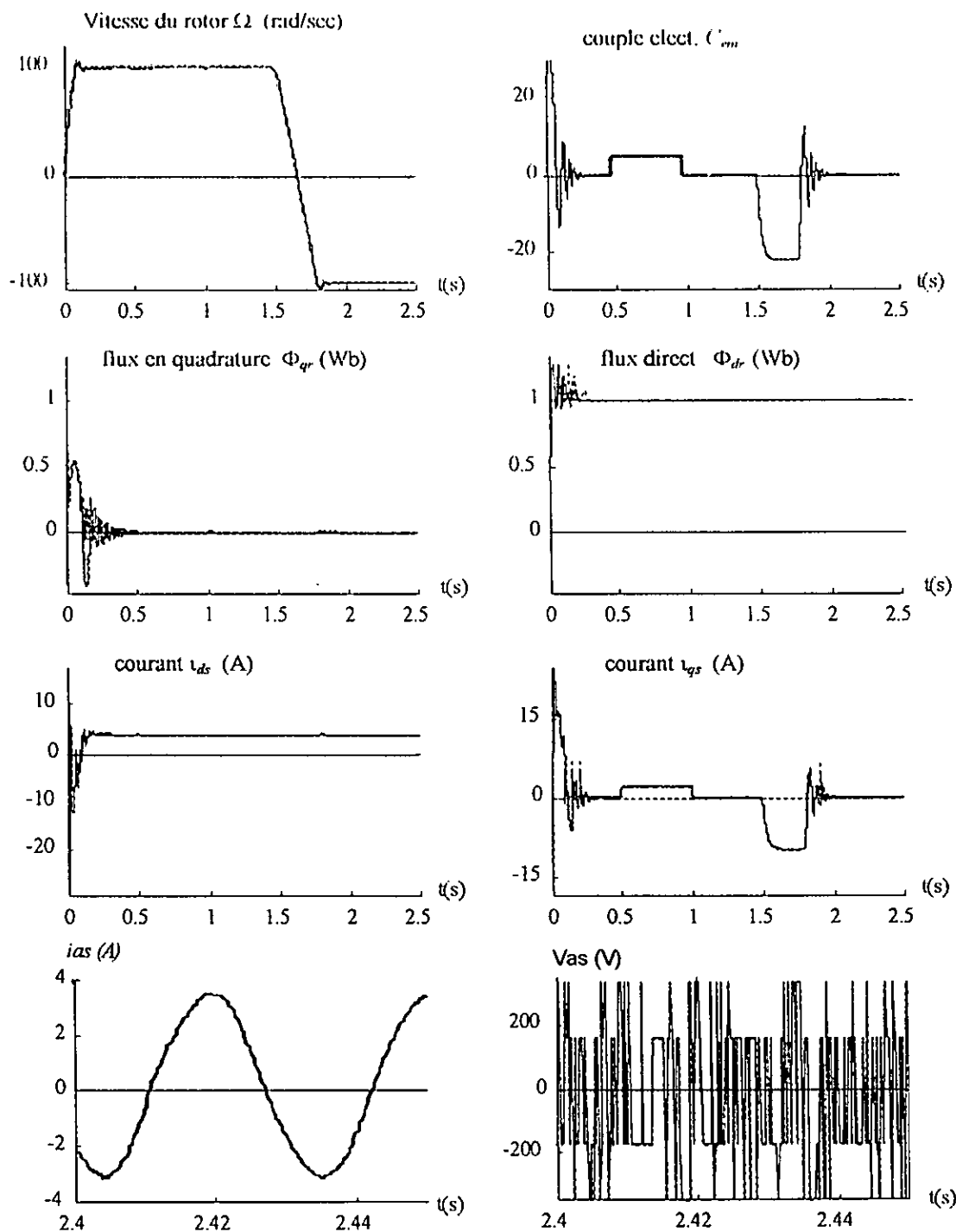


Figure II.9: Commande à structures variables adoucie pour différentes variations de $R_r = 90\% R_{r0}$ (les inductances rotoriques sont connues avec certitude).

L'application de la commande à structures variables à la machine asynchrone a permis de mettre en évidence la robustesse vis à vis des variations paramétriques et perturbations externes et la poursuite de trajectoires désirées obtenues avec ce type de commande quelles que soient les conditions de fonctionnement du système, relativement à celles obtenues avec la régulation par placement de pôle.

II.5 LES INCONVENIENTS ET LES CONTRAINTES LIEES A LA COMMANDE CLASSIQUE

L'étude critique des différents modes de commande présentés précédemment permet de conclure que, d'un côté, ces techniques assurent de bonnes performances et permettent de résoudre plusieurs problèmes liés à la commande de la machine asynchrone, mais d'un autre côté, leur utilisation est accompagnée par plusieurs inconvénients, entre autres nous pouvons citer:

- Pour avoir une commande précise, on doit disposer d'un modèle qui est le plus représentatif possible de la machine asynchrone, ce qui n'est pas toujours évident. En effet, les procédures d'estimation et de mesure engendrent des erreurs systématiques qui ne peuvent pas être totalement compensées,
- Pour obtenir une robustesse face à la variation et à l'incertitude des paramètres, nous devons procéder à une identification en temps réel. Toutefois, les algorithmes utilisés ne permettent pas d'avoir les paramètres avec exactitude. Ainsi, l'écart entre le modèle identifié et le modèle réel persistera toujours et une compensation complète de l'effet de ces variations paramétriques sur les performances de la machine asynchrone n'est jamais atteinte.

Par ailleurs, en plus de la dynamique non-linéaire de la machine asynchrone, si on considère que certaines variables d'état sont mesurées, que d'autres sont estimées ou observées et que certaines d'entre elles sont identifiées, des erreurs peuvent survenir rendant l'analyse du système de commande résultant impossible.

En outre, la mise en œuvre de ces techniques de commande est plus ou moins difficile à réaliser, voire impossible. En effet, Les résultats de simulation numérique obtenus avec la loi de commande adoucie proposée permettent de voir que le chattering est réduit, mais se fait au prix d'une difficulté dans la mise en œuvre en particulier celle de u_n qui est de forme exponentielle.

En effet, si le recours à des microprocesseurs puissants permet d'implémenter des lois de commande complexes, leur utilisation se répercute sur les frais d'installation du système global et l'avantage lié au faible coût de la machine asynchrone n'est plus de mise.

De plus, pour avoir de bonnes performances, une connaissance précise du flux est exigée, d'où la contrainte de la nécessité d'avoir recours à un observateur de flux fiable.

II.6 CONCLUSION

A la lumière de l'analyse développée concernant les diverses solutions envisagées pour résoudre les problèmes liés à la commande de la machine asynchrone, nous pouvons conclure que chaque type de commande peut être avantageux dans un sens et désavantageux dans un autre. A cet effet, de nos jours, la recherche de solutions adéquates pour la commande des systèmes non-linéaires d'une manière générale s'oriente de plus en plus vers les commandes par les systèmes intelligents entre autres les réseaux de neurones artificiels permettant d'élaborer une commande neuronale non-linéaire susceptible de surmonter certains problèmes lié à la commande de la machine asynchrone entre autres:

- Permettre une identification précise des termes non-linéaires utiles pour générer l'action de commande par retour d'état linéarisant,
- la compensation des erreurs dues à l'estimation et aux mesures,
- le découplage exact entre la commande de la vitesse et du flux rotorique,
- l'établissement d'une commande précise, robuste et simple à implémenter qui assure de meilleures performances lors de la poursuite et de la régulation dans les différents régimes de fonctionnement de la machine asynchrone.

Dans ce sens, le chapitre suivant est consacré au préalable à l'analyse des principes de fonctionnement, d'apprentissage et de commande par réseaux de neurones artificiels afin de nous faciliter la sélection du type de réseaux le plus approprié et son introduction dans l'application envisagée.

Chapitre III

Commande Hybride Par Réseaux de Neurones

Artificiels de la Machine Asynchrone

Chapitre III

Commande Hybride Par Réseaux de Neurones Artificiels de la Machine Asynchrone

III.1 HISTORIQUE

Le développement, dans les années 40, des réseaux de neurones artificiels (RNA) ou réseaux de neurones formels résultent des travaux de plusieurs chercheurs visant à reproduire un comportement intelligent à partir d'un modèle neuronal simplifié.

Pour cela, au travers de différents modèles, ces chercheurs ont essayé de développer des outils capables d'apprendre, de reconnaître et surtout d'acquérir la capacité d'effectuer une tâche spécifique par apprentissage. Malheureusement, les résultats atteints sont loin d'égaliser les capacités d'apprentissage développées par le cerveau humain, qui peut s'expliquer par la relative simplicité du modèle de réseau employé comparée à la complexité du cerveau. À partir des ressemblances biologiques, les réseaux de neurones permettent de réaliser des opérations dites intelligentes.

Dans ce sens, l'intérêt des neurologues et des biologistes pour les réseaux de neurones artificiels s'est orienté vers la compréhension du cerveau humain à partir de l'élaboration et de l'étude de modèles artificiels complexes et plausibles au sens biologique, qui tentent de reproduire certaines caractéristiques du cerveau.

En revanche, les ingénieurs et les informaticiens privilégient l'aspect calcul avec la construction de réseaux de neurones simplifiés, arborant une puissance de calcul élevée. Selon cette approche, les modèles neuronaux sont usuellement réduits à l'essentiel, l'efficacité primant sur la plausibilité biologique.

Arrivés sur le marché industriel au milieu des années 80, les réseaux de neurones artificiels ont largement bénéficié des capacités de calcul offertes par les ordinateurs modernes. Des environnements de développement sont apparus, permettant aux ingénieurs de développer des applications grandeur réelle.

Après l'enthousiasme de la fin des années 80, les chercheurs et ingénieurs ont été confrontés à des problèmes que les statisticiens connaissent depuis toujours et qui résident dans la taille des bases d'exemples, la représentativité des données, la signification et l'interprétation des résultats et des estimations..., etc. Ces problèmes soulèvent l'insuffisance des bases théoriques des pratiques heuristiques qui longtemps ont tenu lieu de méthodes pour le développement des techniques neuronales [BRE99].

Les liens qui unissent la statistique et les réseaux de neurones sont forts, et l'intersection des deux disciplines importante. Le processus d'apprentissage d'un réseau est en effet un processus stochastique (aléatoire) qui dépend des propriétés statistiques de la distribution des exemples de la base d'apprentissage. Des connotations statistiques apparaissent également dans la formulation des réseaux de neurones, dans l'étude de leur capacité de généralisation, ainsi que dans l'évaluation de leurs performances et du processus de validation des résultats [HER91], [HER94], [DAV93], [BRE99].

Dans notre travail, après avoir explicité la nature, les principales architectures et les mécanismes d'apprentissage des réseaux de neurones artificiels, ce chapitre leur associe une formalisation fondée sur une théorie de l'apprentissage afin d'acquérir la loi de commande des régulateurs à structures variables de la structure de réglage hybride de la machine asynchrone présentée dans le chapitre précédent et pour pouvoir les supplanter une fois l'apprentissage réalisé.

III.2 DU NEURONE BIOLOGIQUE AU NEURONE ARTIFICIEL

Un réseau de neurones artificiels est un processeur parallèle de traitement d'informations distribuées, qui présente une propension naturelle à la mémorisation et à l'exploitation de connaissances relatives à l'environnement dans lequel il est immergé, connaissances acquises à partir de l'expérience.

Sa structure repose sur une interconnexion massive de cellules élémentaires de traitement d'information, appelées neurones formels ou modèle de *Pits* et *MC-Culloch*, dont la représentation est un graphe dirigé [BAR86], [HEC90], [DAV93], [BRE99].

L'analogie entre un réseau de neurones formels et le cerveau humain se résume par [DAV93], [HAY94]:

1. un réseau de neurones artificiels acquiert la connaissance de son environnement par l'intermédiaire d'un apprentissage qui "simule" la plasticité du cerveau. A cet effet, en cours d'apprentissage, le réseau peut être amené à modifier sa structure en créant ou supprimant des neurones ou des liaisons entre neurones.
2. la connaissance acquise par un réseau de neurones est encodée par les forces ou intensités évolutives des connexions établies entre neurones formels. Les forces de ces connexions reliant les neurones se réfèrent aux valeurs des poids ou coefficients synaptiques qui définissent les paramètres du modèle interne du réseau.

Des propos ainsi soulevés, il apparaît que l'intérêt des réseaux de neurones artificiels réside dans le parallélisme de leur structure, leur capacité d'adaptation, ainsi que leur mémoire distribuée.

Il est important de citer également la capacité de généralisation des réseaux de neurones qui pourrait éventuellement émerger de la phase d'apprentissage. Cette capacité désigne l'aptitude d'un réseau à présenter un comportement acceptable en réponse à des stimuli externes de son environnement non rencontrés lors de l'apprentissage (interpolation et extrapolation).

A la lumière des propriétés citées, il est possible de déterminer les caractéristiques des problèmes qui pourraient nécessiter une résolution par les réseaux de neurones formels [DAV93].

3. le modèle du problème considéré est inconnu ou difficile à formaliser. Un ensemble d'exemples, constitué d'entrées du problème auxquelles sont associées des solutions fournies par un expert, est néanmoins accessible,
4. les données du problème sont entachées de bruit,
5. le problème est de nature évolutive,
6. le problème nécessite un traitement en temps réel.

Les domaines d'application privilégiés, présentant les caractéristiques d'une résolution neuronale exposées ci-dessus, concernent le regroupement et la classification de données, le traitement du signal, la modélisation et l'identification de processus, le contrôle (surveillance) et la commande de processus, ...etc [TAY83], [BAR86], [RUM86], [PSA87], [NAR90], [DAV93], [JOD94], [BAR99], [BAR00b].

III.4 RESEAUX DE NEURONES ARTIFICIELS

De nos jours, une manière simple de concevoir un réseau de neurones consiste à considérer qu'il s'agit d'un système de traitement de l'information composé d'un grand nombre de processeurs interconnectés appelées cellules. Chacune de ces cellule calcule sa sortie sur la base d'informations qui lui sont transmises à partir des autres cellules qui sont connectées à elle et des poids de ces connexions. Un réseau de neurones peut-être complètement décrit par la spécification des quatre éléments suivants :

1. l'élément de traitement,
2. l'architecture ou topologie du réseau qui correspond au schéma d'interconnexions,
3. le type d'apprentissage
4. le schéma d'activation du système.

III.4.1 L'élément de traitement

Il est décrit par une fonction de transfert, appelée également fonction d'activation, qui établie une correspondance entre les valeurs présentées à l'entrée de la cellule et la valeur de sortie [DAV93].

III.4.2 L'architecture d'un réseau

Elle est entièrement déterminée par le modèle de ses neurones et la topologie de leurs connexions.

Trois classes d'architectures de réseaux peuvent être considérées à savoir : les réseaux mono- et multicouches [HAY94], les réseaux récurrents [HAY94] et les réseaux topologiques [DAV93], [BRE99].

III.4.1.1 Réseaux statiques ou multicouches

Un réseau statique présente une architecture de neurones organisée en couches successives. L'interconnexion des neurones est réalisée entre couches adjacentes, suivant un sens préférentiel de

En raison des retards ou délais introduits par les boucles de réaction, un réseau récurrent présente un comportement dynamique qui lui permet d'intégrer explicitement la dimension temporelle. En effet, l'évolution de ce type de réseaux dépend intégralement du temps, ou en d'autre terme, l'état présent du réseau dépend de ces états passés [BAR86], [DAV93].

L'architecture d'un tel réseau de neurones est illustrée à la figure (III.2).

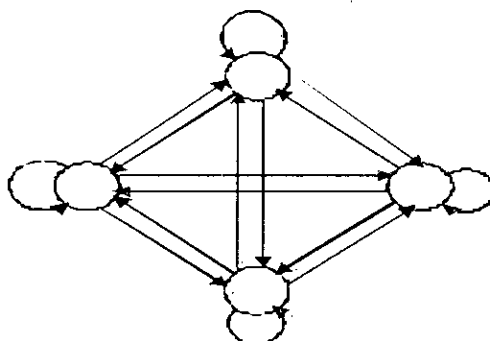


Figure.III.2: Architecture d'un réseau récurrent mono-couche à interconnexion totale.

III.4.2.3 Réseaux topologiques

Un réseau topologique consiste en une couche mono- ou multidimensionnelle de neurones structurés en topologie, terme qui signifie une organisation spatiale des neurones dans la couche qui détermine une relation de voisinage entre neurones.

La notion de voisinage est définie par une mesure de proximité ou de distance qui obéit à une métrique particulière (distance euclidienne [DAV93], ou distance de Hamming généralisée [BRE99]). Elle n'est pas explicitement représentée dans le réseau et il n'existe pas de connexions entre les neurones de la couche topologique. En effet, elle n'est apparente que dans le comportement d'apprentissage des neurones du réseau qui signifie que des neurones voisins de la couche topologique mémorisent des connaissances similaires mais nuancées.

L'apprentissage des réseaux topologiques aboutit ainsi à une organisation distribuée, structurée de la connaissance [BRE99].

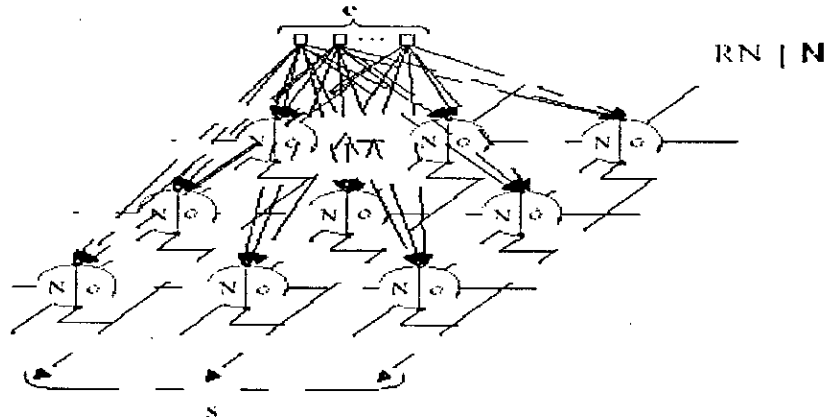


Figure.III.3: Architecture d'un réseau topologique bidimensionnel à métrique euclidienne.

III.4.3 Apprentissage et adaptation

La caractérisation d'un système par le terme général "réseau de neurones" implique souvent son aptitude à acquérir des connaissances. L'apprentissage est le processus par lequel le réseau de neurones acquiert la capacité d'effectuer certaines tâches et ce, en ajustant ses paramètres internes (poids synaptiques) selon un schéma d'apprentissage spécifique. En fin d'apprentissage, le réseau est susceptible de nous fournir des sorties aussi proches que possible des sorties désirées.

Par ailleurs, si le terme adaptation, au sens de l'ajustement des paramètres internes d'un réseau de neurones artificiels réfère à l'assimilation et la mémorisation de connaissances, il dénote également la capacité du réseau à modifier dynamiquement son comportement afin de répondre à de nouvelles attentes ou de nouvelles situations.

La nuance de sens accordée au terme adaptation dans le domaine des réseaux de neurones est liée à la prise en considération ou non du facteur temps dans le processus d'apprentissage. Cet apprentissage, qu'il soit supervisé ou non supervisé, est en effet conventionnellement réalisé soit en ligne (temps réel), soit hors ligne (temps différé).

L'apprentissage hors ligne convient aux réseaux de neurones qui opèrent en environnement stationnaire. En raison de l'invariance d'un tel environnement au cours du temps, les paramètres du réseau peuvent être déterminés par un apprentissage limité à un ensemble de stimuli fini et représentatif.

Une fois l'apprentissage terminé, l'identification du modèle de connaissance stationnaire de l'environnement justifie le gel des paramètres du réseau en phase de restitution de la connaissance. Le réseau présente alors un comportement statique [TAY83], [JOD94].

Cependant, l'environnement dans lequel est immergé le réseau est souvent non stationnaire. En ce cas, l'état et le comportement de l'environnement sont sujets à des transformations au cours du temps. D'autre part et par nature, l'apprentissage hors ligne ne permet pas de réactualiser le modèle de connaissance d'un environnement évolutif. Par conséquent, afin de surpasser cette limitation, l'apprentissage doit permettre l'ajustement en continu des paramètres du réseau. Ce qui est réalisé en

amenant le réseau à considérer chaque stimulus produit par l'environnement comme un nouveau stimulus à assimiler. La dépendance temporelle de la connaissance est ainsi implicitement intégrée dans le modèle du réseau par le biais d'un apprentissage en ligne exploitant une base d'exemples ordonnés dans le temps. Le réseau présente alors un comportement adaptatif, au sens large du terme, lié à la nature spatio-temporelle implicite de l'apprentissage en ligne [DAV93], [BRE99].

III.4.3.1 Apprentissage supervisé

Dans ce type d'apprentissage, les poids des connexions sont déterminés par le biais d'algorithmes qui consistent à minimiser l'erreur quadratique entre la sortie désirée et la sortie du réseau jusqu'à l'obtention d'une performance acceptable.

III.4.3.2 Apprentissage non supervisé

La détermination des poids, dans ce cas, n'est pas en fonction des erreurs, mais en présentant au réseau une quantité suffisante d'exemples contenant des corrélations de telle sorte que celui-ci en dégage les régularités automatiquement. Ces réseaux sont souvent appelés "auto-organiseurs" (self-organizing neural networks) ou encore à apprentissage compétitif.

III.4.3.3 Différents algorithmes d'apprentissage des Réseaux de Neurons Artificiels

Il existe plusieurs types d'algorithmes d'apprentissage, parmi lesquels nous pouvons citer : la Rétro-propagation, Fast-Retropropagation, Recurrent-Retropropagation, ROM (Random-Optimisation-Method),.....etc.

Dans le présent travail, nous mettons le point sur l'algorithme de la rétro-propagation avec momentum qui a été utilisé pour l'entraînement des deux réseaux de neurones destinés à remplacer les régulateurs à structures variables dans le processus de commande hybride de la machine asynchrone de la figure (II.6), [TAY83], [HEC90], [HER91], [DAV93], [JOD94].

➤ Algorithme de la Rétro-propagation

La backpropagation ou rétropropagation de l'erreur est une méthode d'apprentissage basée sur la minimisation d'un critère quadratique de l'erreur réalisée en adaptant continuellement les paramètres du réseau dans la direction de la plus grande descente de l'erreur. Le changement de chaque paramètre est proportionnel à sa relative contribution dans la somme des carrés des erreurs.

L'apprentissage par cette méthode s'effectue en cinq étapes successives :

– Première étape : Présentation du patron d'exemples

Un signal d'entrée est appliqué à l'entrée du réseau et se propage à travers les couches supérieures ou caches cachées dans le but de générer un signal de sortie.

– Deuxième étape : Calcul de l'erreur quadratique

La sortie du réseau obtenue est comparée à la sortie désirée pour générer un signal d'erreur. Ce dernier va se propager en sens inverse (d'où le nom rétro-propagation) en partant des sorties, provoquant au fur et à mesure une modification des poids dans le sens de la minimisation de l'erreur.

Pour des raisons de simplicité, nous présentons cet algorithme pour un réseau de trois couches. La généralisation à un réseau de plus de trois couches peut être faite par simple analogie.

Considérons le réseau à trois couches de la figure (III.4).

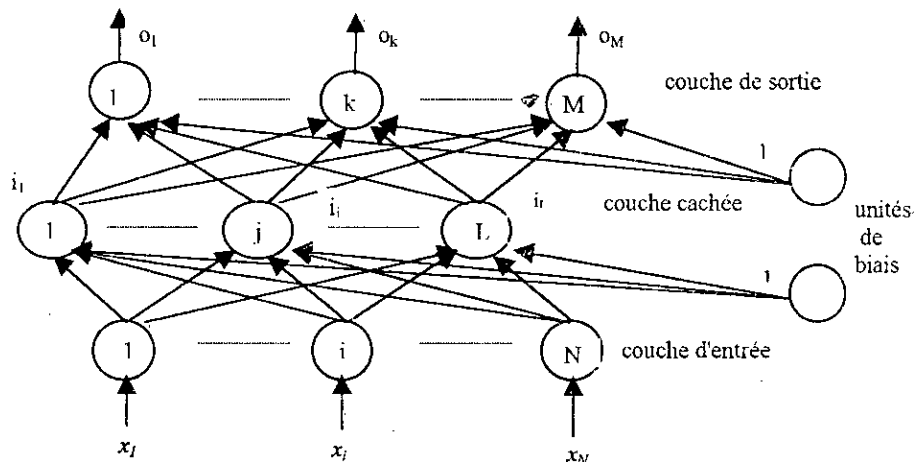


Figure III.4: Structure d'un réseau à 3 couches.

Les notations suivantes sont adoptées:

q : nombre d'exemple à présenter au réseau lors de l'apprentissage. Nous désignons par le mot "exemple" la paire (X_p, Y_p) où :

$X_p = (x_{p1} \ x_{p2} \ \dots \ x_{pN})^T$: vecteur d'entrée avec N le nombre d'entrées du réseau.

$Y_p = (y_{p1} \ y_{p2} \ \dots \ y_{pM})^T$: vecteur des sorties désirées avec M le nombre de sorties du réseau.

$I_p = (i_{p1} \ i_{p2} \ \dots \ i_{pL})$: vecteur de sortie de la couche cachée avec L le nombre de neurones de la couche cachée.

$O_p = (o_{p1} \ o_{p2} \ \dots \ o_{pM})$: vecteur de sortie du réseau en réponse à X_p .

w_{ji}^1 : poids de la connexion entre le $i^{\text{ième}}$ neurone de la couche d'entrée et le $j^{\text{ième}}$ neurone de la couche cachée.

w_{kj}^2 : poids de la connexion entre le $j^{\text{ième}}$ neurone de la couche cachée et le $k^{\text{ième}}$ neurone de la couche de sortie.

net_{pj}^1 : somme pondérée du $j^{\text{ème}}$ neurone de la couche cachée, pour l'exemple p .

net_{pk}^2 : somme pondérée du $k^{\text{ème}}$ neurone de la couche de sortie, pour l'exemple p .

f : fonction d'activation des neurones.

Nous pouvons alors écrire les équations suivantes :

$$\begin{cases} \text{net}_{pj}^1 = \sum_{i=1}^{N+1} w_{ji}^1 \cdot x_{pi} \\ i_{pj} = f(\text{net}_{pj}^1) \end{cases} \quad j = \overline{1, L} \quad (\text{III.1})$$

$$\begin{cases} \text{net}_{pk}^2 = \sum_{j=1}^{L+1} w_{kj}^2 \cdot i_{pj} \\ o_{pk} = f(\text{net}_{pk}^2) \end{cases} \quad k = \overline{1, M} \quad (\text{III.2})$$

Le réseau est entraîné pour minimiser l'erreur totale E comme suit:

$$E = \frac{1}{2} \sum_{p=1}^q \sum_{k=1}^M (y_{pk} - o_{pk})^2 \quad (\text{III.3})$$

Généralement, au lieu de considérer l'erreur sur tous les exemples, seule l'erreur E_p sur l'exemple p est prise en compte donnée par :

$$E_p = \frac{1}{2} \sum_{k=1}^M (y_{pk} - o_{pk})^2 \quad (\text{III.4})$$

Ce qui facilite la dérivation de l'algorithme et conduit au même résultat. Pour déterminer le changement des poids, nous calculons le gradient de E_p par rapport à w_{kj}^2 dans le but d'ajuster les poids de la couche de sortie, puis par rapport à w_{ji}^1 pour celui des poids de la couche cachée. Enfin, pour minimiser l'erreur, nous faisons évoluer les poids dans le sens inverse du gradient.

– Troisième étape : Ajustement des poids de la couche de sortie :

$$\begin{aligned} \frac{\partial E_p}{\partial w_{kj}^2} &= - (y_{pk} - o_{pk}) \frac{\partial f}{\partial (\text{net}_{pk}^2)} \frac{\partial (\text{net}_{pk}^2)}{\partial w_{kj}^2} \\ &= - (y_{pk} - o_{pk}) f'(\text{net}_{pk}^2) i_{pj} \end{aligned} \quad (\text{III.5})$$

et

$$w_{kj}^2(t+1) = w_{kj}^2(t) - \eta \frac{\partial E_p}{\partial w_{kj}^2} \quad (\text{III.6})$$

où : η est un paramètre d'apprentissage ajouté pour des considérations pratiques.

Ainsi, nous obtenons :

$$w_{kj}^2(t+1) = w_{kj}^2(t) + \eta (y_{pk} - o_{pk}) f'(\text{net}_{pk}^2) i_{pj} \quad (\text{III.7})$$

– Quatrième étape : Ajustement des poids de la couche cachée :

L'erreur E_p peut être écrite comme :

$$\begin{aligned} E_p &= \frac{1}{2} \sum_{k=1}^M (y_{pk} - o_{pk})^2 \\ &= \frac{1}{2} \sum_{k=1}^M (y_{pk} - f(\text{net}_{pk}^2))^2 \\ &= \frac{1}{2} \sum_{k=1}^M \left(y_{pk} - f \left(\sum_{j=1}^{L+1} w_{kj}^2 i_{pj} \right) \right)^2 \end{aligned} \quad (\text{III.8})$$

Ce qui nous permet d'aboutir à :

$$\begin{aligned} \frac{\partial E_p}{\partial w_{ji}^1} &= - \sum_{k=1}^M (y_{pk} - o_{pk}) \frac{\partial o_{pk}}{\partial (\text{net}_{pk}^2)} \frac{\partial (\text{net}_{pk}^2)}{\partial i_{pj}} \frac{\partial i_{pj}}{\partial (\text{net}_{pj}^1)} \frac{\partial (\text{net}_{pj}^1)}{\partial w_{ji}^1} \\ &= - f'(\text{net}_{pj}^1) x_{pi} \sum_{k=1}^M (y_{pk} - o_{pk}) f'(\text{net}_{pk}^2) w_{kj}^2 \end{aligned} \quad (\text{III.9})$$

d'où :

$$w_{ji}^1(t+1) = w_{ji}^1(t) + \eta f'(\text{net}_{pj}^1) x_{pi} \sum_{k=1}^M (y_{pk} - o_{pk}) f'(\text{net}_{pk}^2) w_{kj}^2 \quad (\text{III.10})$$

Il existe plusieurs versions améliorées de la rétro-propagation qui ont été proposées en vue de surmonter les inconvénients de cette dernière. Parmi celles-ci nous pouvons citer l'algorithme de la rétro-propagation avec momentum.

– Cinquième étape : Backpropagation avec momentum

Dans cette version, lors du calcul du changement $\Delta w(t)$ des poids, une fraction du changement précédent $\Delta w(t-1)$ est prise en compte. Ce terme additionnel tend à maintenir une évolution des poids dans la même direction résolvant ainsi le problème des minimums locaux. Elle est régie par :

$$w_{ji}^1(t+1) = w_{ji}^1(t) + \eta f'(\text{net}_{pj}^1) x_{pi} \sum_{k=1}^M (y_{pk} - o_{pk}) f'(\text{net}_{pk}^2) w_{kj}^2 + \Delta_{ij} w_{ji}^1(t-1) \quad (\text{III.11})$$

III.5 CERTAINES CONSIDERATIONS PRATIQUES POUR LE CHOIX DU RESEAU DE NEURONES ADEQUAT

Avant de procéder à l'apprentissage d'un réseau de neurones artificiels, un certain nombre de paramètres doivent être fixés (dimensionnement du réseau, pas d'apprentissage..., etc), au risque d'engendrer et de compromettre le processus d'entraînement dans le cas d'un mauvais choix.

1. Choix et dimensionnement d'un réseau

D'une manière générale, il n'existe pas de méthodes systématiques pour la détermination du nombre de couches dans le réseau ou le nombre de neurones par couche. Néanmoins, il existe dans ce sens des règles pratiques tirées à partir de l'expérience, parmi lesquelles nous pouvons citer:

Règle 1 : le nombre de neurones par couche doit être d'autant plus élevé que la relation entre les données en entrée et les sorties désirées est complexe.

Règle 2 : pour les réseaux à une seule couche cachée, le nombre de neurones dans cette couche doit être égal à :

$$h = \frac{c}{10(M + N)} \quad (\text{III.12})$$

avec :

c : le nombre de vecteurs dans le fichier d'exemples.

M : nombre de neurones dans la couche de sortie.

N : nombre d'entrées du réseau.

Règle 3 : le nombre de couches cachées doit être égal au nombre de relations qui existent entre les neurones de la couche d'entrée et ceux de la couche de sortie.

2. Paramètre d'apprentissage ou pas de correction η

Le choix du paramètre d'apprentissage η a un effet important sur les performances du réseau. Généralement, il prend des valeurs petites de l'ordre de 0.05 à 0.25, afin d'assurer une certaine convergence. Ce paramètre présente en réalité un dilemme : s'il est trop grand, il peut entraîner des oscillations, s'il est trop petit, le temps d'apprentissage devient infini. Par conséquent, ce le paramètre η est choisi variable à partir d'une valeur grande qu'on diminue progressivement.

3. Poids initiaux

Les poids initiaux sont choisis aléatoirement entre -0.5 et 0.5. Par ailleurs, il est recommandé de ne pas initialiser les poids synaptiques à une même valeur.

4. Minimums locaux

L'algorithme d'apprentissage peut éventuellement, converger vers un minimum local du critère d'erreur. Dans ce cas, l'erreur stagnera autour d'une valeur relativement grande, et les performances requises ne seront jamais atteintes. Pour remédier à ce problème, il existe plusieurs alternatives à savoir : changer le pas de correction, changer les poids initiaux, utiliser rétro-propagation avec momentum, changer la fonction d'activation, augmenter le nombre de couches cachées ou bien augmenter le nombre de neurones par couche.

5. Saturation du réseau

Etant donné que, les fonctions d'activation dans un réseau de neurones sont généralement bornées, et fournissent des sorties entre 0 et 1 ou entre -1 et 1, il est de ce fait nécessaire de mettre à l'échelle les données présentées en entrée.

III.6 LES RESEAUX DE NEURONES ET LA COMMANDE DES PROCESSUS

Widrow fut l'un des premiers chercheurs à avoir utiliser ce type de commande à base de réseaux de neurones artificiels à un pendule inversé en 1960. Il s'est référé à cette technique de commande comme à une méthode de construction d'un système expert neuronal par l'acquisition du savoir-faire d'un expert existant. Guez et Selinsky ont fourni une version plus améliorée et plus sophistiquée de la commande supervisée en mettant en évidence la capacité d'un réseau de neurones à apprendre à commander un système dynamique à partir d'exemples générés par des lois de commande linéaire, non linéaire et par un opérateur humain. Jorgenson a fourni lui aussi un excellent exemple de la commande supervisée dans le domaine de l'aéronautique.

Ainsi, les facultés dont jouissent les réseaux de neurones, ont fait de ceux-ci un outil très intéressant et recherché pour la commande des systèmes, en particulier les systèmes non linéaires.

Il existe une multitude de commande par réseaux de neurones dont la plus fréquemment utilisée est la commande supervisée basée sur la supervision d'un régulateur déjà existant. Cette technique a été adoptée dans le processus de commande hybride. Pour son implémentation, des régulateurs à structures variables sont synthétisés au préalable, puis supervisés par des réseaux de neurones appropriés.

Ce type de commande s'effectue en deux phases: une phase d'apprentissage et une autre de commande comme le montre la figure (III.5).

- **Phase d'apprentissage :** lors de cette phase, le réseau de neurones apprend la fonction entrées/sorties du régulateur existant. En effet, un fichier d'exemples est généré en laissant ce régulateur neuronal commander le système en utilisant un algorithme d'apprentissage supervisé.
- **Phase de commande :** une fois l'apprentissage effectué, le régulateur est remplacé par le réseau de neurones entraîné afin de commander le système.

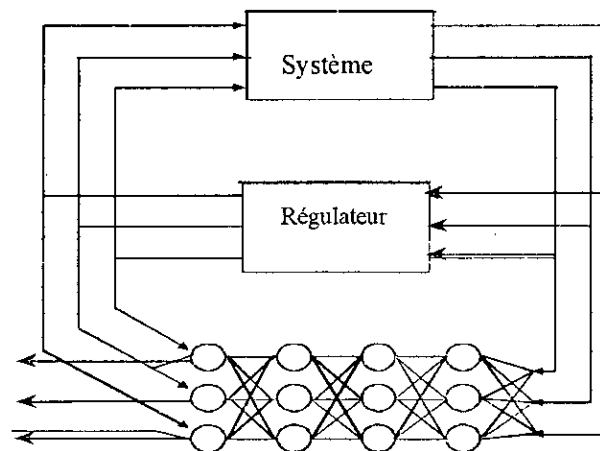


Figure III.5: Commande supervisée

III.6.1 Utilité de la commande supervisée :

La question concernant l'utilité de cette méthode de commande peut être posée. En effet, puisqu'un régulateur effectif existe déjà, et sait accomplir sa tâche, quel est l'intérêt d'avoir une copie de ce régulateur sous forme d'un réseau de neurones?

En réalité, il existe plusieurs réponses à cette question à savoir [BRE99]:

- Le régulateur existant peut être un élément impraticable lors de l'utilisation (comme un opérateur humain). Guez et Selinsky ont obtenu d'excellents résultats en supervisant une loi de commande de type "opérateur humain".

- Comme les calculs (transmission d'informations) dans un réseau de neurones se font en parallèle, la prise de décision et la réaction du réseau aux changements se fait très rapidement, contrairement à un opérateur humain ou un programme de commande implanté dans un calculateur séquentiel.
- Il n'est pas nécessaire que la loi de commande à superviser soit donnée sous forme explicite étant donné que l'apprentissage des réseaux de neurones se fait à partir d'exemples. À priori, nous ne pouvons expliciter sous forme d'équations mathématiques une loi de commande présentée sous forme d'exemples.
- Le caractère distribué de la représentation des réseaux de neurones fournit une meilleure robustesse par rapport aux perturbations (internes et externes) et prévient la dégradation de l'information.
- Robustesse naturelle relativement aux paramètres non modélisés, due au pouvoir de généralisation des réseaux de neurones.
- Le réseau de neurones peut former une loi de commande sur la base d'une représentation d'état du système qui est plus facile à mesurer que la représentation requise par le régulateur existant.

III.7 COMMANDE SUPERVISEE PAR RESEAUX DE NEURONES DE LA COMMANDE HYBRIDE

Dans le présent travail, pour concevoir la commande hybride par réseaux de neurones nous procédons de la manière suivante :

1. adopter au préalable l'architecture adéquate des réseaux de neurones à utiliser,
2. apprentissage de ces réseaux de neurones par l'algorithme de rétropropagation avec momentum,
3. concevoir une commande supervisée pour le processus de commande hybride à structure variable.

III.7.1 Architecture des réseaux de neurones artificiels adoptée

Les deux réseaux de neurones destinés à remplacer les régulateurs de vitesse et de flux à structures variables du processus de commande de la figure (II.6) sont choisis comme étant des réseaux statiques, multi-couches, ayant trois couches respectives comme illustré à la figure (III.6) [BAR86], [BAR99], [BAR00b]:

- une couche d'entrée avec deux neurones,
- une couche cachée ayant 10 neurones,
- et un seul neurone de sortie.

Les fonctions d'activation adoptées sont la sigmoïde hyperbolique pour les neurones d'entrée et ceux de la couche cachée, alors que celle correspondant au neurone de sortie est linéaire.

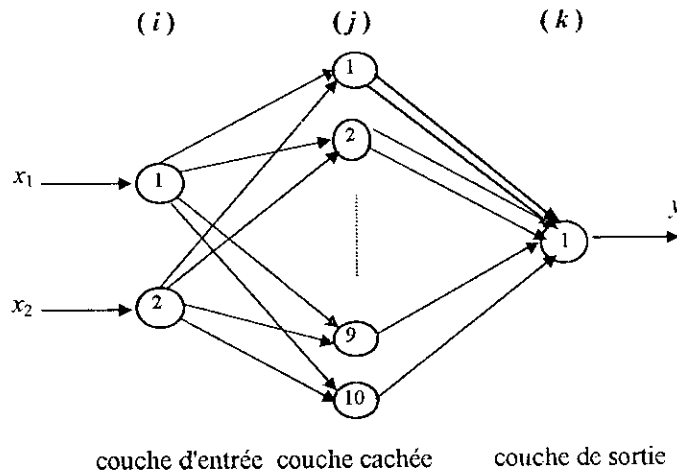


Figure III.6: Architecture des réseaux de neurones adoptés.

III.7.2 Processus d'entraînement du réseau de neurone de la vitesse (RNA_1)

Les entrées du réseau de neurone artificiel sont considérées comme étant l'erreur $e_1(t)$ et $e_1(t-1)$ entre la vitesse de référence Ω_{ref} et la vitesse réelle Ω , estimées respectivement aux instants t et $(t-1)$ afin de reproduire la dynamique du régulateur à structures variables. La sortie du réseau est considérée comme étant la variable de commande v_3^* .

La structure d'apprentissage du réseau de neurone proposée est représentée à la figure (III.7). Le procédé d'apprentissage est basé sur l'aptitude du réseau adopté à reproduire la loi de commande réalisée par le régulateur de vitesse par à structures variables (Fig.II.6).

La phase d'apprentissage de cette loi de commande s'effectue en deux étapes: une phase d'apprentissage et une autre de cross-validation.

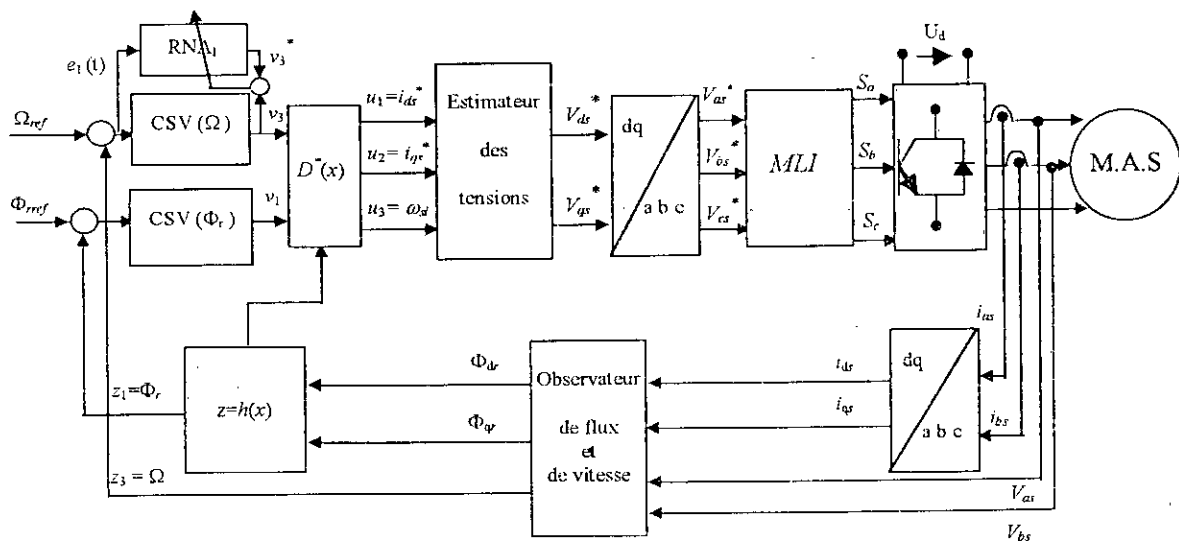


Figure III.7 : Processus d'apprentissage du réseau de neurone de vitesse RNA_1

Le patron d'exemples est obtenu à partir du fichier de valeurs obtenu lors de la simulation numérique effectuée avec des régulateurs par à structures variables de la figure (II.6). L'algorithme d'entraînement adopté est la rétro-propagation avec momentum. Nous avons utilisé 100 exemples uniformément distribués entre 0 et 0.3 sec, relevés de la simulation effectuée à la section (&II.4.2). Avec des poids initiaux choisis aléatoirement entre -0.5 et 0.5 et un pas d'adaptation de 0.3 et 0.2 pour la première et la deuxième couche respectivement. Ainsi, 2100 cycles se sont avérés suffisants pour accomplir l'apprentissage.

Un test de cross-validation a été ensuite effectué pour le même patron d'exemples en considérant dans cette étape 200 exemples. L'entraînement effectué lors de cette phase a montré qu'effectivement le réseau de neurone adopté a parfaitement appris la fonction réalisée par les régulateurs à structures variables. Par conséquent, il est apte à être injecté dans le processus de commande.

III.7.3 Processus d'entraînement du réseau de neurone du flux rotorique (RNA₂)

Une architecture de réseau similaire au cas précédent est adoptée pour l'obtention de la variable de commande v_1^* . Les entrées du réseau de neurone artificiel dans ce cas sont considérées comme étant l'erreur $e_2(t)$ et $e_2(t-1)$ entre le flux rotorique de référence Φ_{ref} et le flux réel Φ_r estimés respectivement aux instants t et $(t-1)$. La sortie du réseau est considérée comme étant la variable de référence v_1^* .

La structure d'apprentissage du réseau de neurone proposée est celle représentée à la figure (III.8). Il est important de noter que le régulateur de vitesse est considéré comme étant le réseau de neurone RNA₁.

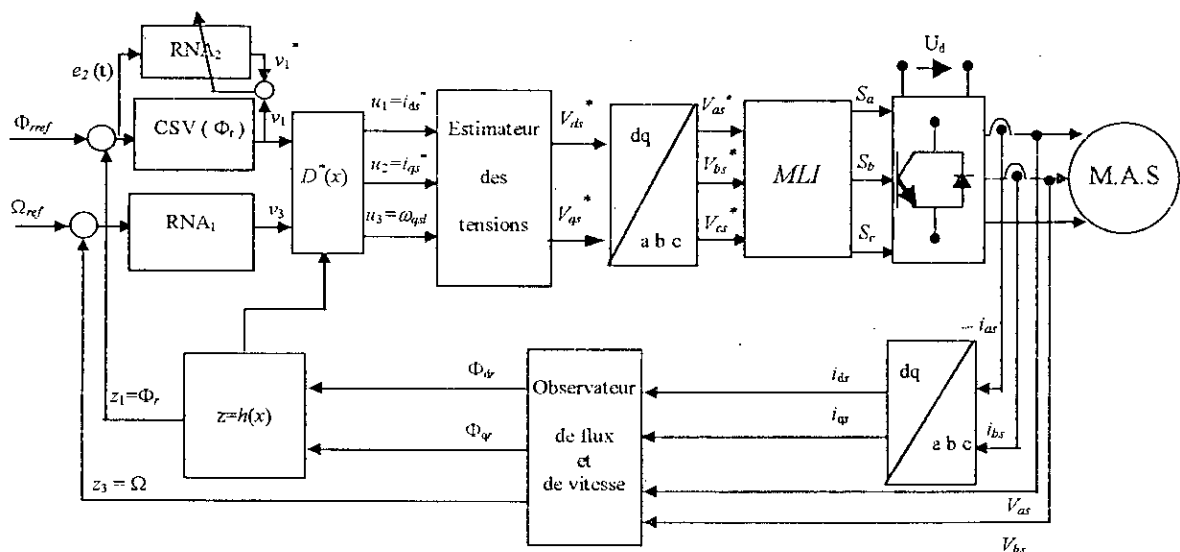


Figure III.8: Processus d'apprentissage du réseau de neurone de vitesse RNA₂

Le même patron d'exemples est utilisé pour l'entraînement du RNA₁. Dans ce cas, les pas d'adaptation choisis sont égaux à 0.2 et 0.1 pour la première et la deuxième couche respectivement. Pour accomplir

l'apprentissage de RNA₂, 2700 cycles sont suffisants pour que la sortie v_1^* suive celle du régulateur de flux à structures variables v_1 .

De la même façon que pour le cas précédent, un test de cross-validation a été effectué pour le même patron d'exemples. L'entraînement effectué lors de cette phase a montré que le réseau de neurone adopté a parfaitement appris la fonction réalisée par le régulateur à structures variables et qu'il est susceptible de le supplanter dans le processus de commande hybride de la figure (II.6).

III.7.4 Validation de la régulation de vitesse avec les régulateurs robustes par réseaux de neurones (RNAs)

Une simulation numérique de la commande avec régulateurs à base de réseaux de neurone a été effectuée dans les mêmes conditions que ceux de la section (&II.4.2) pour le cas où la commande discontinue est à relais (Fig. III.9), puis adoucie (Fig. III.10).

D'autre part, un test de robustesse a été effectué pour la commande adoucie avec une variation de $R_r = 90\% R_{r0}$ (Fig. III.11).

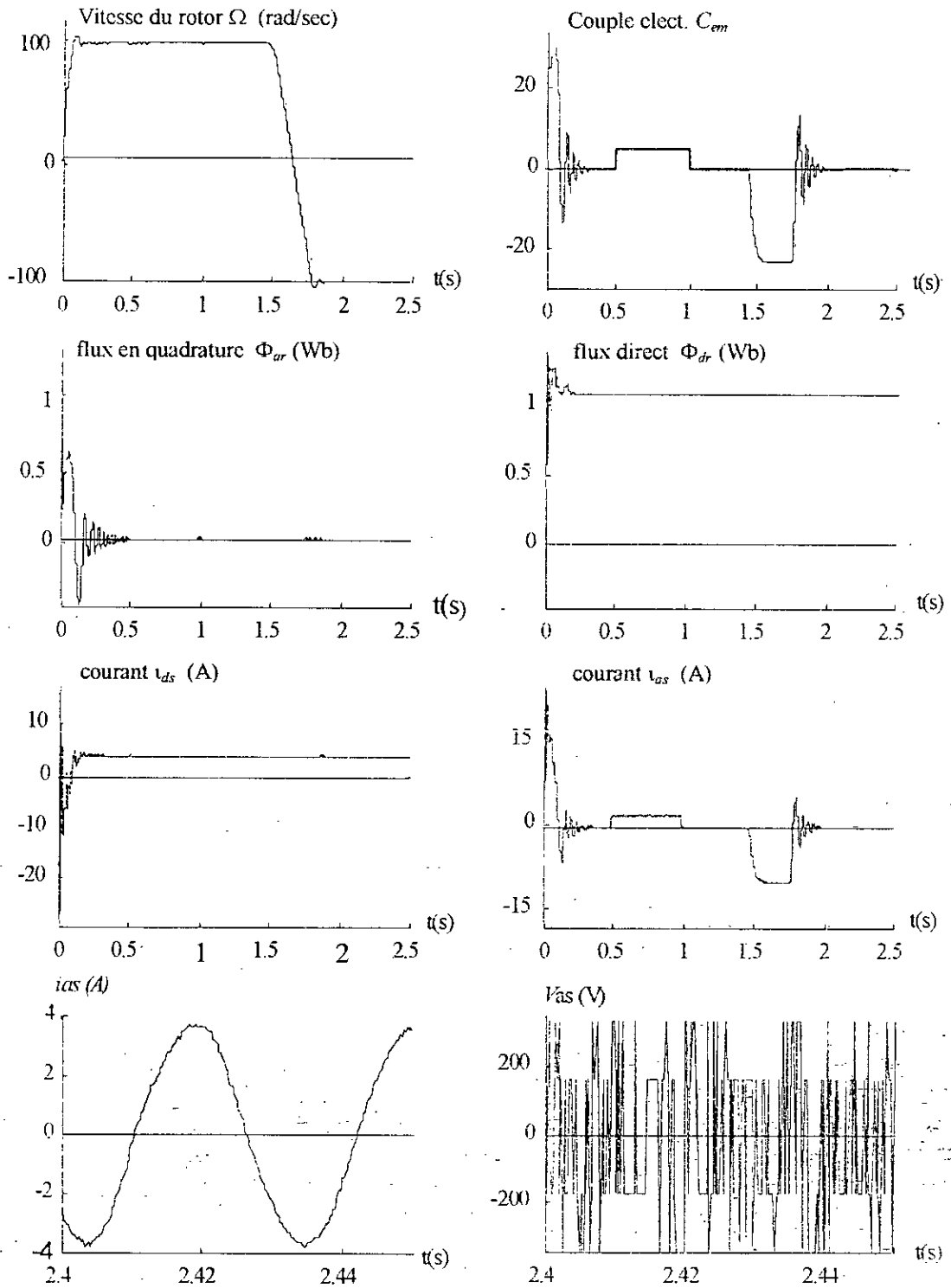


Figure III.9: Résultats de simulation de la régulation en vitesse avec des régulateurs à base de réseaux de neurones artificiels (cas relais).

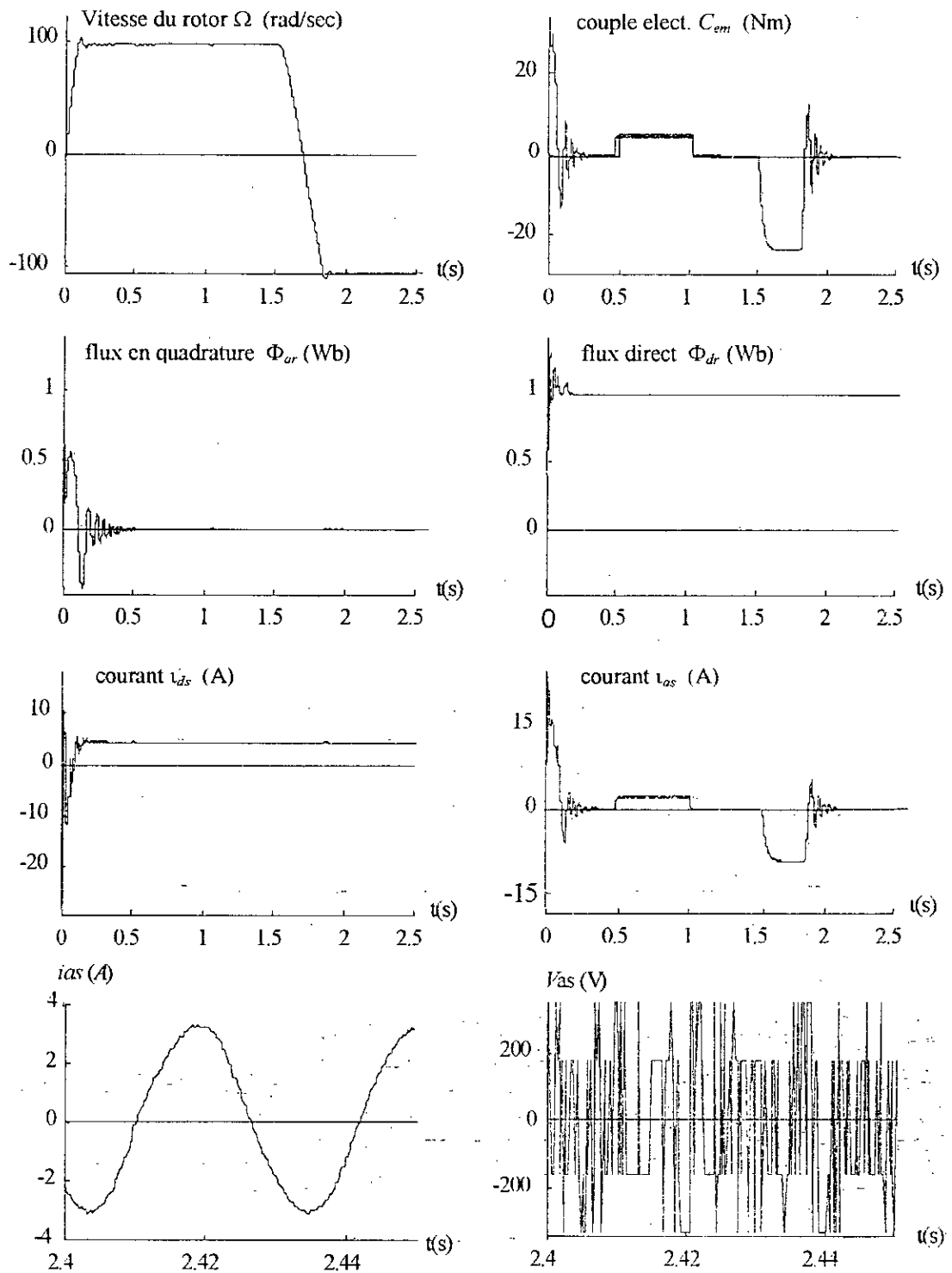


Figure III.10: Résultats de simulation de la régulation en vitesse avec des régulateurs à base de réseaux de neurones artificiels (cas adouci).

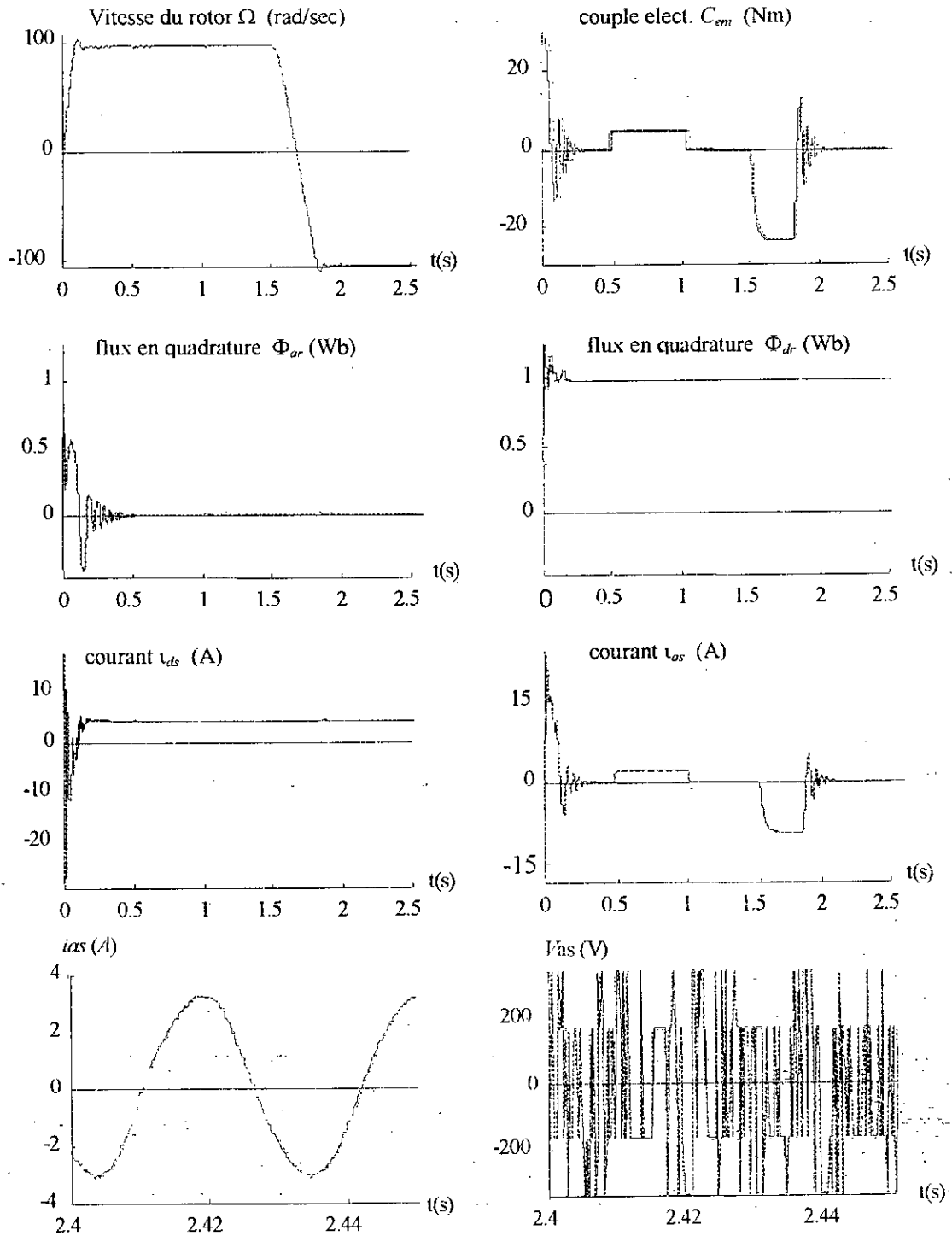


Figure III.11: Résultats de simulation de la régulation en vitesse avec des régulateurs à base de réseaux de neurones artificiels pour différentes variations de $R_r = 90\% R_{r0}$ (commande adoucie).

Il est clair que les résultats obtenus sont similaires à ceux obtenus avec des régulateurs à structures variables que ce soit dans le cas avec ou sans variations paramétriques. Cependant, ce qui est important à noter en plus, c'est une réduction des oscillations sur les courants comparés au cas avec réglage à structures variables. De plus, ce qui est important à relever c'est qu'ils donnent des résultats satisfaisants même dans le cas de variations des paramètres de la machine et l'inversion brusque de la vitesse, en sachant que dans les exemples d'apprentissage nous n'avons aucunement tenu compte de ces variations.

Ainsi, nous pouvons dire que les réseaux de neurones choisis ont pu non seulement implanter fidèlement les régulateurs à structures variables existants mais en plus introduisent une réduction du phénomène de chattering ce qui représente un apport considérable du point de vue performances du système. Par ailleurs, le pouvoir d'approximation des réseaux de neurones, ainsi que celui de généralisation ont été mis en évidence.

III.10 CONCLUSION

L'utilisation des réseaux de neurones artificiels a permis de résoudre plusieurs problèmes dans divers domaines d'application. Dans ce contexte, les propriétés offertes par ces derniers ont été mises à profit pour développer des régulateurs permettant de surmonter les restrictions liées à la reconstruction des dynamiques non-linéaires nécessaires pour générer les lois de commande développées dans le chapitre précédent.

Nous avons proposé d'utiliser les réseaux de neurones pour reconstituer en temps réel les dynamiques non-linéaires nécessaires pour générer les lois de commande des régulateurs de la commande hybride à structures variables. A cet effet, les réseaux de neurones multicouches sont utilisés. Les paramètres de ces réseaux sont ajustés par le biais des règles d'adaptation non-linéaires afin de ramener les erreurs de poursuite à des valeurs infiniment petites. Ces erreurs de poursuite sont prises en compte comme indice de performance pour optimiser le fonctionnement de réseau de neurones adoptés. De plus et comme avantage, la commande résultante prend en compte la variation des paramètres internes de la machine sans recourir à aucune procédure d'identification.

Ainsi, nous avons montré que ce type de réseaux peut apporter quelques légères améliorations entre autres : réduction du chattering par rapport à la commande à structures variables. Toutefois, ils présentent l'inconvénient d'être développés après l'établissement d'une commande standard adéquate et lors du fonctionnement, ils ne font que reproduire les fonctions des régulateurs à structures variables déjà existants (avec une capacité de généralisation limitée). De plus, ces réseaux de neurones à mode d'apprentissage supervisé utilisant un ensemble d'entrées/sorties adoptées, se trouvent limités par la nécessité d'avoir un meilleur pilote "teacher" (base de données adéquates) pour un meilleur apprentissage, dont la durée est importante dans notre cas. D'autre part, une fois entraînés, les poids synaptiques sont constants et, lors du fonctionnement, ils ne font que reproduire les exemples d'apprentissage avec une puissance de généralisation limitée. Ce qui ne leur permet pas de s'adapter complètement avec les différents régimes de fonctionnement du système commandé.

Néanmoins nous pouvons dire que la commande supervisée peut s'avérer très intéressante surtout pour des régulateurs difficilement implémentables ou des régulateurs qui n'ont pas une expression mathématique bien précise mais qui se présentent sous forme d'un fichier entrée/sortie.

Par conséquent, les restrictions suscitées nous ont contraint à élaborer un autre schéma de commande de la machine plus performante quant à la réduction du chattering, et ce en utilisant une autre technique de commande émergente appliquée pour la première fois pour la commande des moteurs dite : commande à structures variables floue basée sur le modèle flou de la machine asynchrone, qui fera l'objet du chapitre suivant.

Chapitre IV

Modélisation et Commande Robuste Floues

de la Machine Asynchrone

Chapitre IV

Modélisation et Commande Robuste Floues de la Machine Asynchrone

IV.1. INTRODUCTION

L'approche traditionnelle de la modélisation d'un système quelconque est basée sur une connaissance parfaite de ce dernier, ou de son comportement dynamique, plus particulièrement, sur des bases et relations mathématiques appropriées au développement du modèle en question.

D'autre part, pour des processus grossièrement connus, plusieurs méthodes ont été élaborées pour la mise en œuvre de leurs modèles et ceux de leurs contrôleurs. Parmi ces méthodes, nous pouvons citer la modélisation floue qui englobe les techniques de modélisation basées sur la logique floue.

Un système d'inférence flou est un système basé sur les concepts de la théorie de la logique floue, des règles floues de type "If-Then", et du raisonnement flou. En ce qui concerne les modèles flous à base de règles ils sont considérés comme étant des systèmes d'inférence flous qui utilisent des règles de type "If-Then" associées à des opérateurs logiques dans le but d'établir des relations qualitatives entre les diverses variables du système réel à modéliser. La nature de la base de règles de ces modèles flous permet l'utilisation d'informations exprimées en langage naturel, ce qui conduit à une certaine transparence dans l'analyse et l'interprétation de ces modèles. D'autre part, les modèles flous peuvent approximer une large classe de systèmes non-linéaires avec un certain degré d'exactitude.

Dans ce chapitre, nous allons présenter les notions de base et la classification des modèles flous. Notre investigation sera ensuite consacrée à une nouvelle approche de modélisation floue des systèmes incertains entre autre; le moteur asynchrone et ce, dans le but d'interpréter son système dynamique qui est relativement incertain comme un système dynamique flou, et de fournir par conséquent, le modèle flou correspondant. Ce dernier sera décrit par une équation différentielle floue, qui sera exploitée lors de la conception d'un contrôleur flou robuste approprié.

IV.2. ÉVALUATION DES SYSTEMES REELS ET FLOUS

Nous pouvons distinguer trois différents types de systèmes, comme le montre la figure (IV.1) [BAB98]:

- les systèmes décrits par des fonctions réelles, qui sont généralement des systèmes certains.
- les systèmes décrits par des fonctions intervalles.

- les systèmes décrits par des fonctions floues.

La différence entre la fonction intervalle et la fonction floue réside dans le fait que dans le premier cas, l'appartenance d'un point donné de l'espace est absolue, alors que dans le second, chaque point appartient à la fonction spécifique avec un degré de validité bien précis.

D'une manière générale, les systèmes flous peuvent être vus comme étant une généralisation de systèmes dans lesquels les variables d'entrée, les variables d'état et les paramètres sont des intervalles de valeurs réelles.

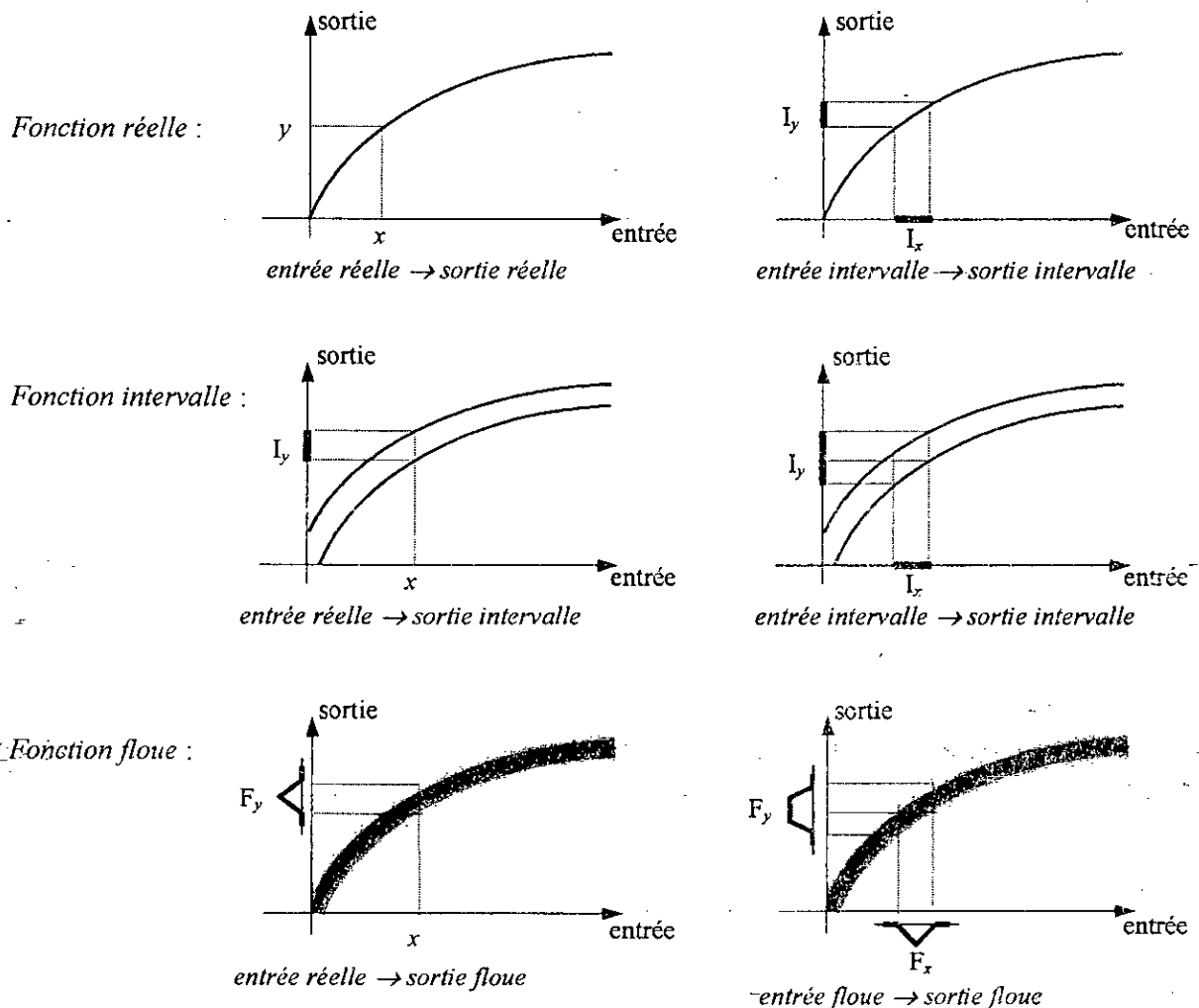


Figure IV.1 : Évaluation d'une fonction réelle, intervalle ou floue pour des arguments réels, intervalles ou flous.

IV.3. LES SYSTEMES D'INFERENCE FLOUS

Un système d'inférence flou est un système basé sur les concepts de la théorie de la logique floue, des règles floues de type "If-Then", et du raisonnement flou. Il a été appliqué dans plusieurs domaines tels

que le contrôle automatique, la classification des données, l'analyse des données...etc. A cause de sa nature multi-disciplinaire, il est connu par différents noms, entre autres : un système d'inférence flou [JAN92], système flou à base de règles [BAB98], système expert flou, modèle flou [TAK85], contrôleur (ou régulateur) flou [MAM75], [LEE90] ou simplement un système flou.

En effet, la sortie peut être une conclusion, une décision ou une action de commande [KEL97].

Les ensembles flous sont introduits de plusieurs façons dans les systèmes :

1. *Dans la spécification des paramètres du système* : Un système est défini par une équation différentielle (algébrique ou en différence) dans laquelle les paramètres sont des nombres flous au lieu de nombres réels. Ce qui est une manière de représenter les incertitudes paramétriques du système.
2. *Dans la description du système* : Un système peut être entièrement défini par une série de règles floues de type "If-Then" (ou *Si-Alors*) ou par une relation floue.

IV.3.1. Configuration de base d'un système d'inférence flou

Un système d'inférence flou est constitué de quatre parties principales (Fig. IV.2) à savoir : une base de connaissances, une interface de fuzzification (ou interface de quantification floue), un moteur d'inférence et une interface de défuzzification.

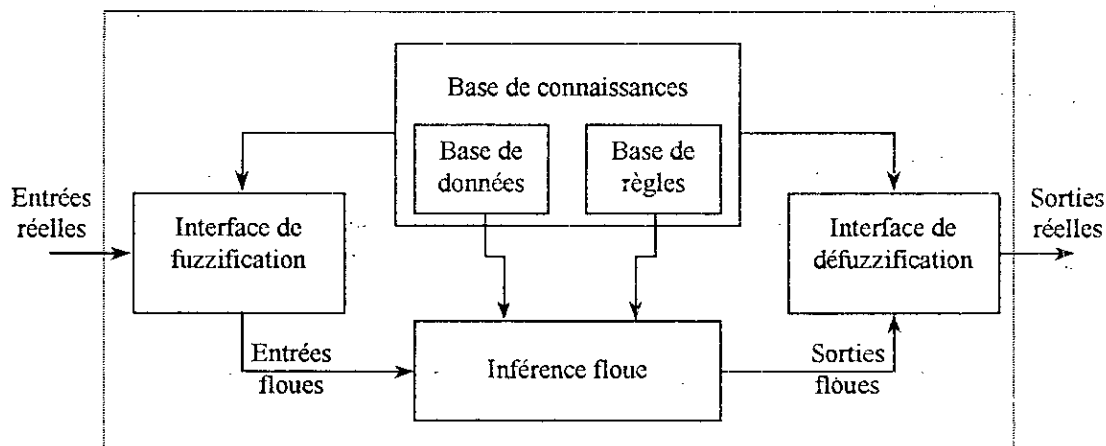


Figure.IV.2 : Structure d'un système d'inférence flou

1. *L'interface de fuzzification* a pour but de déterminer les degrés d'appartenance des variables d'entrées aux ensembles flous correspondants. Elle reçoit des valeurs réelles (ou floues), représentant les entrées du système d'inférence flou, et les transforme en des termes linguistiques sous forme de valeurs entre 0 et 1.

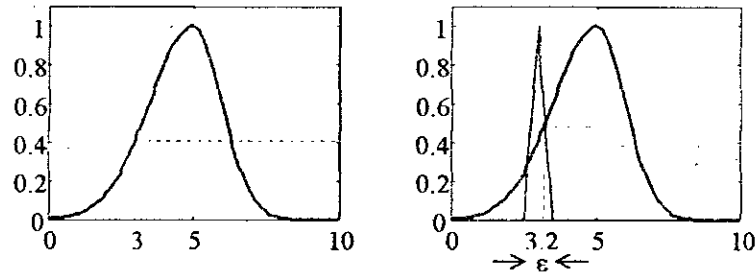


Figure. IV.3 : Fuzzification, a) entrée réelle, b) entrée floue

La figure (IV.3) montre la fuzzification d'une valeur exacte x_0 et d'une valeur x_0 entachée d'une incertitude ε .

2. La base de connaissances est constituée de deux blocs :

– La base de données : elle contient des informations qui caractérisent le système flou, telles que les univers de discours et les ensembles flous associés aux termes linguistiques d'entrées et de sorties.

– La base de règles : elle contient un nombre de règles de type If-Then de la forme :

$$\text{IF Condition (ou antécédent) THEN Conséquence} \quad (\text{IV.1})$$

La complexité des systèmes flous dépend fortement du nombre d'entrées et du nombre d'ensembles flous définis dans chaque univers de discours. La taille maximale de la base de règles croît exponentiellement avec le nombre des variables d'entrée du système flou :

$$\text{Nombre maximal des règles} = N \binom{x_4}{x_3} \quad (\text{IV.2})$$

N , n et M_i : sont respectivement le nombre de sorties, le nombre d'entrées et le nombre d'ensembles flous délimités par l'univers de discours de la $i^{\text{ième}}$ entrée.

La réduction de la complexité de la base des règles est un domaine de recherche très important pour les systèmes flous complexes [KEL97].

3. L'unité d'inférence floue (unité de décision) : elle crée les variables floues de sortie en utilisant les opérations sur les fonctions d'appartenance.

Une fois l'antécédent évalué, son degré de validité (ou de certitude) est attribué à la conséquence de chaque règle. L'implication est fonction du degré de validité de l'antécédent β et de la fonction d'appartenance de la sortie $\mu_B(y)$. Plusieurs méthodes d'implication ont été proposées dans la littérature, qui conservent et généralisent l'implication de la logique booléenne [BOU95]. L'implication de Lukasiewicz définie par l'équation (IV.3) en est un exemple.

$$\mu_B'(y) = \text{imp}(\beta(x_0), \mu_B(y)) = \min(1 - \beta(x_0) + \mu_B(y), 1) \quad (\text{IV.3})$$

Malgré cela, la règle est en général modélisée par un opérateur de T-norme, bien que cette représentation ne généralise pas l'implication classique. Les exemples les plus classiques sont les implications de Larsen et de Mamdani définies par les opérations de produit ou de minimum respectivement (Fig IV.4) :

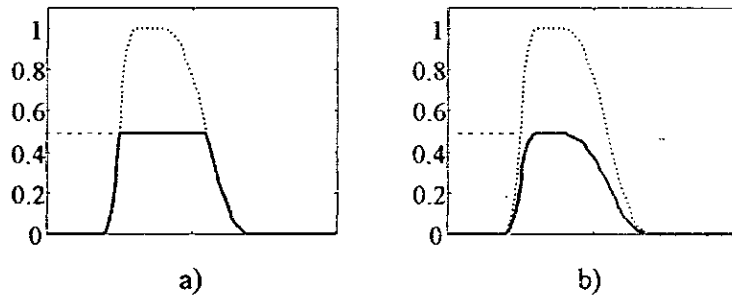


Figure. IV.4 : Implication, a) par saturation, b) par produit

$$\begin{aligned} \mu_B'(y) &= \text{imp}(\beta(x_0), \mu_B(y)) = \beta(x_0) \cdot \mu_B(y) \quad (\text{produit}) \\ \mu_B'(y) &= \text{imp}(\beta(x_0), \mu_B(y)) = \min(\beta(x_0), \mu_B(y)) \quad (\text{saturation}) \end{aligned} \quad (\text{IV.4})$$

4. **Agrégation** : Dans un système flou, plusieurs règles sont actionnées en sortie, ainsi, plusieurs ensembles flous de sortie sont concernés. L'agrégation détermine la manière de combiner les sorties des différentes règles pour construire un seul ensemble flou dans la conséquence.

Pour deux conséquences A et B , la combinaison de ces différentes règles se fait à l'aide de l'opérateur OU. Le résultat de l'agrégation est obtenu par plusieurs méthodes (généralement par une opération de T-conorme) [KEL97], [MAT99], à savoir :

– *Méthode probabiliste* (Fig. IV.5a) :

$$C = \text{agg}(\mu_A(y), \mu_B(y)) = \mu_A(y) + \mu_B(y) - \mu_A(y) \cdot \mu_B(y) \quad (\text{IV.5})$$

– *Méthode de maximum* (Fig. IV.5b) :

$$C = \text{agg}(\mu_A(y), \mu_B(y)) = \max(\mu_A(y), \mu_B(y)) \quad (\text{IV.6})$$

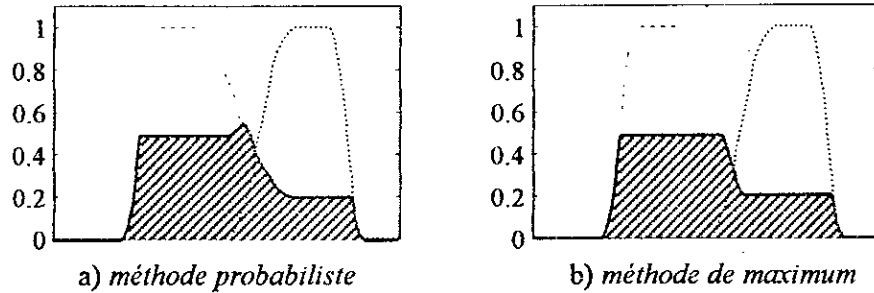


Figure. IV.5 : Différentes méthodes d'agrégation

5. *L'interface de défuzzification* : elle convertit l'ensemble flou résultant de l'agrégation en une seule valeur réelle dans le but de donner plus de certitude à la sortie du système d'inférence flou.

Il existe plusieurs méthodes pour réaliser cette opération, parmi lesquelles nous citons :

– *La méthode du maximum* : Elle consiste à ne prendre en compte que les valeurs possédant un degré maximum d'appartenance à l'ensemble flou agrégé. Cette technique est simple à appliquer, mais peu employée étant donné qu'elle est basée sur l'opérateur maximum qui privilégie une seule règle parmi celles qui sont présentes.

– *La méthode de la moyenne pondérée (méthode des hauteurs pondérées ou méthode de la moyenne des maximums)* : Celle-ci considère comme valeur de sortie la moyenne des valeurs préconisées par chaque règle et pondérées par leur degré respectif de validité.

$$\text{mp}(y) = \frac{\sum_{i=1 \dots N} \bar{y}_i \cdot \mu_{B_i}(\bar{y}_i)}{\sum_{i=1 \dots N} \mu_{B_i}(\bar{y}_i)} \quad (\text{IV.7})$$

avec \bar{y}_i la valeur de sortie préconisée par la $i^{\text{ème}}$ règle, et qui peut correspondre au maximum de la fonction d'appartenance de l'ensemble flou B_i tel que :

$$\bar{y}_i = \sup_{y \in Y} (\mu_{B_i}(y)) \quad (\text{IV.8})$$

B_i : l'ensemble flou résultant de la $i^{\text{ème}}$ règle.

– *La méthode du centre de gravité* : Cette méthode, qui est la plus utilisée, génère le centre de gravité de la surface engendrée par la fonction d'appartenance de l'ensemble flou agrégé. L'abscisse du centre de gravité peut être déterminée par la relation suivante :

$$\text{cdg}(y) = \frac{\int_Y y \cdot \mu_{B_i}(y) \, dy}{\int_Y \mu_{B_i}(y) \, dy} \quad (\text{IV.9})$$

Comme elle est très coûteuse en temps de calcul, certains préfèrent utiliser des formes simples de fonctions d'appartenance, telle que la forme triangulaire, dont les surfaces sont déterminées en fonction de leurs paramètres. Alors que d'autres préfèrent la méthode de défuzzification de la moyenne pondérée, pour sa simplicité et pour la facilité de sauvegarde des fonctions d'appartenance de sortie [BOU99].

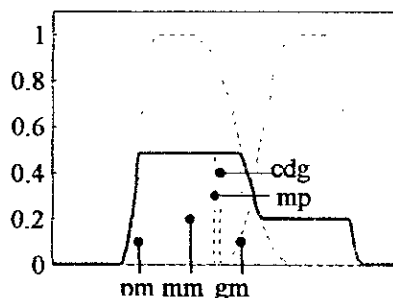


Figure. IV.6 : Différentes méthodes de défuzzification

(*cdg* : centre de gravité, *mp* : moyenne pondérée, *pm* : petit maximum, *mm* : milieu des maxima, *gm* : grand maximum)

IV.4. LES MODELES FLOUS A BASE DE REGLES

Dans les modèles flous à bases de règles, les relations entre les variables d'entrée et de sortie sont assurées par des règles floues de type "If-Then" qui s'expriment comme suit :

$$\text{IF } \tilde{x} \text{ is } F_x \text{ THEN } \textit{conséquence} \quad (\text{IV.10})$$

Pour un modèle flou multivariable :

\tilde{x} : représente un vecteur de variables réelles ou un vecteur de variables floues ou une région floue variable.

F_x : est un ensemble flou obtenu par le produit cartésien d'ensembles flous ayant chacun une fonction d'appartenance ou d'une manière plus générale, par une région floue définie par une *fonction d'appartenance multivariables* $\mu_x(x)$ définie dans l'intervalle $[0, 1]$ comme suit :

$$\mu_x(x) : x \in \mathbb{R}^n \rightarrow [0, 1] \quad (\text{IV.11})$$

Dans des cas particuliers, cette fonction d'appartenance multivariables peut être déterminée par des combinaisons logiques (min, max, prod, ...) entre des fonctions d'appartenance monovariables. Des opérateurs flous, tels que la conjonction, la disjonction et la négation (complément), sont utilisés pour combiner les termes flous des entrées.

Concernant l'antécédent du modèle flou, trois types de partitionnement peuvent être considérés à savoir [BAB98], [JAN92], [JAN95] :

- le partitionnement homogène,
- le partitionnement par concaténation d'ensembles flous,
- et celui optimal.

IV.4.1. Partitionnement homogène

Soit un système flou, constitué de K règles, sous la forme conjonctive présentée sous la forme générale suivante:

$$\mathfrak{R}_i: \text{IF } x_1 \text{ is } F_{x_1}^i \text{ AND ... AND } x_n \text{ is } F_{x_n}^i \text{ THEN ...} \quad (\text{IV.12})$$

Nous remarquons que ce modèle est un cas particulier de la forme générale (IV.1), avec l'ensemble flou F_x qui représente le produit cartésien des ensembles flous $F_{x_j}^i$; ($j : 1, \dots, n$).

Dans ce cas, le degré de validité de l'antécédent de la règle, pour un vecteur d'entrée réel, $x = (x_1, \dots, x_n)^t$ sera évaluée de la manière suivante:

$$\beta_i = \mu_{F_x^i}(x_1, x_2, \dots, x_n) = \mu_{F_{x_1}^i}(x_1) \wedge \mu_{F_{x_2}^i}(x_2) \wedge \dots \wedge \mu_{F_{x_n}^i}(x_n) \quad (\text{IV.13})$$

β_i ($i : 1, \dots, K$) : le degré de validité de l'antécédent de la règle \mathfrak{R}_i .

\wedge : opérateur de la norme T-norme (minimum, produit, ...).

L'ensemble des règles, dans ce cas, divise l'espace du vecteur d'entrées en des régions floues dont chacune est considérée comme étant le produit cartésien des ensembles flous des entrées correspondantes comme le montre la figure (IV.7).

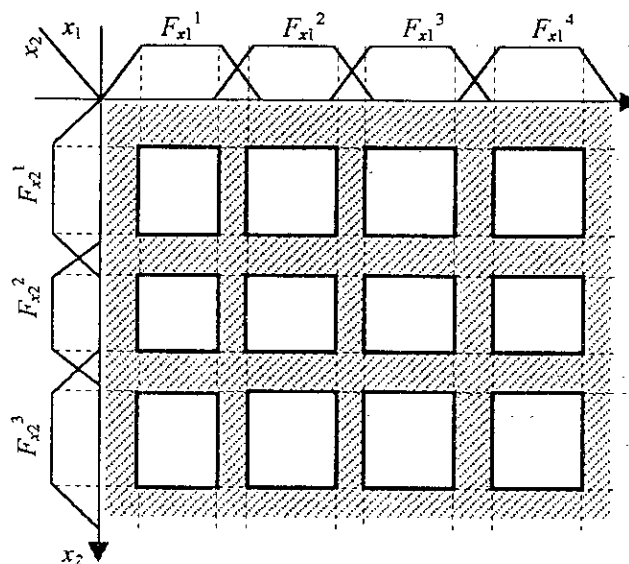


Figure. IV.7 : Partition floue homogène de l'espace d'entrée schématisée par les noyaux des ensembles flous

Le nombre maximal des règles destinées à couvrir l'espace global du vecteur d'entrée est donné par la relation suivante:

$$K = N \prod_{i=1}^n M_i \quad (\text{IV.14})$$

avec

N : nombre de variables de sortie

M_i : nombre d'ensembles flous pour l'entrée x_i

n : nombre de variables d'entrée (la dimension de l'espace d'entrée)

IV.4.2. Partitionnement par concaténation d'ensembles flous

Afin de réduire la complexité de la base de règles, nous pouvons concaténer certaines régions floues de la partition homogène en utilisant des conjonctions, des disjonctions et/ou des négations sur les ensembles flous de l'antécédent.

Par exemple, en se référant à la figure (IV.8), la région floue $F_x^{(1)}$ est obtenue en concaténant les deux régions floues $(F_{x_1}^1 \times F_{x_2}^1)$ et $(F_{x_1}^1 \times F_{x_2}^2)$.

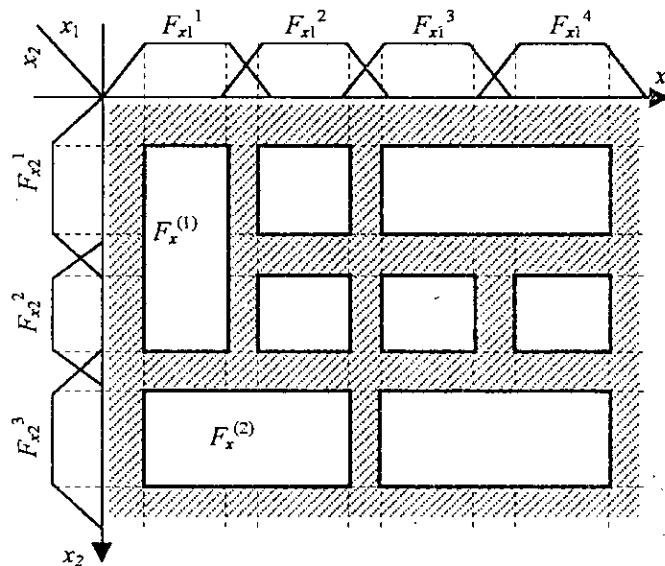


Figure IV.8: Partition floue obtenue par concaténation d'ensembles flous d'une partition homogène

La règle sera alors décrite par :

$$\text{IF } x_1 \text{ is } F_{x_1}^1 \text{ AND } x_2 \text{ is not } F_{x_2}^3 \text{ THEN ...} \quad (\text{IV.15})$$

La fonction d'appartenance de cette région floue est obtenue par l'expression suivante :

$$\mu_{F_x}(x) = \mu_{F_{x_1}}(x_1) \wedge (1 - \mu_{F_{x_2}}(x_2)) \quad (\text{IV.16})$$

L'avantage de cette nouvelle structure réside dans le fait que la cardinalité de la partition floue de l'espace d'entrées est minimisée. Ce qui se traduit par une réduction de la complexité du modèle.

IV.4.3. Partitionnement optimal

Dans le cas le plus général, il n'y a pas de restriction sur la forme des régions floues comme le montre la figure (IV.9), lesquelles sont définies par des fonctions d'appartenances multivariables.

Le nombre de régions floues nécessaire pour couvrir l'espace du vecteur d'entrées peut être inférieur à celui des cas précédents. Pour les systèmes complexes multivariables, cette partition est considérée comme étant la représentation la plus efficace.

Un point donné dans cet espace, pour une région floue donnée, fournit en même temps une information sur le degré d'appartenance à cette région de l'entrée x_1 et sur celui de l'entrée x_2 .

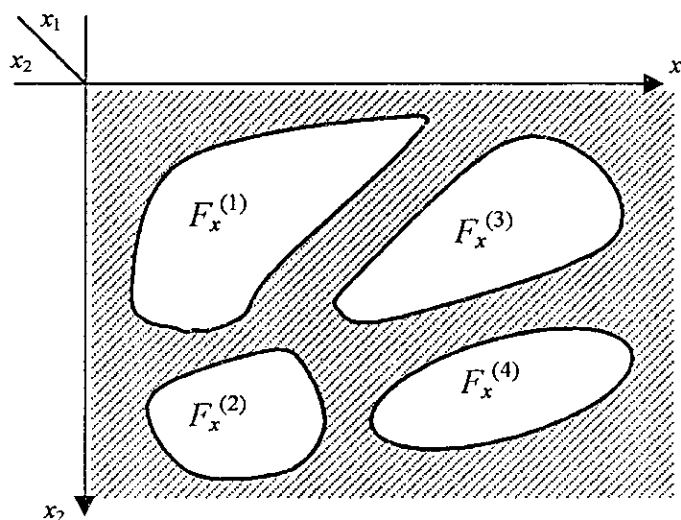


Figure.IV.9 : Partition floue optimale

VI.4.4. Différents types de modèles flous

Suivant la nature de la conséquence, quatre types de modèles flous peuvent être considérés à savoir:

- *Modèle flou linguistique* : dans lequel la conséquence est un ensemble ou une région floue F_y .
- *Modèle flou relationnel* : qui est une généralisation du modèle flou linguistique, caractérisé par l'existence d'une relation floue entre les termes de la conséquence et ceux de l'antécédent.

- *Modèle flou de type Takagi-Sugeno* : dans ce cas la conséquence de chaque règle est une fonction réelle.
- *Modèle flou de Tsukamoto* : la conséquence d'un tel modèle est un ensemble flou ayant une fonction d'appartenance monotone.

Dans le présent travail, nous nous sommes intéressés à un nouveau type de modélisation floue introduite par Ben-Ghalia, qui est une technique basée sur l'élaboration d'un modèle flou linguistique et que nous appliquons pour la première fois à la machine asynchrone à cage. Pour ce faire, nous avons jugé utile de donner au préalable un aperçu sur le modèle flou linguistique.

IV.4.4.1. Le modèle flou linguistique

La forme générale d'un modèle flou linguistique basé sur des règles de type "If-Then" est donnée par :

$$\mathfrak{R}_i : \text{IF } \tilde{s}_i \text{ is } F_x^i \text{ THEN } y \text{ is } F_y^i \quad (\text{IV.17})$$

avec :

\mathfrak{R}_i : la $i^{\text{ème}}$ règle ; $i = 1, \dots, K$

\tilde{x} : variable linguistique d'entrée (antécédent) correspondante au vecteur des variables d'état et/ou des variables d'entrée, $x \in \mathbb{R}^n$.

\tilde{y} : variable linguistique de sortie (conséquence) correspondante au vecteur de sortie $y \in \mathbb{R}^q$.

F_x^i et F_y^i : sont des ensembles ou des régions floues constantes.

Les fonctions d'appartenance correspondantes à l'antécédent et à la conséquence sont définies par des fonctions multivariées :

$$\begin{aligned} \mu_{F_x^i}(x) &: x \in \mathbb{R}^n \rightarrow [0, 1] \\ F_s^{(k_1 \rightarrow k_2)} &: y \in \mathbb{R}^q \rightarrow [0, 1] \end{aligned} \quad (\text{IV.18})$$

➤ *Mécanisme d'inférence* :

Le mécanisme d'inférence d'un modèle flou linguistique peut être décrit par les étapes suivantes :

◆ *Calcul du degré de validité de chaque règle* :

Supposons que le vecteur x est une variable linguistique connue par un ensemble flou A . Le degré d'appartenance de ce vecteur à F_x^i ($i = 1, \dots, n$) correspond au degré de validité de l'antécédent (\tilde{s}_i is F_x^i) de la $i^{\text{ème}}$ règle et est calculé comme suit (Fig. IV.10) :

$$\beta_i = \max_{U_x} \left(\mu_A(x) \wedge \mu_{F_x^i}(x) \right) \quad ; \quad i = 1, \dots, K \quad (\text{IV.19})$$

où X est l'univers de discours ou l'espace du vecteur x .

$\mu_A(x)$ est la fonction d'appartenance du vecteur x à l'ensemble flou A .

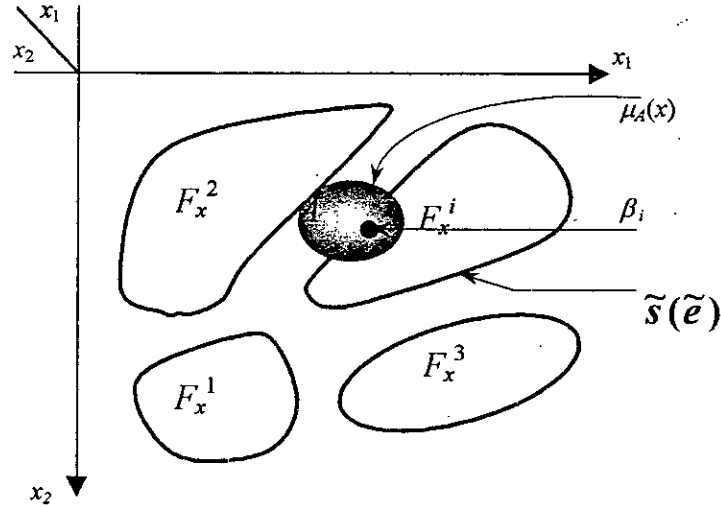


Figure. IV.10 : Fuzzification ou obtention du degré de validité de l'antécédent d'une règle

Dans le cas où F_x^i et A sont des produits cartésiens des ensembles flous $F_{x_j^i}$ et A_j ($j=1, \dots, n$) respectivement, c.-à-d. que l'univers de discours constitue une partition floue homogène (Fig.IV.7), l'équation (IV.19) peut être développée comme suit :

$$\mu_A(x) \wedge \mu_{F_x^i}(x) = \left(\mu_{A^1}(x) \wedge \dots \wedge \mu_{A^n}(x) \right) \wedge \left(\mu_{F_{x_1}^i}(x) \wedge \dots \wedge \mu_{F_{x_n}^i}(x) \right) \quad (\text{IV.20})$$

Suivant la propriété de l'associativité des opérateurs T-norme, l'équation (IV.20) donne :

$$\begin{aligned} \mu_A(x) \wedge \mu_{F_x^i}(x) &= \left(\mu_{A^1}(x) \wedge \mu_{F_{x_1}^i}(x) \right) \wedge \dots \wedge \left(\mu_{A^n}(x) \wedge \mu_{F_{x_n}^i}(x) \right) \\ &= \bigwedge_{j=1:n} \left(\mu_{A^j}(x) \wedge \mu_{F_{x_j}^i}(x) \right) \end{aligned} \quad (\text{IV.21})$$

avec $\bigwedge_{j=1:n}$: opération de T-norme multivariable.

D'où l'expression du degré de validité de l'antécédent de la $i^{\text{ème}}$ règle :

$$\beta_i = \max_X \bigwedge_{j=1:n} \left(\mu_{A^j}(x) \wedge \mu_{F_{x_j}^i}(x) \right) = \bigwedge_{j=1:n} \max_X \left(\mu_{A^j}(x) \wedge \mu_{F_{x_j}^i}(x) \right) \quad (\text{IV.22})$$

Ou encore :

$$\beta_i = \bigwedge_{j=1:n} \beta_i^j \quad (\text{IV.23})$$

avec $\beta_i^j = \max_x \left(\mu_{A_j}(x) \wedge \mu_{F_{x_j}}(x) \right)$ comme illustré par la figure (IV.11).

Par conséquent, le degré de validité de l'antécédent de la règle peut être déduit à partir d'une opération de T-norme sur les degrés d'appartenance du vecteur x aux ensembles flous qui constituent la partition floue homogène de l'espace d'entrées.

Dans certains cas, les règles peuvent être pondérées par des poids permettant d'augmenter ou de réduire l'importance d'une règle par rapport aux autres [MAT99]. Ainsi, le degré de validité de la règle sera une opération de T-norme (produit en général) entre le degré de validité de l'antécédent β_i et le facteur de pondération correspondant.

◆ *Déduction de l'ensemble flou de chaque règle :*

La région floue de la sortie y est déduite par une opération d'inférence, qui peut être définie par une opération de T-norme [BOU95] :

$$\mu_{B_i}(y) = \beta_i \wedge \mu_{F_i}(y) \quad (\text{IV.24})$$

avec : $i = 1, \dots, K$

$y \in Y$; avec Y l'espace du vecteur de sortie y .

B_i : l'ensemble flou conséquent de la règle \mathfrak{R}_i .

◆ *Agrégation des ensembles flous de sortie des règles :*

Pour déduire une seule région floue de sortie résultant de l'ensemble des règles, une opération de T-co-norme (max, somme, ...) peut être utilisée :

$$\mu_B(y) = S_{1 \leq i \leq K} \left(\mu_{B_i}(y) \right) ; y \in Y \quad (\text{IV.25})$$

$S(\cdot)$: opération de T-conorme.

♦ *Défuzzification:*

La région floue résultante de l'agrégation doit être défuzzifiée pour fournir des valeurs réelles à la sortie du modèle. La défuzzification peut être réalisée par l'une des plusieurs méthodes sus citées qui existent.

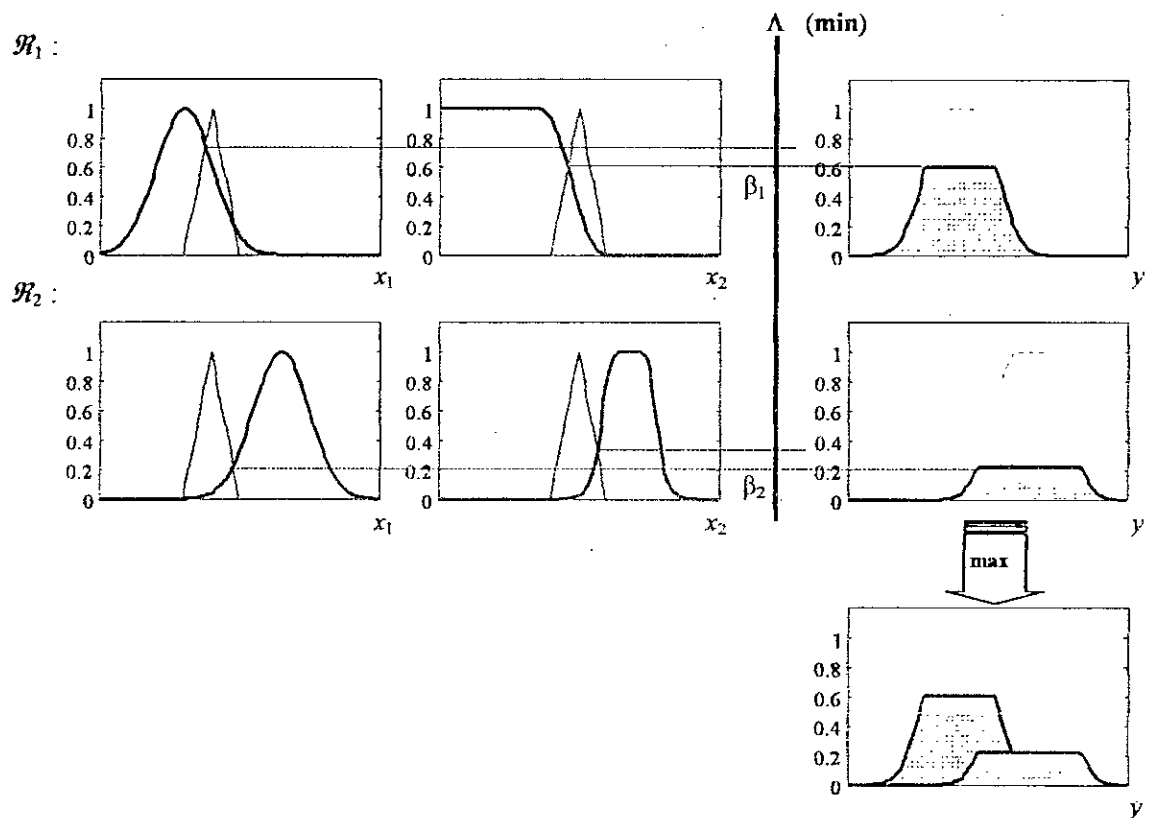


Figure. IV.11 : Système d'inférence flou de Mamdani utilisant min et max pour les opérations ET et OU respectivement

IV.5. ÉLABORATION D'UN MODELE FLOU- MODELISATION FLOUE

Pour élaborer un modèle flou, deux étapes d'identification sont nécessaires : une identification structurelle et une identification paramétrique [SUG88], [BAB98]. Dans la première étape, nous définissons la base de règles et le type du modèle. Dans la seconde étape, nous déterminons les formes des ensembles flous et les paramètres des fonctions d'appartenance.

Nous pouvons citer trois sources d'information pour construire des modèles flous :

1- la connaissance linguistique préalable d'un expert humain :

Le système dans ce cas peut être inconnu. L'expert doit avoir une connaissance préalable sur le fonctionnement de son système et le comportement du contrôleur. De plus, lorsqu'il s'agit de

problèmes complexes et multivariés, il est difficile d'établir directement une base de règles permettant le contrôle flou du système [BOR96].

2- les données de mesure :

Pour plusieurs processus, les données sont disponibles sous forme d'enregistrements ou de résultats d'identification. La conception et le réglage des modèles flous basés sur des données d'E/S est souvent appelé : *identification floue* [SUG88], [BRA96], [LU98], [MAN99].

3- la formulation mathématique grossière du modèle :

Cette méthode utilise la modélisation physique du système afin de fournir une représentation floue correspondante en tenant compte des incertitudes et ce quelle que soit la nature de la conséquence, donc, du type de modèle. Par conséquent, la base de règles est obtenue automatiquement, quel que soit le nombre de variables d'entrée du système flou, assurant ainsi une parfaite imitation du système [GHA96a], [PED97].

La modélisation linguistique à laquelle nous nous intéressons proposée par de Ben-Ghalia est un cas particulier du troisième type de modélisation suscitée et dont le principal avantage réside dans le fait qu'il s'adapte bien au problème qui nous concerne.

IV.6. MODELISATION FLOUE LINGUISTIQUE DE BEN-GHALIA

Cette méthode est applicable pour la classe de système incertains décrits par une équation différentielle du type:

$$\dot{x} = f(x, \eta) + g(x, \eta) \cdot u \quad (\text{IV.26})$$

où : $x = [x_1, x_2, \dots, x_n]^t \in \mathbb{R}^n$ le vecteur d'état du système,

$u(t) = [u_1, u_2, \dots, u_m]^t \in \mathbb{R}^m$ le vecteur d'entrée,

$\eta(t) \in \mathbb{R}^p$: le vecteur des incertitudes,

f, g : sont respectivement un vecteur de dimension n et une matrice de fonctions de dimensions $n \times m$.

Toutes les incertitudes dans le modèle du système sont incluses dans le vecteur $\eta(t)$. Ce dernier représente les incertitudes paramétriques, non paramétriques et structurelles. La figure (IV.12) montre la famille des systèmes dynamiques engendrée par les incertitudes η^i . A chaque valeur possible de $\eta(t) = \eta^i$ correspond un comportement particulier du système.

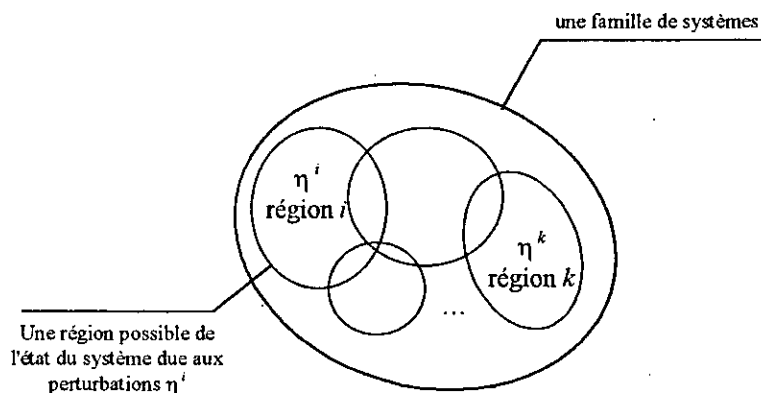


Figure. IV.12 : Effet des incertitudes sur un système

Dans cette étude, les incertitudes non-paramétriques sont vues comme des dynamiques non modélisées et sont considérées comme des perturbations additives.

L'approche de modélisation floue de Ben-Ghalia [GHA94], [GHA96a], [GHA97] consiste à trouver une représentation basée sur des ensembles flous et ce, en faisant appel au principe d'extension appliqué à l'équation (IV.26):

$$\tilde{\mathbf{x}} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) \tilde{+} \tilde{\mathbf{g}}(\tilde{\mathbf{x}}) \tilde{\cdot} \tilde{\mathbf{u}} \quad (\text{IV.27})$$

où : $\tilde{\mathbf{x}}$, $\tilde{\mathbf{y}}$ sont des vecteurs flous dont les composants sont des variables flous.

$\tilde{\mathbf{f}}$, $\tilde{\mathbf{g}}$ sont respectivement un vecteur et une matrice contenant des fonctions flous.

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = \begin{pmatrix} \tilde{f}_1(\tilde{x}_1, \dots, \tilde{x}_n) \\ \vdots \\ \tilde{f}_n(\tilde{x}_1, \dots, \tilde{x}_n) \end{pmatrix}; \quad \tilde{\mathbf{g}}(\tilde{\mathbf{x}}) = [g_{ij}(\tilde{x}_1, \dots, \tilde{x}_n)]_{\substack{i=1, \dots, n \\ j=1, \dots, m}} \quad (\text{IV.28})$$

$\tilde{+}$, $\tilde{\cdot}$ sont respectivement l'addition et la multiplication étendue sur les variables floues [DRI93].

L'équation (IV.27) est appelée *équation différentielle floue*. Les modèles décrits par cette équation sont appelés *modèles dynamiques flous*.

Notons que l'élément $\eta(t)$ n'apparaît plus dans la nouvelle représentation du système. Son effet est pris en compte implicitement lors de la représentation du modèle par des ensembles flous.

IV.6.1. Condition d'existence de modèle flou selon la méthode adoptée

Concernant l'équation (IV.26), trois suppositions sont à considérer:

1. L'espace d'état $\mathbf{x} = [x_1, \dots, x_n]^T$ appartient à un ensemble borné U_x , dont les bornes sont connues.

$$U_x = U_{x_1} \times U_{x_2} \times \dots \times U_{x_n} \subset \mathbb{R}^n. \quad (\text{IV.29})$$

2. L'élément représentant les incertitudes $\eta(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}^p$ est mesurable au sens de Lebesgue et ses valeurs $\eta(t)$ appartiennent à un ensemble borné $P \subset \mathbb{R}^p$, $\forall t \in \mathbb{R}_+$.
3. Le vecteur de fonctions $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ et la matrice $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^m$ sont fortement Carathéodory, $\forall \eta \in P$ (Annexe C).

Les deux dernières suppositions confirment l'existence d'une possibilité de stabilité globale du système (IV.26) soumis au contrôleur approprié qui devient lui-même Carathéodory [GHA96b], [HAL69].

IV.6.2. Algorithme de la modélisation floue de Ben-Ghalia

Pour définir les ensembles et les règles appropriés correspondants à l'équation différentielle floue (IV.27), nous présentons l'algorithme de Ben-Ghalia qui s'effectue en quatre étapes successives.

Étape 1 : Partitionnement de l'espace d'entrée

La première étape de la modélisation floue considérée consiste à construire une partition floue homogène de l'univers de discours U_{x_i} de chaque composante x_i du vecteur d'état, $i = 1, \dots, n$. C'est-à-dire définir M_i ensembles flous $F_{x_i}^{k_i}$ pour chaque variable x_i , réalisé en partitionnant chaque univers de discours U_{x_i} en un nombre fini d'intervalles $I_{x_i}^{k_i}$ et en associant à chacun une fonction d'appartenance $\mu_{F_{x_i}^{k_i}}$.

L'univers de discours de la variable x_i doit vérifier la relation suivante :

$$U_{x_i} \subseteq \bigcup_{k_i=1}^{M_i} I_{x_i}^{k_i}; \quad i = 1, \dots, n \quad (\text{IV.30})$$

$I_{x_i}^{k_i} \subset U_{x_i}$; représente le support du k_i ième ensemble flou de la variable linguistique x_i .

$k_i = 1, \dots, M_i$; avec k_i un nombre entier définit l'ensemble flou $F_{x_i}^{k_i}$.

Les ensembles flous $F_{x_i}^{k_i}$; $k_i = 1, \dots, M_i$; forment une partition floue de U_{x_i} pour $i = 1, \dots, n$:

$$P_F(U_{x_i}) = \left\{ F_{x_i}^{k_i} \mid k_i = 1, \dots, M_i \right\} \quad (\text{IV.31})$$

avec la cardinalité $P_F(U_{x_i}) = M_i$ pour $i = 1, \dots, n$.

La largeur de ce support, la cardinalité de la partition floue, ainsi que la forme des fonctions d'appartenance sont choisies par le concepteur.

La variable x_i est alors associée à l'un des M_i ensembles flous $F_{x_i}^{k_i}$ dans l'univers de discours U_{x_i} . Ainsi, le vecteur x peut correspondre à l'une des $\prod_{i=1}^n M_i$ combinaisons possibles du vecteur flou \tilde{x} . D'une façon plus explicite, nous pouvons dire que \tilde{x} représente un des $\prod_{i=1}^n M_i$ vecteurs d'état flous définis par :

$$F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \quad (\text{IV.32})$$

avec : $k_i = 1, \dots, M_i; i = 1, \dots, n$.

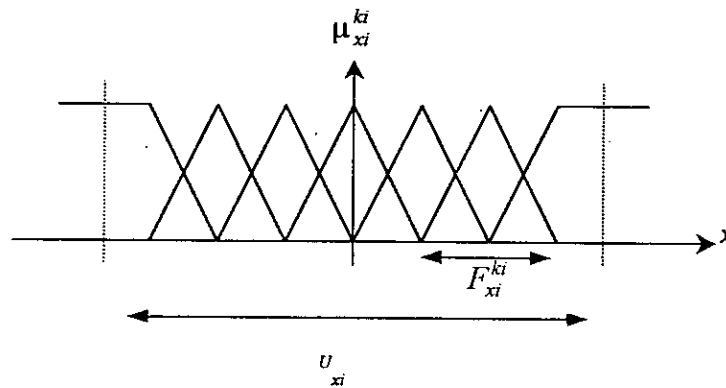


Figure.IV.13 : Répartition des ensembles flous pour une variable d'entrée

Etape 2 : Construction des intervalles des fonctions f et g

Elle concerne la construction du vecteur flou $\tilde{f}(\cdot)$ et la matrice floue $\tilde{g}(\cdot)$. A cet effet, nous commençons tout d'abord par le traitement du vecteur des fonctions $f(\cdot) = [f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot)]^t$. A cet effet, pour chaque combinaison de \tilde{x} nous définissons les intervalles $I_{\tilde{f}}^{(k_1, \dots, k_n)}$ $i = 1, \dots, m$ correspondants à f_i de la manière suivante :

$$I_{\tilde{f}}^{(k_1, \dots, k_n)} = \left\{ f_i(x(t), \eta(t)) / x_1(t) \in I_{x_1}^{k_1}, \dots, x_n(t) \in I_{x_n}^{k_n}, \eta(t) \in P \right\} \quad (\text{IV.33})$$

Ou bien :

$$I_{\tilde{f}}^{(k_1, \dots, k_n)} = \left\{ f_i(x(t), \eta(t)) / x(t) \in I_x^{(k_1, \dots, k_n)}, \eta(t) \in P \right\}$$

avec : $I_x^{(k_1, \dots, k_n)} = I_{x_1}^{k_1} \times I_{x_2}^{k_2} \times \dots \times I_{x_n}^{k_n}$

Ainsi, il est clair que pour un seul élément f_i , nous pouvons avoir jusqu'à $\prod_{i=1}^n M_i$ intervalles $I_{\tilde{f}}^{(k_1, \dots, k_n)}$.

La largeur de chaque intervalle, correspondant à une combinaison possible du vecteur flou \tilde{x} , dépend des largeurs de $I_{x_i}^{k_i}$ et de la plage de variation des paramètres P .

L'univers de discours U_{f_i} (ou la plage de toutes les valeurs possibles) de $f_i(x(t), \eta(t))$ est défini par :

$$U_{f_i} = \bigcup_{(k_1, \dots, k_n)} I_{f_i}^{(k_1, \dots, k_n)} = \bigcup_{k_1=1}^{M_1} \bigcup_{k_2=1}^{M_2} \dots \bigcup_{k_n=1}^{M_n} I_{f_i}^{(k_1, \dots, k_n)} \quad (IV.34)$$

La même procédure que précédemment est utilisée pour déterminer les plages des fonctions scalaires $g_{ij}(\cdot)$, avec $i = 1, \dots, n$; $j = 1, \dots, m$:

$$I_{g_{ij}}^{(k_1, \dots, k_n)} = \left\{ g_{ij}(x(t), \eta(t)) \quad / \quad x(t) \in I_x^{(k_1, \dots, k_n)} \quad , \quad \eta(t) \in P \right\} \quad (IV.35)$$

pour : $k_i = 1, \dots, M_i$; $i = 1, \dots, n$; $j = 1, \dots, m$.

L'univers de discours $U_{g_{ij}}$ de la fonction g_{ij} est défini par l'union des intervalles $I_{g_{ij}}^{(k_1, \dots, k_n)}$ correspondant à toutes les combinaisons possibles (k_1, \dots, k_n) du vecteur \tilde{x} :

$$U_{g_{ij}} = \bigcup_{(k_1, \dots, k_n)} I_{g_{ij}}^{(k_1, \dots, k_n)} = \bigcup_{k_1=1}^{M_1} \bigcup_{k_2=1}^{M_2} \dots \bigcup_{k_n=1}^{M_n} I_{g_{ij}}^{(k_1, \dots, k_n)} \quad (IV.36)$$

Remarque :

Les deux dernières propositions garantissent que les intervalles des fonctions $I_{f_i}^{(k_1, \dots, k_n)}$ et $I_{g_{ij}}^{(k_1, \dots, k_n)}$ sont bornées quel que soit la plage des incertitudes $\eta(t) \in P$ et pour tout $x_i \in I_{x_i}^{(k_1, \dots, k_n)}$, $i = 1, \dots, n$.

Etape 3 : Définition des ensembles flous des sorties

Dans cette étape, à chaque intervalle $I_{f_i}^{(k_1, \dots, k_n)}$ est associée une fonction d'appartenance $\mu_{f_i}^{(k_1, \dots, k_n)}$ dont la forme est choisie par le concepteur afin de bien définir les ensembles flous $F_{f_i}^{(k_1, \dots, k_n)}$ relatifs aux fonctions f_i comme le montre la figure (IV.14).

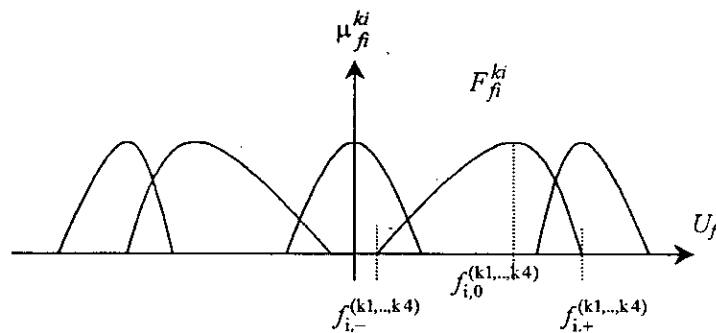


Figure IV.14 : Répartition des ensembles flous pour une fonction de sortie

La partition floue de l'univers de discours U_{f_i} est définie comme:

$$P_F(U_{f_\ell}) = \left\{ F_{f_\ell}^{(k_1, \dots, k_n)} \mid k_\ell = 1, \dots, M_\ell ; \ell = 1, \dots, n \right\} \quad (\text{IV.37})$$

Le vecteur de fonctions floues $\tilde{f}(\cdot)$ est défini comme:

$$\begin{aligned} \tilde{f}(\cdot) : P_F(U_{x_1}) \times \dots \times P_F(U_{x_m}) &\rightarrow [P_F(\mathcal{R})]^n \\ \tilde{x} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n}) &\rightarrow \tilde{f}(\tilde{x}) \end{aligned} \quad (\text{IV.38})$$

avec : $\tilde{f}(\tilde{x}) = (F_{f_1}^{(k_1, \dots, k_n)}, \dots, F_{f_n}^{(k_1, \dots, k_n)})^t$: un vecteur flou

$F_{f_i}^{(k_1, \dots, k_n)} \in P_F(U_{f_i})$: est un ensemble flou

$$k_i = 1, \dots, M_i ; i = 1, \dots, n.$$

De la même manière, nous attribuons à chaque intervalle $I_{g_{ij}}^{(k_1, \dots, k_n)}$ une fonction d'appartenance $\mu_{g_{ij}}^{(k_1, \dots, k_n)}$ afin de construire les ensembles flous $F_{g_{ij}}^{(k_1, \dots, k_n)}$ relatifs aux fonctions g_{ij} . Nous obtenons ainsi, les $(n \times m)$ partitions floues des univers $U_{g_{ij}}$ définis comme suit:

$$P_F(U_{g_{ij}}) = \left\{ F_{g_{ij}}^{(k_1, \dots, k_n)} \mid k_\ell = 1, \dots, M_\ell ; \ell = 1, \dots, n \right\} \quad (\text{IV.39})$$

pour : $i = 1, \dots, n$ et $j = 1, \dots, m$.

Par conséquent, la matrice de fonctions $\tilde{g}(\tilde{x})$ est donnée par:

$$\begin{aligned} \tilde{g}(\cdot) : P_F(U_{x_1}) \times \dots \times P_F(U_{x_m}) &\rightarrow [P_F(\mathcal{R})]^{n \times m} \\ \tilde{x} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n}) &\rightarrow \tilde{g}(\tilde{x}) \end{aligned} \quad (\text{IV.40})$$

$$\text{avec : } \tilde{g}(\tilde{x}) = \left[F_{g_{ij}}^{(k_1, \dots, k_n)} \right]_{\substack{i=1, \dots, n \\ j=1, \dots, m}} = \begin{bmatrix} F_{g_{11}}^{(k_1, \dots, k_n)} & \dots & F_{g_{1m}}^{(k_1, \dots, k_n)} \\ \vdots & & \vdots \\ F_{g_{n1}}^{(k_1, \dots, k_n)} & \dots & F_{g_{nm}}^{(k_1, \dots, k_n)} \end{bmatrix} \text{ une matrice floue}$$

$$F_{g_{ij}}^{(k_1, \dots, k_n)} \in P_F(U_{g_{ij}})$$

$$k_i = 1, \dots, M_i ; i = 1, \dots, n ; j = 1, \dots, m.$$

Etape 4 : Dédution de la base de règles

Les fonctions floues \tilde{f}_i et \tilde{g}_i correspondants au nouveau modèle flou peuvent être concrétisées en utilisant des règles de type *If-Then*. En effet, pour chaque fonction f_i et g_{ij} , sont définies $\prod_{i=1}^n M_i$ règles comme suit :

$$\begin{aligned} \mathfrak{R}_{f_i}^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad x_1 \text{ is } F_{x_1}^{k_1} \text{ AND } \dots \text{ AND } x_n \text{ is } F_{x_n}^{k_n} \\ & \text{THEN} \quad f_i \text{ is } F_{f_i}^{(k_1, \dots, k_n)} \end{aligned} \quad (\text{IV.41})$$

$$\begin{aligned} \text{ou } \mathfrak{R}_{f_i}^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad x \text{ is } F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \\ & \text{THEN} \quad f_i \text{ is } F_{f_i}^{(k_1, \dots, k_n)} \end{aligned} \quad (\text{IV.42})$$

Et de même :

$$\begin{aligned} \mathfrak{R}_{g_{ij}}^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad x_1 \text{ is } F_{x_1}^{k_1} \text{ AND } \dots \text{ AND } x_n \text{ is } F_{x_n}^{k_n} \\ & \text{THEN} \quad g_{ij} \text{ is } F_{g_{ij}}^{(k_1, \dots, k_n)} \end{aligned} \quad (\text{IV.38})$$

D'une manière plus générale, les règles correspondant aux vecteur et matrice de fonctions $f(\cdot)$ et $g(\cdot)$ peuvent être représentées par:

$$\begin{aligned} \mathfrak{R}_f^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad x \text{ is } F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \\ & \text{THEN} \quad f(x, \eta) \text{ is } F_f^{(k_1, \dots, k_n)} = [F_{f_1}^{(k_1, \dots, k_n)}, F_{f_2}^{(k_1, \dots, k_n)}, \dots, F_{f_n}^{(k_1, \dots, k_n)}]^t \end{aligned} \quad (\text{IV.44})$$

avec : $k_i = 1, \dots, M_i$; $i = 1, \dots, n$.

$$\begin{aligned} \mathfrak{R}_g^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad x \text{ is } F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \\ & \text{THEN} \quad g(x, \eta) \text{ is } F_g^{(k_1, \dots, k_n)} = [F_{g_{ij}}^{(k_1, \dots, k_n)}]_{\substack{i=1, \dots, n \\ j=1, \dots, m}} \end{aligned} \quad (\text{IV.45})$$

avec : $k_i = 1, \dots, M_i$; $i = 1, \dots, n$; $j = 1, \dots, m$.

$f(x, \eta)$ et $g(x, \eta)$ sont respectivement le vecteur et la matrice de fonctions caractérisant le système.

IV.6.3. Évaluation de la base de règles du vecteur de commande

En se basant sur le vecteur et la matrice de fonctions floues qui caractérisent le modèle flou du système (Fig. IV.15), nous pouvons générer l'entrée floue \tilde{u} et l'écrire sous forme d'un vecteur de fonctions floues comme suit:

$$\begin{aligned} \tilde{u}(\cdot) : P_F(U_{x1}) \times \dots \times P_F(U_{x_n}) &\rightarrow [P_F(\mathfrak{R})]^m \\ \tilde{x} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n}) &\rightarrow \tilde{u}(\tilde{x}) \end{aligned} \quad (IV.46)$$

avec : $\tilde{u}(\tilde{x}) = (F_{u_1}^{(k_1, \dots, k_n)}, \dots, F_{u_n}^{(k_1, \dots, k_n)})^t$ vecteur flou de commande ;

$$k_i = 1, \dots, M_i; i = 1, \dots, n.$$

La base de règles du vecteur d'entrée de commande u sera alors donnée par les $\prod_{i=1}^n M_i$ règles suivantes :

$$\begin{aligned} \mathfrak{R}_u^{(k_1, \dots, k_n)} : \quad \text{IF } x \text{ is } F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \\ \text{THEN } u \text{ is } F_u^{(k_1, \dots, k_n)} = [F_{u_1}^{(k_1, \dots, k_n)}, F_{u_2}^{(k_1, \dots, k_n)}, \dots, F_{u_n}^{(k_1, \dots, k_n)}]^t \end{aligned} \quad (IV.47)$$

avec : $k_i = 1, \dots, M_i; i = 1, \dots, n.$

IV.6.4. Obtention du modèle flou

En se basant sur les équations (IV.27), (IV.44), (IV.45) et (IV.47), le modèle flou peut être représenté par les $\prod_{i=1}^n M_i$ règles suivantes :

$$\begin{aligned} \mathfrak{R}_{\dot{x}}^{(k_1, \dots, k_n)} : \quad \text{IF } x \text{ is } F_x^{(k_1, \dots, k_n)} = [F_{x_1}^{k_1}, F_{x_2}^{k_2}, \dots, F_{x_n}^{k_n}]^t \\ \text{THEN } \dot{x} \text{ is } F_{\dot{x}}^{(k_1, \dots, k_n)} = [F_{\dot{x}_1}^{(k_1, \dots, k_n)}, F_{\dot{x}_2}^{(k_1, \dots, k_n)}, \dots, F_{\dot{x}_n}^{(k_1, \dots, k_n)}]^t \end{aligned} \quad (IV.48)$$

avec :

$$F_{\dot{x}_i}^{(k_1, \dots, k_n)} = F_{f_i}^{(k_1, \dots, k_n)} \tilde{\vdash} \sum_{j=1}^m F_{g_j}^{(k_1, \dots, k_n)} \tilde{\sim} F_{u_j}^{(k_1, \dots, k_n)} \quad (IV.49)$$

où : \sum opérateur de sommation étendue.

Il est clair que le modèle déduit à partir des quatre étapes précédemment définies est élaboré en "off-line". Finalement, nous pouvons dire que, la méthode d'élaboration de modèles flous proposée exploite

la connaissance analytique du modèle du système. Elle présente l'avantage d'être plus systématique, et valable pour tout l'univers de discours considéré de l'espace d'entrée et que la prise en compte de la plage des incertitudes permet d'assurer une meilleure robustesse lors de l'élaboration d'une commande floue robuste qui fera l'objet du paragraphe suivant.

IV.7. COMMANDE A STRUCTURE VARIABLE FLOUE BASEE SUR LE MODELE FLOUE DE BEN-GHALIA

Un contrôleur flou est un système d'inférence flou ayant pour sortie des actions de commande, alors que ses entrées sont le plus fréquemment considérées comme étant l'erreur et la variation de l'erreur entre une consigne et une variable du processus à contrôler. Cette dernière représente en général la (ou les) grandeurs à réguler et d'autres grandeurs mesurées qui traduisent l'évolution du système.

Il existe plusieurs types de contrôleurs flous qui diffèrent les uns des autres par la nature des conséquences des règles comme il a été spécifié précédemment. L'élaboration de ces contrôleurs repose généralement sur les connaissances de l'expert ou sur des essais expérimentaux, sans avoir recours aux équations représentatives (ou au modèle) du système. Ce qui se traduit par une certaine difficulté à prévoir le comportement dynamique du processus de commande, en particulier du point de vue de la stabilité et de la robustesse. Dans ce sens, il est préférable, en cas de présence du modèle, de concevoir le contrôleur flou, en se basant directement soit sur le modèle du système à contrôler, soit sur des connaissances préalables du système permettant d'élaborer un modèle flou qui l'approxime.

Typiquement, dans le cas d'une commande floue basée sur un modèle flou, on peut distinguer quatre blocs principaux à savoir (Fig IV.15) [GHA96c]:

- un modèle flou,
- un contrôleur flou,
- le système à contrôler,
- un bloc de défuzzification.

Le modèle flou décrit grossièrement le comportement du système à contrôler et est représenté par l'une des quatre représentations floues que nous avons présentées précédemment (§IV.4).

L'entrée floue de commande \tilde{u} est générée par le contrôleur flou. La phase de défuzzification permet de convertir celle-ci en une entrée réelle de commande. La méthode de défuzzification doit vérifier la stabilité asymptotique et la robustesse de la boucle de retour pour toute la plage d'incertitudes [GHA96c].

Le bloc du modèle flou, dans ce cas, fournit une base de règles et des ensembles flous nécessaires pour la conception du contrôleur flou. En pratique, ce bloc est fictif, et une seule base de règles est utilisée. Tous les calculs dans l'élaboration de la commande floue se font en "off-line".

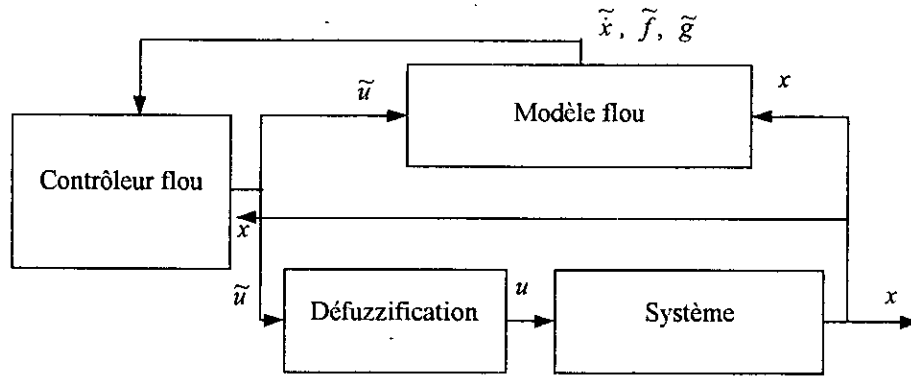


Figure. IV.15 : Structure de base d'un système de commande basé sur un modèle flou

Plusieurs stratégies de commande ont échoué dans le réglage de certains systèmes, pour la simple raison qu'elles nécessitent des connaissances sur les incertitudes telles que leurs moyennes, leurs variances ou leurs types (bruit blanc, ayant une densité gaussienne, ... , etc).

La technique de modélisation floue proposée par Ben-Ghalia adoptée, permet de concevoir des contrôleurs flous robustes pour les systèmes incertains, sans avoir à connaître ni la structure, ni la forme ou nature des incertitudes.

Dans le présent travail, nous essayons de tirer profit des avantages de cette modélisation floue et de l'introduire dans la commande du moteur asynchrone afin d'obtenir un contrôle flou robuste basé sur le modèle flou de notre système. L'approche proposée pour la conception d'un contrôleur flou est une extension des techniques de commande à structures variables pour les systèmes flous représentés par des équations différentielles floues.

Dans l'approche proposée, le contrôleur flou illustré à la figure (IV.15) évalue l'entrée de commande floue \tilde{u} qui sera défuzzifiée pour générer l'entrée réelle u .

IV.7.1. Algorithme de la commande à structure variable floue adopté

Soit le système décrit par l'équation (IV.27) et par sa représentation floue suivante :

$$\tilde{\dot{x}} = \tilde{f}(\tilde{x}) + \tilde{g}(\tilde{x}) \tilde{u} \quad (\text{IV.50})$$

avec :

$$\tilde{f}(x) = \begin{pmatrix} \tilde{f}_1(\tilde{x}_1, \dots, \tilde{x}_n) \\ \dots \\ \tilde{f}_1(\tilde{x}_1, \dots, \tilde{x}_n) \end{pmatrix} ; \quad \tilde{g}(\tilde{x}) = [g_{ij}(\tilde{x}_1, \dots, \tilde{x}_n)]_{\substack{i=1, \dots, n \\ j=1, \dots, m}} ; \quad \tilde{u}(t) = \begin{pmatrix} \tilde{u}_1(t) \\ \dots \\ \tilde{u}_m(t) \end{pmatrix}$$

Avant de présenter l'algorithme de la commande à structure variable floue, nous jugeons utile de rappeler quelques notations et définitions utilisées par ce dernier.

Soit $U_x = U_{x_1} \times \dots \times U_{x_n}$ et $U_u = U_{u_1} \times \dots \times U_{u_m}$: les univers de discours (ou l'espace) des vecteurs d'entrée et de commande respectivement.

L'univers de discours du vecteur d'erreur e est défini par :

$$U_e = \left\{ e \in \mathbb{R}^n \mid e = x_d - x ; x \in U_x \right\} \quad (\text{IV.51})$$

Le vecteur d'erreur flou, $\tilde{e}(t) \in P_F(U_e)$, correspondant au vecteur $e(t)$ est donné par :

$$\tilde{e}(t) = \tilde{x}_d(t) \simeq \tilde{x}(t) = \begin{pmatrix} \tilde{e}_1(t) \\ \dots \\ \tilde{e}_n(t) \end{pmatrix} = \begin{pmatrix} \tilde{x}_{d1}(t) - \tilde{x}_1(t) \\ \dots \\ \tilde{x}_{dn}(t) - \tilde{x}_n(t) \end{pmatrix} \quad (\text{IV.52})$$

avec : $\tilde{e}_i(t) = \tilde{x}_{di}(t) \simeq \tilde{x}_i(t)$ pour $i = 1, \dots, n$;

et $\tilde{x}_{di}(t)$ un singleton flou centré sur la $i^{\text{ème}}$ référence x_{di} .

L'univers de discours du vecteur de commutation $s(x)$ défini par l'équation (II.17) est décrit par :

$$U_s = U_{s1} \times U_{s2} \times \dots \times U_{sm} \subset \mathbb{R}^m \quad (\text{IV.53})$$

$$U_{si} = \{ s_i(x) \in \mathbb{R} \mid e \in U_e \} ; i = 1, \dots, m$$

U_{si} est l'ensemble de toutes les variables possibles de la fonction de commutation s_i pour tout $e \in U_e$.

Définition IV.1 : Le vecteur flou de commutation \tilde{s} est un vecteur de m surfaces, défini par :

$$\begin{aligned} \tilde{s} : P_F(U_e) &\rightarrow P_F(U_s) \\ \tilde{e} &\rightarrow \tilde{s}(\tilde{e}) \end{aligned} \quad (\text{IV.54})$$

avec : $\tilde{s}(\tilde{e})$ l'extension de la relation définissant la surface en fonction de l'erreur donnée précédemment:

$$\tilde{s}(\tilde{e}) = \begin{pmatrix} \tilde{s}_1(\tilde{e}) \\ \dots \\ \tilde{s}_m(\tilde{e}) \end{pmatrix} \quad (\text{IV.55})$$

$\tilde{s}_i(\tilde{e}) \in P_F(U_{si})$ la fonction floue de commutation définie dans U_{si} et qui correspond au vecteur d'erreur \tilde{e} défini dans U_e .

Définition IV.2 : La surface floue de glissement est définie par :

$$\begin{aligned} \tilde{\mathcal{S}}_s &= \tilde{\mathcal{S}}_{s_1} \times \dots \times \tilde{\mathcal{S}}_{s_m} \\ \tilde{\mathcal{S}}_{s_i} &= \left\{ \tilde{e} \in P_F(U_e) / \tilde{s}_i(\tilde{e}) = \tilde{0} \right\} \end{aligned} \quad (\text{IV.56})$$

avec : $\tilde{0}$ un ensemble flou constant entourant la valeur 0.

L'étendue de la dynamique désirée \tilde{s}_d peut être définie par un vecteur flou comme suit:

$$\tilde{\mathbf{s}}(\tilde{e}) = \begin{pmatrix} \tilde{s}_1(\tilde{e}) \\ \dots \\ \tilde{s}_m(\tilde{e}) \end{pmatrix} \in P_F(U_s) \quad (\text{IV.57})$$

avec : $P_F(U_s) = P_F(U_{s_1}) \times \dots \times P_F(U_{s_m})$

Définition IV.3 : La condition d'atteignabilité est régie par :

$$\exists \tilde{N} \text{ tel que } \tilde{\mathbf{s}}(\tilde{e})' \cdot \tilde{\mathbf{s}}(\tilde{e}) \subset \tilde{N} \quad (\text{IV.58})$$

\tilde{N} est un ensemble flou dont le support est strictement négatif.

Il est important de noter que d'après l'approche adoptée, la condition d'atteignabilité définie par l'équation (IV.58) est équivalente à la condition suivante :

$$\forall s_i(e) \in \text{Supp}(\tilde{s}_i(\tilde{e})) - \{0\} \text{ et } \forall \dot{s}_i(e) \in \text{Supp}(\tilde{\dot{s}}_i(\tilde{e})) - \{0\} \quad (\text{IV.59})$$

Alors : $s_i(e) \cdot \dot{s}_i(e) < 0$

L'algorithme de l'élaboration de la commande à structure variable floue, basée sur le modèle flou linguistique présenté ultérieurement, peut être réalisée en cinq étapes:

Étape 1. Construction du vecteur flou d'erreur $F_e^{(k_1, \dots, k_n)}$

Soit $P_F(U_x)$ et $P_F(U_e)$, les partitions floues des univers de discours U_x et U_e respectivement, définies comme :

$$\begin{aligned} P_F(U_x) &= P_F(U_{x_1}) \times \dots \times P_F(U_{x_n}) \\ P_F(U_e) &= P_F(U_{e_1}) \times \dots \times P_F(U_{e_n}) \end{aligned} \quad (\text{IV.60})$$

avec : $P_F(U_{e_i}) = \left\{ F_{e_i}^{k_i} \text{ défini dans } U_{e_i} \text{ de } \mathbb{R} ; k_i = 1, \dots, M_i ; i = 1, \dots, n \right\}$

$F_{e_i}^{k_i} = \tilde{x}_{di} \simeq F_{x_i}^{k_i}$ des ensembles flous que peut prendre la variable floue \tilde{e}_i .

Le vecteur d'erreur $e \in \mathbb{R}^n$ est caractérisé par les $\prod_{i=1}^n M_i$ vecteurs flous suivants:

$$F_e^{(k_1, \dots, k_n)} = \begin{pmatrix} F_{e_1}^{k_1} \\ \dots \\ F_{e_n}^{k_n} \end{pmatrix} ; k_i = 1, \dots, M_i ; i = 1, \dots, n \quad (\text{IV.61})$$

Cependant le vecteur flou \tilde{e} prend un des $\prod_{i=1}^n M_i$ vecteurs flous $F_e^{(k_1, \dots, k_n)}$.

Étape 2. Construction du vecteur flou de commutation $F_s^{(k_1, \dots, k_n)}$

Soit $P_F(U_s)$ la partition floue de l'univers de discours U_s :

$$P_F(U_s) = P_F(U_{s_1}) \times \dots \times P_F(U_{s_n}) \quad (\text{IV.62})$$

avec $P_F(U_s) = \left\{ F_s^{(k_1, \dots, k_n)} \text{ défini dans } U_s \text{ de } \mathbb{R}^m ; k_i = 1, \dots, M_i ; i = 1, \dots, n \right\}$

et $P_F(U_{s_j}) = \left\{ F_{s_j}^{(k_1, \dots, k_n)} \text{ défini dans } U_{s_j} \text{ de } \mathbb{R} ; k_j = 1, \dots, M_j ; j = 1, \dots, n \right\}$

Le vecteur flou $F_s^{(k_1, \dots, k_n)}$ est constitué par m ensembles flous $F_{s_i}^{(k_1, \dots, k_n)}$ définissant les fonctions floues de commutation $\tilde{s}_i(\tilde{e})$, tel que :

$$F_s^{(k_1, \dots, k_n)} = \begin{pmatrix} F_{s_1}^{(k_1, \dots, k_n)} \\ \dots \\ F_{s_m}^{(k_1, \dots, k_n)} \end{pmatrix} \quad (\text{IV.63})$$

avec : $F_{s_i}^{(k_1, \dots, k_n)} = \left\{ s_i(e) \in U_{s_i} \mid e \in U_e \right\}$

La base de règles correspondante à s_i est présentée par les $\prod_{i=1}^n M_i$ règles de type *If-Then* donné par:

$$\begin{aligned} \mathfrak{R}_{s_i}^{(k_1, \dots, k_n)} : \quad & \text{IF} \quad e = (e_1, \dots, e_n)^t \quad \text{is} \quad F_e^{(k_1, \dots, k_n)} = (F_{e_1}^{k_1}, \dots, F_{e_n}^{k_n})^t \\ & \text{THEN} \quad s_i(e) \quad \text{is} \quad F_{s_i}^{(k_1, \dots, k_n)} \end{aligned} \quad (\text{IV.64})$$

Étape 3. Choix de la dynamique floue désirée $F_s^{(k_1, \dots, k_n)}$

Les fonctions \tilde{s}_i , correspondants à la dynamique désirée du système sont fonction des signes de s_i , et doivent par conséquent, vérifier la condition d'atteignabilité régie par l'équation (II.5):

Dans le cas d'une commande à structure variable floue, le problème se pose sur l'ensemble flou $F_{s_i}^{(k_1, \dots, k_n)}$, ($i = 1, \dots, n$) dont le support contient une partie positive, et une autre négative de l'univers de discours U_{s_i} (Fig. IV.16). Dans ce cas, l'ensemble flou $F_{s_i}^{(k_1, \dots, k_n)}$ est décomposé en deux parties : $PF_{s_i}^{(k_1, \dots, k_n)}$ et $NF_{s_i}^{(k_1, \dots, k_n)}$ qui sont des ensembles flous positifs et négatifs respectivement. L'ensemble $F_{s_i}^{(k_1, \dots, k_n)}$ vérifie les conditions suivantes :

$$\begin{aligned} \text{Supp}(U_{s_i}^-) \cap \text{supp}(NF_{s_i}^{(k_1, \dots, k_n)}) &= \emptyset \\ \text{Supp}(PF_{s_i}^{(k_1, \dots, k_n)}) &= \text{supp}(NF_{s_i}^{(k_1, \dots, k_n)}) \cup \text{supp}(NF_{s_i}^{(k_1, \dots, k_n)}) \cup \{0\} \\ \mu(PF_{s_i}^{(k_1, \dots, k_n)}) &= \mu(F_{s_i}^{(k_1, \dots, k_n)}) \quad \text{si } s > 0 \quad \text{et } 0 \text{ ailleurs} \\ \mu(NF_{s_i}^{(k_1, \dots, k_n)}) &= \mu(F_{s_i}^{(k_1, \dots, k_n)}) \quad \text{si } s < 0 \quad \text{et } 0 \text{ ailleurs} \end{aligned} \quad (\text{IV.65})$$

où $\text{supp}(F)$ représente le support de l'ensemble flou F .

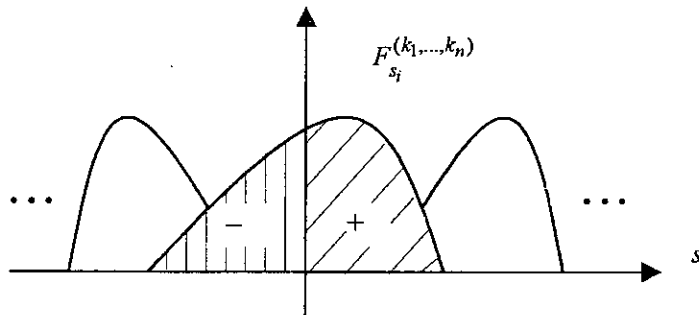


Figure. IV.16 : Décomposition des ensembles flous en des ensembles flous positifs et négatifs.

Nous définissons le vecteur flou \tilde{s}_d caractérisé par l'un des $\prod_{i=1}^n M_i$ vecteurs d'ensembles flous $F_{s_i}^{(k_1, \dots, k_n)}$ tel que :

$$F_{\tilde{s}_d}^{(k_1, \dots, k_n)} = \begin{pmatrix} F_{s_1}^{(k_1, \dots, k_n)} \\ \dots \\ F_{s_m}^{(k_1, \dots, k_n)} \end{pmatrix} \quad (\text{IV.66})$$

Le vecteur flou \tilde{s}_d définit grossièrement le comportement du système en dehors de la surface de glissement.

Chaque terme $F_{s_i}^{(k_1, \dots, k_n)}$ doit commuter entre deux ensembles flous suivant le signe de la fonction de commutation s , et doit en plus garantir la condition floue d'atteignabilité définie par la relation (IV.59). Ce qui conduit à :

$$\begin{aligned}
F_{s_i}^{(k_1, \dots, k_n)} &= -\tilde{M}(F_{s_i}^{(k_1, \dots, k_n)}) \quad \text{si } \text{supp}(F_{s_i}^{(k_1, \dots, k_n)}) \subset U_{s_i}^+ \\
&= +\tilde{M}(F_{s_i}^{(k_1, \dots, k_n)}) \quad \text{si } \text{supp}(F_{s_i}^{(k_1, \dots, k_n)}) \subset U_{s_i}^-
\end{aligned} \tag{IV.67}$$

et par :

$$\begin{aligned}
F_{s_i}^{(k_1, \dots, k_n)} &= -\omega_s \sim \tilde{M}(PF_{s_i}^{(k_1, \dots, k_n)}) \tilde{\mp} (1 - \omega_s) \sim \tilde{M}(NF_{s_i}^{(k_1, \dots, k_n)}) \quad \text{si} \\
\text{supp}(F_{s_i}^{(k_1, \dots, k_n)}) &\text{ contient en même temps une partie négative et une autre positive de } U_{s_i}
\end{aligned} \tag{IV.68}$$

avec :

$$\begin{aligned}
\omega_s &= 1 \quad \text{si } s_i \in \text{supp}(F_{s_i}^{(k_1, \dots, k_n)}) \cap U_{s_i}^+ \\
&= 0 \quad \text{si } s_i \in \text{supp}(F_{s_i}^{(k_1, \dots, k_n)}) \cap U_{s_i}^-
\end{aligned} \tag{IV.69}$$

$U_{s_i}^+ \subset \mathbb{R}^+$ et $U_{s_i}^- \subset \mathbb{R}^-$ sont les parties strictement positives et strictement négatives correspondants respectivement à : $U_{s_i} \subset \mathbb{R}$.

Ainsi, la base de règles qui caractérise la dynamique floue désirée durant la phase transitoire d'atteignabilité, correspond à $\prod_{i=1}^n M_i$ règles suivantes :

$$\begin{aligned}
\mathfrak{R}_{s_i}^{(k_1, \dots, k_n)} : \text{IF } e = (e_1, \dots, e_n)^t \quad \text{is } F_e^{(k_1, \dots, k_n)} = (F_{e_1}^{k_1}, \dots, F_{e_n}^{k_n})^t \\
\text{THEN } \hat{s}_i(e) \quad \text{is } -\omega_s \sim \tilde{M}(PF_{s_i}^{(k_1, \dots, k_n)}) \tilde{\mp} (1 - \omega_s) \sim \tilde{M}(NF_{s_i}^{(k_1, \dots, k_n)})
\end{aligned} \tag{IV.70}$$

$$i = 1, \dots, m; \quad k_j = 1, \dots, M_j; \quad j = 1, \dots, n$$

Étape 4. Calcul de la loi de commande floue $F_u^{(k_1, \dots, k_n)}$

Les fonctions de commutation doivent avoir un degré relatif égal à 1. Cela permet l'extraction de la loi de commande à partir de la dérivée \hat{s} .

L'extension de l'équation (II.13) permet l'obtention de la loi de commande floue qui sera définie par :

$$\tilde{u} = \tilde{m}v \left\{ (\tilde{s}_x(e)) \tilde{\sim} \tilde{g}(\tilde{x}) \right\} \tilde{\sim} \left\{ \tilde{s}_d(e) = (\tilde{s}_x)^t \tilde{\sim} \tilde{f}(\tilde{x}) = \tilde{s}_t \right\} \tag{IV.71}$$

avec :

$$\tilde{s}_x \text{ et } \tilde{s}_t \text{ des vecteurs flous obtenus par l'extension des vecteurs } \frac{\partial s}{\partial x} \text{ et } \frac{\partial s}{\partial t} \text{ respectivement.}$$

$\tilde{f}(\tilde{x})$ et $\tilde{g}(\tilde{x})$ sont respectivement le vecteur et la matrice flous caractérisant le modèle flou du système.

$\tilde{u} = (\tilde{u}_1, \dots, \tilde{u}_m)^t$ le vecteur flou de commande qui est l'un des vecteurs d'ensembles flous :

$$F_u^{(k_1, \dots, k_n)} = \begin{pmatrix} F_{u_1}^{(k_1, \dots, k_n)} \\ \dots \\ F_{u_m}^{(k_1, \dots, k_n)} \end{pmatrix} \quad (\text{IV.72})$$

La base de règles correspondant à la loi de commande est de ce fait définie comme:

$\mathfrak{R}_{u_i}^{(k_1, \dots, k_n)}$:

$$\text{IF } x = (x_1, \dots, x_n)^t \text{ is } F_x^{(k_1, \dots, k_n)} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n})^t \text{ AND } s_i \text{ is } F_{s_i}^{(k_1, \dots, k_n)} \quad (\text{IV.73})$$

$$\text{THEN } u_i \text{ is } F_{u_i}^{(k_1, \dots, k_n)}$$

$$i = 1, \dots, m ; k_j = 1, \dots, M_j ; j = 1, \dots, n$$

Étape 5. Défuzzification de la loi de commande floue

Chaque règle génère les ensembles flous $F_{u_i}^{(k_1, \dots, k_n)}$, qui doivent être défuzzifiés afin de fournir une valeur réelle pour chaque sortie de commande. La défuzzification peut être obtenue par la méthode de la moyenne pondérée définie par :

$$u_i = \frac{\sum_{k_1} \dots \sum_{k_n} \bar{u}_i^{(k_1, \dots, k_n)} \cdot T(\mu_{x_1}^{k_1}, \dots, \mu_{x_n}^{k_n})}{\sum_{k_1} \dots \sum_{k_n} T(\mu_{x_1}^{k_1}, \dots, \mu_{x_n}^{k_n})} \quad (\text{IV.74})$$

u_i : représente l'une des variables de sortie, $i = 1, \dots, m$.

$\bar{u}_i^{(k_1, \dots, k_n)}$: est la valeur représentative de l'ensemble flou $F_{u_i}^{(k_1, \dots, k_n)}$ qui est la sortie de la règle $R_{u_i}^{(k_1, \dots, k_n)}$; $i = 1, \dots, m$.

T : opération de T-norme.

IV.7.2. Cas des systèmes mis sous forme canonique

Dans le cas où chaque sous-système est caractérisé par une équation différentielle de la forme:

$$\dot{z} = f(x) + g(x) \cdot u \quad (\text{IV.75})$$

avec :

$$z = \begin{pmatrix} x_1^j \\ \vdots \\ x_r^j \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_r \end{pmatrix} ; \quad f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_{r-1}(x) \\ f_r(x) \end{pmatrix} = \begin{pmatrix} z_2 \\ \vdots \\ z_r \\ a(x) \end{pmatrix} ; \quad g(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_{r-1}(x) \\ g_r(x) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b(x) \end{pmatrix}$$

$$x \in \mathbb{R}^n ; z \in \mathbb{R}^r$$

n, r : les dimensions des sous-systèmes et du système global, respectivement.

$a(\cdot), b(\cdot) : \mathbb{R}^n \times P \rightarrow \mathbb{R}$ sont des fonctions scalaires caractérisant le modèle du sous-système.

IV.7.3. Modélisation floue du sous-système

Pour l'élaboration de la commande à structure variable floue basée sur un modèle flou, nous devons procéder de la manière suivante.

$$\text{Soit : } z_d = \begin{pmatrix} z_{d_1}(t) \\ z_{d_2}(t) \\ \vdots \\ z_{d_r}(t) \end{pmatrix} = \begin{pmatrix} z_{d_1} \\ \dot{z}_{d_1} \\ \vdots \\ z_{d_1}^{(r-1)} \end{pmatrix} \text{ le vecteur de référence (ou la trajectoire désirée) du sous-système.}$$

U_u : l'univers de discours de l'entrée de commande du sous-système.

$\eta \in P$: représente les incertitudes inhérentes dans le modèle.

Le problème est de trouver la commande u (de chaque sous-système) de telle sorte que, pour tout incertitude $\eta \in P$, le système global soit stable et converge asymptotiquement vers les vecteurs de référence, tout en supposant que les dynamiques internes sont stables (au moins bornées).

Chaque sous-système est décrit par la nouvelle représentation floue définie par une équation différentielle floue donnée par :

$$\tilde{\dot{z}} = \tilde{f}(\tilde{x}) \tilde{+} \tilde{g}(\tilde{x}) \tilde{\approx} \tilde{u} \quad (\text{IV.76})$$

avec :

$$\tilde{z} = \begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_r \end{pmatrix} ; \quad \tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{pmatrix} \text{ les vecteurs d'état flous ; } r \leq n$$

$$\tilde{f}(\tilde{x}) = \begin{pmatrix} \tilde{f}_1(\tilde{x}) \\ \vdots \\ \tilde{f}_{r-1}(\tilde{x}) \\ \tilde{f}_r(\tilde{x}) \end{pmatrix} = \begin{pmatrix} \tilde{z}_2 \\ \vdots \\ \tilde{z}_r \\ \tilde{a}(\tilde{x}) \end{pmatrix} ; \quad \tilde{g}(\tilde{x}) = \begin{pmatrix} \tilde{g}_1(\tilde{x}) \\ \vdots \\ \tilde{g}_{r-1}(\tilde{x}) \\ \tilde{g}_r(\tilde{x}) \end{pmatrix} = \begin{pmatrix} \tilde{0} \\ \vdots \\ \tilde{0} \\ \tilde{b}(\tilde{x}) \end{pmatrix}$$

$\tilde{0}$: est un singleton flou centré sur la valeur « zéro ».

La modélisation floue du présent modèle suit les mêmes étapes que celles de la section (§IV.5).

Après avoir suivie les étapes de l'algorithme de Ben-Ghalia, nous obtenons les ensembles flous $F_{f_i}^{(k_1, \dots, k_n)}$, $F_{g_i}^{(k_1, \dots, k_n)}$ ($k_j = 1, \dots, M_j$; $j = 1, \dots, n$; $i = 1, \dots, r$) correspondant aux fonctions $f_i(\cdot)$ et $g_i(\cdot)$, et qui dépendent de la combinaison du vecteur d'état flou \tilde{x} et de la plage des incertitudes P .

Dans ce cas, les fonctions $f_i(\cdot)$, $i = 1, \dots, (r-1)$ sont mono-variables et les ensembles flous correspondants sont plus faciles à obtenir et prennent chacun comme valeur l'un des éléments du vecteur flou \tilde{x} .

L'extension des fonctions restantes $a(\cdot)$ et $b(\cdot)$ mène à deux applications floues $\tilde{a}(\cdot)$ et $\tilde{b}(\cdot)$ données par :

$$\begin{aligned} \tilde{a}(\cdot) : P_F(U_{x1}) \times \dots \times P_F(U_{x_n}) &\rightarrow P_F(\mathbb{R}) \\ \tilde{x} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n}) &\rightarrow \tilde{a}(\tilde{x}) = F_a^{(k_1, \dots, k_n)} \end{aligned} \quad (IV.77)$$

$$\begin{aligned} \tilde{b}(\cdot) : P_F(U_{x1}) \times \dots \times P_F(U_{x_n}) &\rightarrow P_F(\mathbb{R}) \\ \tilde{x} = (F_{x_1}^{k_1}, \dots, F_{x_n}^{k_n}) &\rightarrow \tilde{b}(\tilde{x}) = F_b^{(k_1, \dots, k_n)} \end{aligned} \quad (IV.78)$$

La base de règles correspondante se caractérise par $\prod_{i=1}^n M_i$ règles au maximum pour chaque fonction.

IV.7.4. Élaboration de la Commande à structure variable floue

L'objectif que nous nous fixons consiste à trouver une commande floue robuste pour chaque sortie et ce, en se basant sur le modèle flou du sous-système résultant. A cet effet, nous procédons de la manière suivante:

Étape 1. Construction du vecteur d'erreur

De la manière que précédemment, le vecteur d'erreur est défini par :

Étape 3. Choix de la dynamique floue désirée $F_i^{(k_1, \dots, k_r)}$

Pendant le mode d'atteignabilité, la robustesse est assurée par le choix de la dynamique floue désirée \tilde{s}_d , qui dépend de la fonction floue de commutation $\tilde{s}(\tilde{e})$.

Les ensembles flous $F_i^{(k_1, \dots, k_r)}$ sont choisis en se référant à l'extension d'une fonction linéaire ou non-linéaire $M(s)$.

$$F_i^{(k_1, \dots, k_r)} = -\omega_i \tilde{\cdot} F_{M(PF_i)}^{(k_1, \dots, k_r)} \tilde{\cdot} (\omega_i - 1) \tilde{\cdot} F_{M(NF_i)}^{(k_1, \dots, k_r)} \quad (\text{IV.82})$$

avec :

$$\begin{aligned} \omega_i &= 1 & \text{si } s \in \text{supp}(F_i^{(k_1, \dots, k_r)}) \cap U_{s_i}^+ \\ \omega_i &= 0 & \text{si } s \in \text{supp}(F_i^{(k_1, \dots, k_r)}) \cap U_{s_i}^- \end{aligned} \quad (\text{IV.83})$$

$U_{s_i}^+ \subset \mathbb{R}$ et $U_{s_i}^- \subset \mathbb{R}$ sont les parties strictement positive et strictement négative de $U_{s_i} \subset \mathbb{R}$.

$F_{M(PF_i)}^{(k_1, \dots, k_r)}$ sont les ensembles flous que peut prendre $\tilde{M}(\tilde{s})$ lorsque $\tilde{s} = PF_i^{(k_1, \dots, k_r)}$.

$F_{M(NF_i)}^{(k_1, \dots, k_r)}$ sont les ensembles flous que peut prendre $\tilde{M}(\tilde{s})$ lorsque $\tilde{s} = NF_i^{(k_1, \dots, k_r)}$.

Étape 4. Calcul de la loi de commande $F_u^{(k_1, \dots, k_r)}$

La fonction floue correspondant à $\hat{s}(e)$ est obtenue à partir des équations (II.16) :

$$\tilde{s}(\tilde{e}) = \lambda_1 \tilde{\cdot} \tilde{e}_2 \tilde{\cdot} \lambda_2 \tilde{\cdot} \tilde{e}_3 \tilde{\cdot} \dots \tilde{\cdot} \lambda_{r-1} \tilde{\cdot} \tilde{e}_r \tilde{\cdot} \lambda_r \tilde{\cdot} \tilde{e}_r \quad (\text{IV.84})$$

avec :

$$\tilde{e}_r = \tilde{z}_d \tilde{\cdot} \tilde{z}_r \quad (\text{IV.85})$$

Lorsque le sous-système atteint une dynamique floue désirée $\tilde{s}_d(\tilde{e})$, choisie dans l'étape précédente, il sera forcé de suivre le modèle flou suivant :

$$\tilde{z}_r = \tilde{z}_d \tilde{\cdot} \frac{1}{\lambda_n} \tilde{\cdot} \tilde{s}_d(\tilde{e}) \tilde{\cdot} \frac{\lambda_1}{\lambda_r} \tilde{\cdot} \tilde{e}_2 \tilde{\cdot} \frac{\lambda_2}{\lambda_r} \tilde{\cdot} \tilde{e}_3 \tilde{\cdot} \dots \tilde{\cdot} \frac{\lambda_{r-1}}{\lambda_r} \tilde{\cdot} \tilde{e}_r \quad (\text{IV.86})$$

D'autre part, à partir de l'équation (IV.76), la dynamique floue \tilde{z}_r du sous-système est décrite par :

$$\tilde{z}_r = \tilde{a}(\tilde{x}) \tilde{\cdot} \tilde{b}(\tilde{x}) \tilde{\cdot} \tilde{u} \quad (\text{IV.87})$$

D'après les deux dernières équations, la loi de commande floue qui en résulte est donnée par :

$$\tilde{u} = \left(-\tilde{a}(\tilde{x}) \tilde{z}_{dr} \tilde{z} + \frac{1}{\lambda_n} \tilde{s}_d(\tilde{e}) \tilde{z} + \frac{\lambda_1}{\lambda_r} \tilde{e}_2 \approx \dots \approx \frac{\lambda_{r-1}}{\lambda_r} \tilde{e}_r \right) \tilde{z} \tilde{b}(\tilde{x}) \quad (\text{IV.88})$$

Cette équation nous permet de déduire les ensembles flous $F_u^{(k_1, \dots, k_r)}$, tels que :

$$F_u^{(k_1, \dots, k_r)} = \left(-F_a^{(k_1, \dots, k_n)} \tilde{z}_{dr} \tilde{z} + \frac{1}{\lambda_n} F_{i_d}^{(k_1, \dots, k_r)} \tilde{z} + \frac{\lambda_1}{\lambda_r} F_{e_2}^{k_{e_2}} \approx \dots \approx \frac{\lambda_{r-1}}{\lambda_r} F_{e_r}^{k_{e_r}} \right) \tilde{z} F_b^{(k_1, \dots, k_n)} \quad (\text{IV.89})$$

et la base de règles correspondant à l'entrée de commande u pour chaque sous-système :

$$\begin{aligned} \mathfrak{R}_u^{(k_1, \dots, k_r)} : \quad & \text{IF} \quad z = (z_1, \dots, z_r)^t \quad \text{is} \quad F_z^{(k_1, \dots, k_r)} = (F_{z_1}^{k_1}, \dots, F_{z_r}^{k_r})^t \\ & \text{THEN} \quad u \quad \text{is} \quad F_u^{(k_1, \dots, k_r)} \end{aligned} \quad (\text{IV.90})$$

IV.7.5. Application à la commande du moteur asynchrone

Pour une commande en tension de la machine asynchrone à cage, le modèle réduit correspondant dans le repère lié au stator est obtenu en considérant les composantes de courant (i_{as} , $i_{\beta s}$) comme grandeurs de commande, et les variables (Φ_{or} , $\Phi_{\beta r}$, Ω) comme variables d'état. Ainsi, le modèle réduit est régi par [BOD94], [KEZ96], [LAL99], [BAR03a]:

$$\dot{x} = f(x) + g(x) \cdot u \quad (\text{IV.91})$$

avec :

$$x = (x_3, x_4, x_5)^t = (\Phi_{or}, \Phi_{\beta r}, \Omega)^t;$$

$$u = (u_1, u_2)^t = (i_{as}, i_{\beta s})^t$$

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} = \begin{bmatrix} \frac{-x_3}{T_r} - p \cdot x_4 \cdot x_5 \\ \frac{-x_4}{T_r} + p \cdot x_3 \cdot x_5 \\ -\frac{C_r}{J} \end{bmatrix}; \quad g(x) = [g_1(x) \quad g_2(x)] = \begin{bmatrix} \frac{L_m}{T_r} & 0 \\ 0 & \frac{L_m}{T_r} \\ -\frac{pL_m}{JL_r} x_4 & \frac{pL_m}{JL_r} x_3 \end{bmatrix}$$

IV.7.5.1. Représentation du système sous forme canonique

Dans ce qui suivra, nous adopterons le modèle réduit du moteur asynchrone, régi par le système d'équations (IV.91). Nous remarquons que ce système est non-linéaire et qu'il existe un couplage étroit entre les différentes variables. Afin de mettre le système sous une forme canonique (ou normalisée), une transformation non-linéaire $h(x)$ est nécessaire. Cette transformation doit être difféomorphique [ISI89]. Dans ce sens, nous procédons en quatre étapes :

Étape 1 : Choix des grandeurs de sortie

Afin d'atteindre l'objectif de notre commande qui consiste en la régulation des deux grandeurs à savoir le flux et la vitesse, nous choisissons les variables de sorties suivantes [KAZ96], [LAL99], [BAR03a]:

$$\begin{aligned} h_1(x) &= \Phi_{\omega}^2 + \Phi_{\beta r}^2 = x_3^2 + x_4^2 \\ h_2(x) &= \Omega = x_5 \end{aligned} \tag{IV.92}$$

Étape 2 : Calcul du degré relatif

Nous calculons le degré relatif r_i associé à chaque grandeur de sortie z_i choisie.

– Sortie z_1 :

$$\dot{z}_1 = L_f h_1(x) + L_{g_1} h_1(x) \cdot u_1 + L_{g_2} h_1(x) \cdot u_2$$

$$\text{avec : } L_f h_1(x) = -\frac{2}{T_r} (x_3^2 + x_4^2), \quad L_{g_1} h_1(x) = \frac{2 L_m}{T_r} \cdot x_1, \quad L_{g_2} h_1(x) = \frac{2 L_m}{T_r} \cdot x_2$$

Le degré relatif correspondant à la grandeur de sortie z_1 est : $r_1 = 1$

– Sortie z_2 :

$$\dot{z}_2 = L_f h_2(x) + L_{g_1} h_2(x) \cdot u_1 + L_{g_2} h_2(x) \cdot u_2$$

$$\text{avec : } L_f h_2(x) = -\frac{C_r}{J}, \quad L_{g_1} h_2(x) = -\frac{p L_m}{J L_r} \cdot x_2, \quad L_{g_2} h_2(x) = \frac{p L_m}{J L_r} \cdot x_1$$

Le degré relatif associé à la grandeur de sortie z_2 est : $r_2 = 1$

La somme des degrés relatifs correspondants aux sorties $(x_3^2 + x_4^2)$ et x_5 est donnée par :

$$r_1 + r_2 = 1 + 1 = 2$$

L'ordre du nouveau système à commander est inférieur à celui du système initial. Ceci nous permet de conclure à l'existence d'une dynamique interne non contrôlable, qui doit être stable (au moins bornée) [ISI89].

Par conséquent, la représentation sous forme normalisée de ce système nécessite un choix approprié de la transformation non-linéaire de coordonnées qui englobe le choix d'une dynamique interne stable tout au plus bornée.

Étape 3 : Choix de la transformation $h(x)$

Nous optons ainsi pour la transformation non-linéaire de coordonnées suivante:

$$\begin{aligned} z_1 &= h_1(x) = x_3^2 + x_4^2 \\ z_2 &= h_2(x) = x_5 \\ z_3 &= h_3(x) = \text{atan} \left(\frac{x_4}{x_3} \right) + k \pi \end{aligned} \quad (\text{IV.93})$$

avec :

$$k = \begin{cases} 0 & \text{si } x_3 \geq 0 \\ 1 & \text{si } x_3 < 0 \end{cases} \quad \text{La valeur } k \text{ est utilisée pour assurer l'inversion de la transformation (IV.93).}$$

Ainsi, la transformation inverse est définie comme suit :

$$\begin{aligned} x_3 &= \sqrt{z_1} \cos z_3 \\ x_4 &= \sqrt{z_1} \sin z_3 \\ x_5 &= z_2 \end{aligned} \quad (\text{IV.94})$$

Et en se basant sur les équations (IV.91) et (IV.93), le système dans le nouvel espace d'état devient :

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} -\frac{2}{T_r} z_1 \\ -\frac{C_r}{J} \end{pmatrix} + \begin{pmatrix} 2 \frac{L_m}{T_r} x_3 & 2 \frac{L_m}{T_r} x_4 \\ -\frac{pL_m}{JL_r} x_4 & \frac{pL_m}{JL_r} x_3 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (\text{IV.95a})$$

$$\dot{z}_3 = p \cdot z_2 + \frac{L_m}{T_r} \left(\frac{x_3 u_2 - x_4 u_1}{z_1} \right) \quad (\text{IV.95b})$$

Le système (IV.95a) peut être mis sous la forme matricielle suivante :

$$\dot{z}_r = f(z) + G(z) \cdot u \quad (\text{IV.96})$$

avec : $z_r = (z_1, z_2)^t$ et $z = (z_1, z_2, z_3)^t$

$G(z)$ étant la matrice de découplage.

Etape 4 : Retour d'état non-linéaire

Comme il a été mentionner plus haut, afin que le retour d'état non-linéaire puisse être défini, il est impératif que la matrice de découplage soit inversible.

En effet, nous avons :

$$\det(G(z)) = \frac{2L_m}{T_r} \cdot \frac{pL_m}{JL_r} \cdot z_1 \neq 0 \quad (\text{IV.97})$$

En choisissant le retour d'état sous la forme :

$$u = G^{-1}(z) (v - f(z)) \quad (\text{IV.98})$$

nous obtenons les relations entre les nouvelles variables de commande (v_1, v_2) et les variables (u_1, u_2) définies comme:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \frac{1}{\det(G(z))} \begin{pmatrix} \frac{pL_m}{JL_r} x_3 & -\frac{2L_m}{T_r} x_4 \\ \frac{pL_m}{JL_r} x_4 & \frac{2L_m}{T_r} x_3 \end{pmatrix} \begin{pmatrix} v_1 + \frac{2}{T_r} z_1 \\ v_2 + \frac{C_r}{J} \end{pmatrix} \quad (\text{IV.99})$$

Le nouveau système d'état peut donc être représenté par deux intégrateurs correspondants aux deux sous-systèmes linéaires et entièrement découplés suivants:

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \end{aligned} \quad (\text{IV.100})$$

Néanmoins, nous remarquons que dans ce cas, le système d'équations (IV.99) est plutôt complexe, et dépend étroitement des variables d'état, des paramètres de la machine et des perturbations externes. Ce qui le rend sensible à toute variation paramétrique ou mal-identification. Par conséquent, sa représentation par un modèle approximatif n'est nullement évidente. Afin de minimiser la dépendance de ce système par rapport aux variations paramétriques et aux perturbations externes, tout en maintenant le découplage entre les deux sous-systèmes commandés par les variables de commandes v_1 et v_2 , nous proposons la relation suivante entre les commandes u et v :

$$u = G^{-1}(z) v \quad (\text{IV.101})$$

De manière plus explicite, ceci donne:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \frac{1}{\det(G(z))} \begin{pmatrix} \frac{pL_m}{JL_r} x_3 & -\frac{2L_m}{T_r} x_4 \\ \frac{pL_m}{JL_r} x_4 & \frac{2L_m}{T_r} x_3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (\text{IV.102})$$

Dans ce cas, le système d'état global devient :

$$\begin{aligned} \dot{z}_1 &= -\frac{2}{T_r} z_1 + v_1 \\ \dot{z}_2 &= -\frac{C_r}{J} z_2 + v_2 \end{aligned} \quad (\text{IV.103})$$

A partir de l'équation (IV-102), nous remarquons qu'il peut y avoir une autre écriture plus simple, et indépendante des paramètres de la machine, définie comme suit :

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} x_3 & -x_4 \\ x_4 & x_3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (\text{IV.104})$$

Et ce, en considérant v_1 et v_2 comme nouvelles variables de commande. La forme canonique correspondante au système régi par les équations (IV.95a) est donnée par :

$$\begin{aligned} \dot{z}_1 &= -\frac{2}{T_r} z_1 + \frac{2L_m}{T_r} z_1 \cdot v_1 \\ \dot{z}_2 &= -\frac{C_r}{J} z_2 + \frac{pL_m}{JL_r} z_1 \cdot v_2 \end{aligned} \quad (\text{IV.105})$$

Le système résultant comporte un retour d'état défini par la boucle interne (IV.104) et par un système non-linéaire (IV.105), constitué de deux sous-systèmes mis sous forme canonique et dépendants chacun d'une seule commande v_i ($i = 1, 2$) (Fig. IV.17).

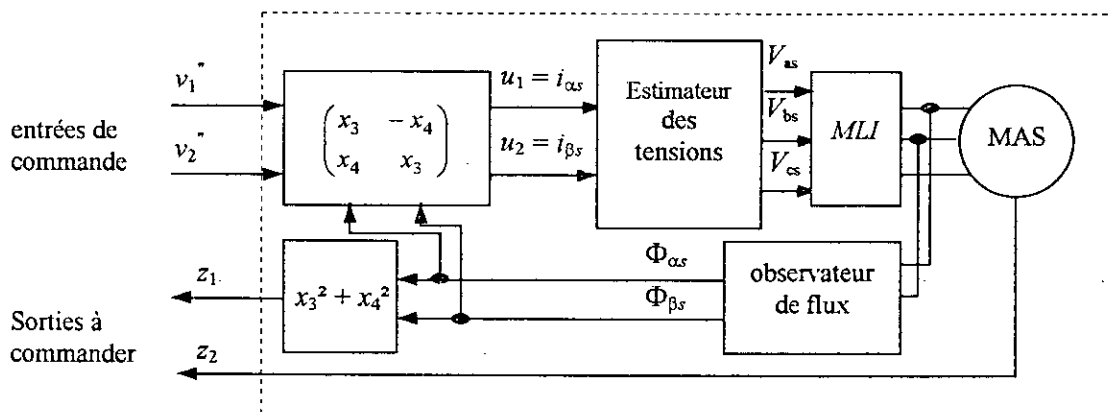


Figure.IV.17 : Représentation du modèle du moteur asynchrone basé sur le retour d'état non-linéaire adopté

Le bloc décrivant la relation entre les commandes u et v , illustré à la figure (IV.17), ne peut être remplacé par un système approximatif, car aucune incertitude entre u et v n'est tolérée. Ce qui signifie qu'une bonne observation des deux composantes du flux rotorique est nécessaire [GHA94], [BEN99a], [BEN99b].

Dans notre approche, nous allons approximer, par une représentation floue, le modèle du système dynamique non-linéaire et incertain régi par les équations (IV.105). Pour cela, on utilise la technique de modélisation floue des systèmes incertains introduite par Ben Ghalia [GHA94], [GHA96a]. Une commande floue robuste basée sur ce modèle flou [GHA96b], [BAR03a] sera ensuite adjointe.

IV.7.5.2. Présentation du modèle floue du moteur asynchrone

D'autre part, afin de mettre en évidence l'aptitude du modèle flou proposé à approximer le système (IV.105), nous proposons le schéma bloc de la figure (IV.19). L'évaluation de l'erreur entre la variable de sortie du modèle flou \tilde{z}_i et celle du système réel z_i permet de savoir si le modèle flou proposé est susceptible d'approximer le modèle réel ou pas. En effet et étant donné que la variable générée par le bloc représentant le modèle flou est une variable floue et afin de pouvoir être comparée à celle fournie par le modèle réel, une phase de déffuzification se trouve être nécessaire. Dans notre cas la méthode de déffuzification utilisée est régie par la relation suivante:

$$\tilde{z}_i = \frac{\sum_{k_1} \sum_{k_2} \tilde{z}_i^{(k_1, k_2)} \cdot \min(\mu_{z_1}^{k_1}, \mu_{v_i}^{(k_1, k_2)})}{\sum_{k_1} \sum_{k_2} \min(\mu_{z_1}^{k_1}, \mu_{v_i}^{(k_1, k_2)})} \quad (\text{IV.106})$$

avec : $\tilde{z}_i^{(k_1, k_2)}$ la variable de commande qui permet d'avoir un degré d'appartenance à l'ensemble flou $F_{v_i}^{(k_1, k_2)}$ maximal définie comme suit:

$$\tilde{z}_i^{(k_1, k_2)} = \sup_{z_i \in U_{z_i}} \mu_{F_{z_i}^{(k_1, k_2)}}(z_i) \quad ; \quad i = 1, 2 \quad (\text{IV.107})$$

$$k_i = 1, \dots, M_i$$

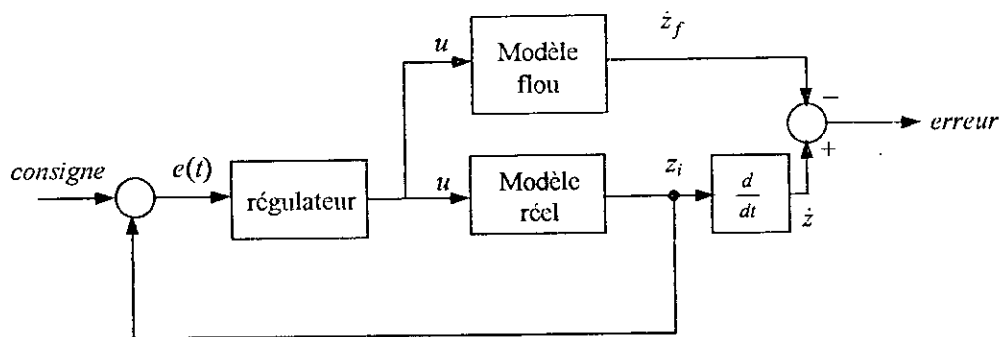


Figure IV.19 : Schéma bloc de la comparaison entre la sortie du modèle réel de la machine et flou proposé dans le cas de la commande par retour d'état linéaire.

IV.7.5.3. Résultats de simulation

Dans le but de mettre en évidence les avantages de l'approche de modélisation floue proposée, basée sur l'algorithme de **Ben-Ghali**, une simulation numérique est effectuée comparant les performances obtenues dans le cas du système avec model flou et celui réel respectivement.

L'essai effectué sur le bloc diagramme de la figure (IV.19), consiste en un démarrage à vide sous une référence de vitesse $\Omega_{ref} = 100$ rd/sec appliquée entre $t = 0$ sec et $t = 1.5$ sec.

Il est important de noter que dans l'approche proposée de commande floue basée sur le modèle flou, il n'est nullement nécessaire de connaître les valeurs nominales des paramètres incertains.

Les résultats de simulation obtenus dans le cas idéal et flou avec des partitionnements de l'univers de discours en 7 et 21 ensembles flous respectivement sont illustrés à la figure (IV.20).

Par ailleurs, il est à noter que plus le nombre de partitions augmente, plus le modèle se rapproche du modèle original.

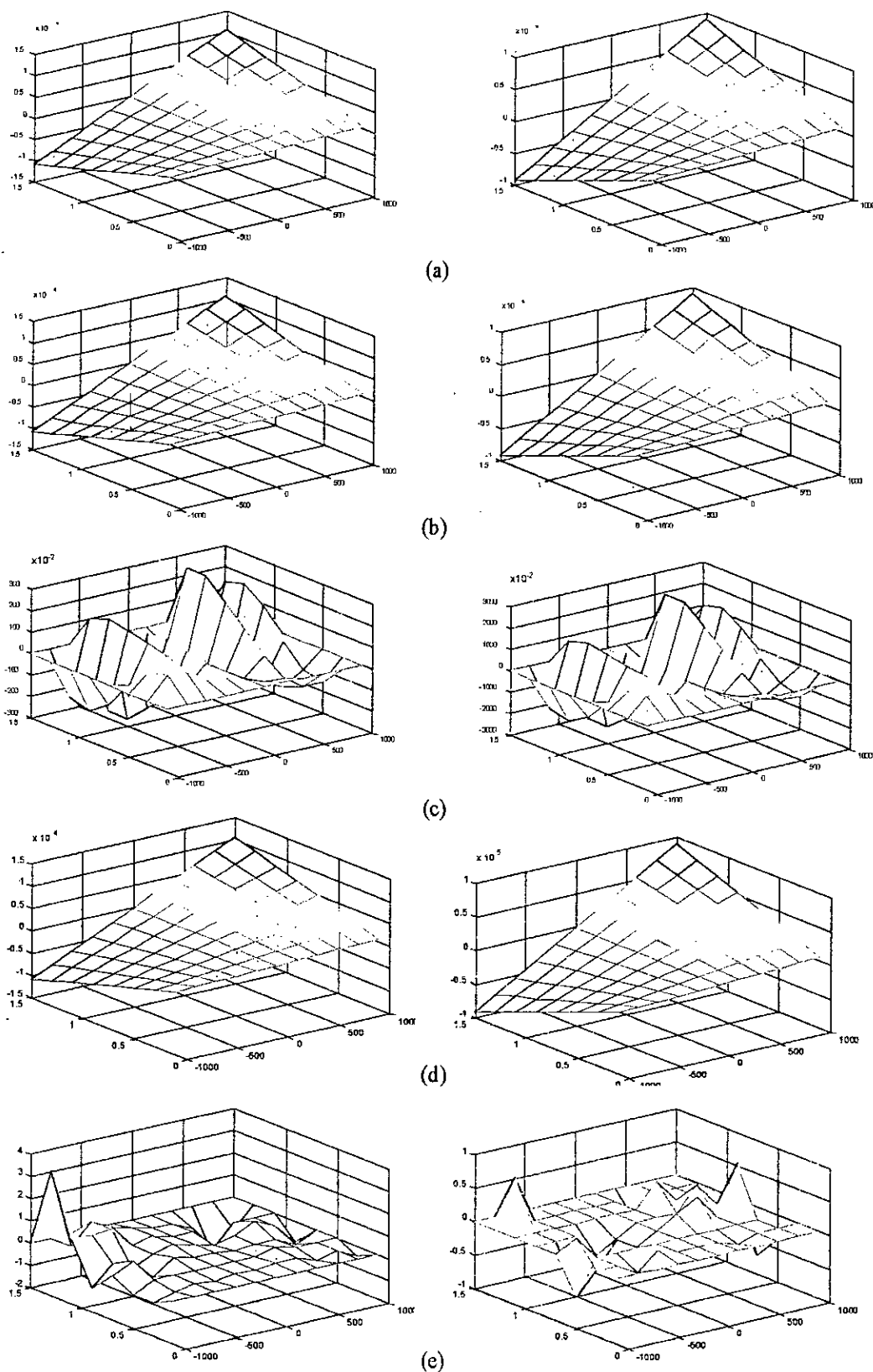


Figure.IV.19: Comparaison entre le modèle flou et l'original.

(a) dérivée du flux et de la vitesse du modèle original respectivement, (b) dérivée du flux et de la vitesse du modèle flou respectivement avec un nombre de partitions égal à 7 chacun. (c) erreurs entre les dérivées du flux et ceux de la vitesse respectivement obtenus pour chacun des deux modèles (d) et (e) sont les résultats obtenus pour les mêmes essais que (b) et (c) avec un nombre de partitions égal à 21.

IV.7.5.4. Commande par mode de glissement

Dans notre cas, les régulateurs sont choisis comme étant des régulateurs à structure variable de la structure de réglage de la machine asynchrone dans le repère lié au stator (Fig. IV.21).

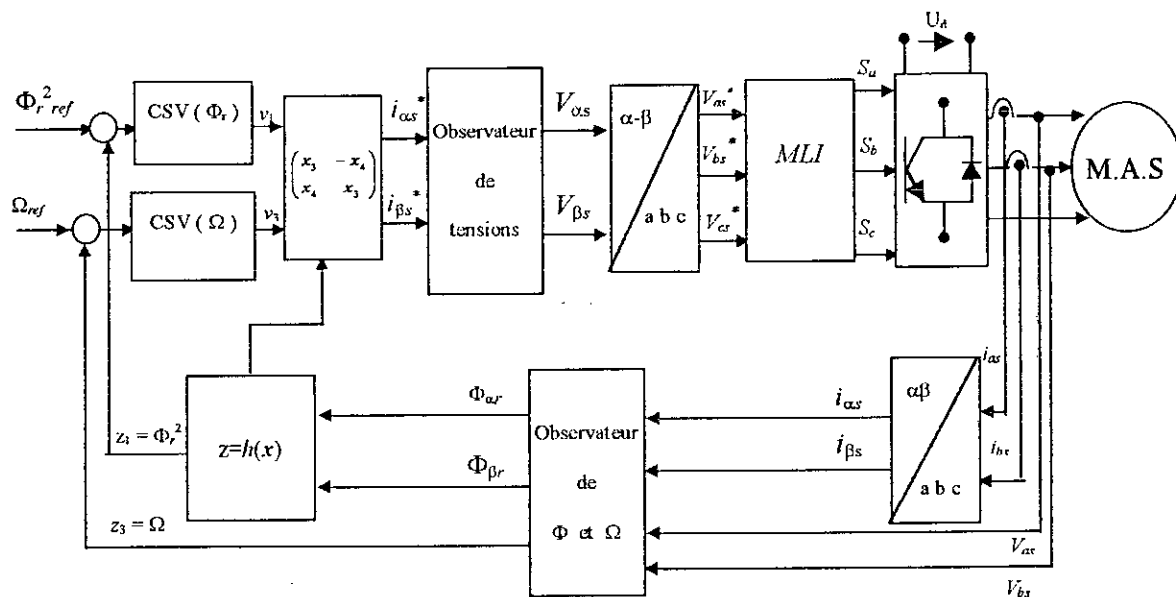


Figure.IV.21: Commande par mode de glissement basée sur un retour d'état non-linéaire de la commande de la machine asynchrone.

Les étapes suivies pour l'élaboration de ces régulateurs sont les suivantes :

– *Choix de la surface de glissement*

Les fonctions de commutation des deux sous-systèmes sont choisies de la forme suivante:

$$\begin{aligned} s_1(z) = e_1(z) &= \Phi_r^2 \text{ref} - \Phi_r^2 = z_{1\text{ref}} - z_1 \\ s_2(z) = e_2(z) &= \Omega_{\text{ref}} - \Omega_r = z_{2\text{ref}} - z_2 \end{aligned} \quad (\text{IV.108})$$

– *Choix des grandeurs de commande*

En utilisant les équations (IV.104) et (IV.105), les deux lois de commande correspondant au réglage du flux et de la vitesse sont définies par :

$$v_1 = \frac{-\lambda_1 \cdot \dot{z}_{1\text{ref}} + \lambda_1 \cdot \frac{2}{T_r} z_1 - \dot{s}_{1d}}{\frac{2L_m}{T_r} z_1} \quad (\text{IV.109})$$

$$v_2 = \frac{-\lambda_2 \cdot \dot{z}_{2ref} + \lambda_2 \cdot \frac{C_r}{J} - \dot{s}_{2d}}{\frac{2L_m}{JL_r} z_1} \quad (\text{IV.110})$$

avec :

$$\dot{s}_{id}(x) = -M_i \cdot \text{sgn}(s_i(x^i)) \quad ; \quad i = 1, 2 \quad (\text{IV.111})$$

a) Pour le régulateur de vitesse

$$v_1 = \left(\frac{-\lambda_1 \cdot \dot{z}_{1ref} + \lambda_1 \cdot \frac{2}{T_r} z_1}{\frac{2L_m}{T_r} z_1} \right) + \left(\frac{-1}{\frac{2L_m}{T_r} z_1} \right) \cdot \dot{s}_{1d} \quad (\text{IV.112})$$

b) Pour le régulateur de flux

$$v_2 = \left(\frac{-\lambda_2 \cdot \dot{z}_{2ref} + \lambda_2 \cdot \frac{C_r}{J}}{\frac{2L_m}{JL_r} z_1} \right) + \left(\frac{-1}{\frac{2L_m}{JL_r} z_1} \right) \cdot \dot{s}_{2d} \quad (\text{IV.113})$$

D'une autre manière:

$$v_1 = v_{1eq}(z_1) + b_1(z_1) \cdot \dot{s}_{1d} \quad (\text{IV.114})$$

$$v_2 = v_{2eq}(z_1) + b_2(z_1) \cdot \dot{s}_{2d} \quad (\text{IV.115})$$

avec:

$$\dot{s}_{1d}(s_1) = M_1(s_1) \cdot \text{sgn}(s_1) \quad (\text{IV.116})$$

$$\dot{s}_{2d}(s_2) = M_2(s_2) \cdot \text{sgn}(s_2) \quad (\text{IV.117})$$

$$M_i = K_i - (K_i - k_i) \cdot \exp\left(-\frac{|s_i(z_i) - \varepsilon|}{\sigma_i}\right) \quad \text{si } s_i(z_i) > \varepsilon \quad i = 1, 2 \quad (\text{IV.118})$$

IV.7.5.5. Résultats de simulations

Afin de mettre en évidence les avantages apportés par la commande à structures variables, une simulation numérique a été effectuée dans les mêmes conditions d'essai que précédemment et avec une plage de variation de la résistance rotorique R_r de 50% à 200% par rapport à la résistance nominale.

Les résultats de simulation montrent la dynamique de réglage de la vitesse et le rejet de toute perturbation. Nous notons que, dans tous les cas, la variation du couple de charge n'affecte pas le flux en

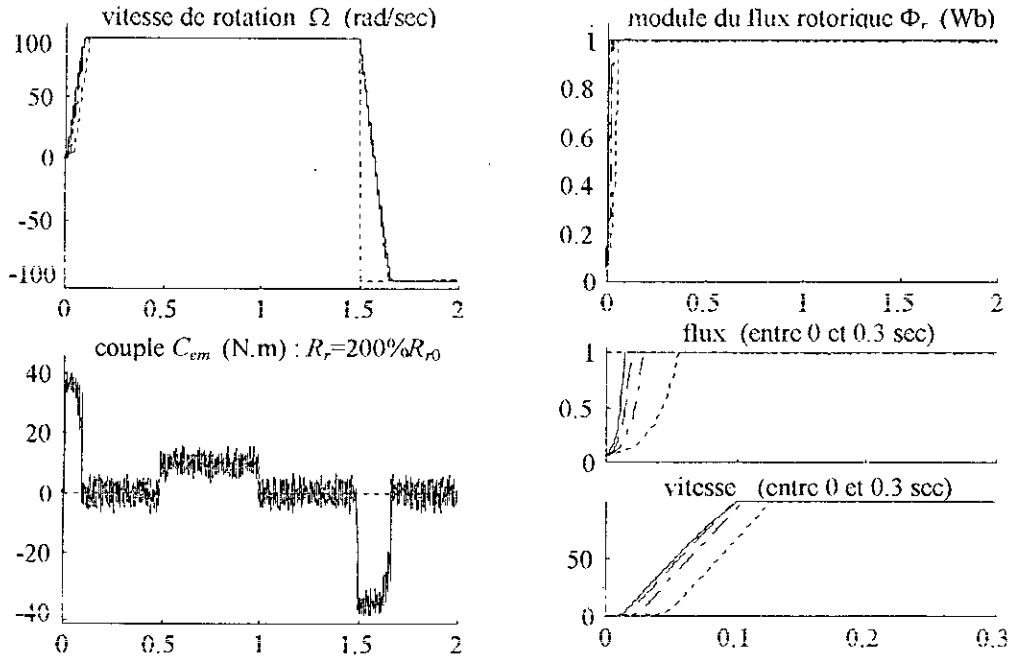
régime permanent, ce qui nous permet d'affirmer que le découplage entre les deux grandeurs, couple et flux rotorique, est réalisé.

Les résultats de simulation ainsi obtenus dans le cas de la commande à structure variable à relais sont représentés aux figures (IV.21a) et (IV.21b). Les résultats montrent clairement l'existence d'une meilleure robustesse par rapport à la commande par placement de pôles vis-à-vis de la variation de la résistance rotorique R_r , et de la perturbation externe. Néanmoins, des oscillations très prononcées apparaissent, et qui peuvent s'avérer néfastes pour les composants semi-conducteurs et le fonctionnement du moteur due au phénomène de chattering.

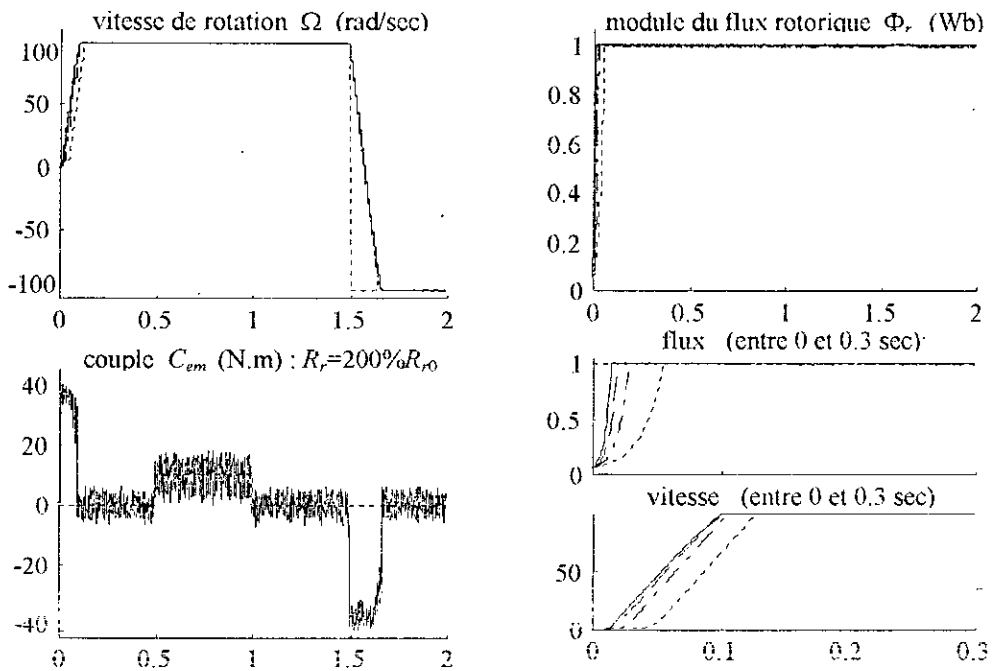
La figure (IV.22) montre qu'effectivement, la commande à structure variable adoucie ($k = 0$) permet une nette réduction des oscillations. Néanmoins, cet avantage se fait au prix d'une perte de la robustesse, vu la sensibilité de la vitesse par rapport au couple résistant.

Nous apprécions, sur la figure (IV.23), les améliorations apportées par la commande à structures variables à condition qu'un choix judicieux des paramètres ε et k est effectué. Le paramètre k agit sur la robustesse et le chattering, alors que ε influe sur la précision et la fréquence de commutation des semi-conducteurs.

L'étude de la robustesse montre que la commande par mode de glissement offre l'avantage d'être, par rapport à la commande par placement de pôles et la commande par flux orienté, insensible aux variations des paramètres internes de la machine et des perturbations externes.



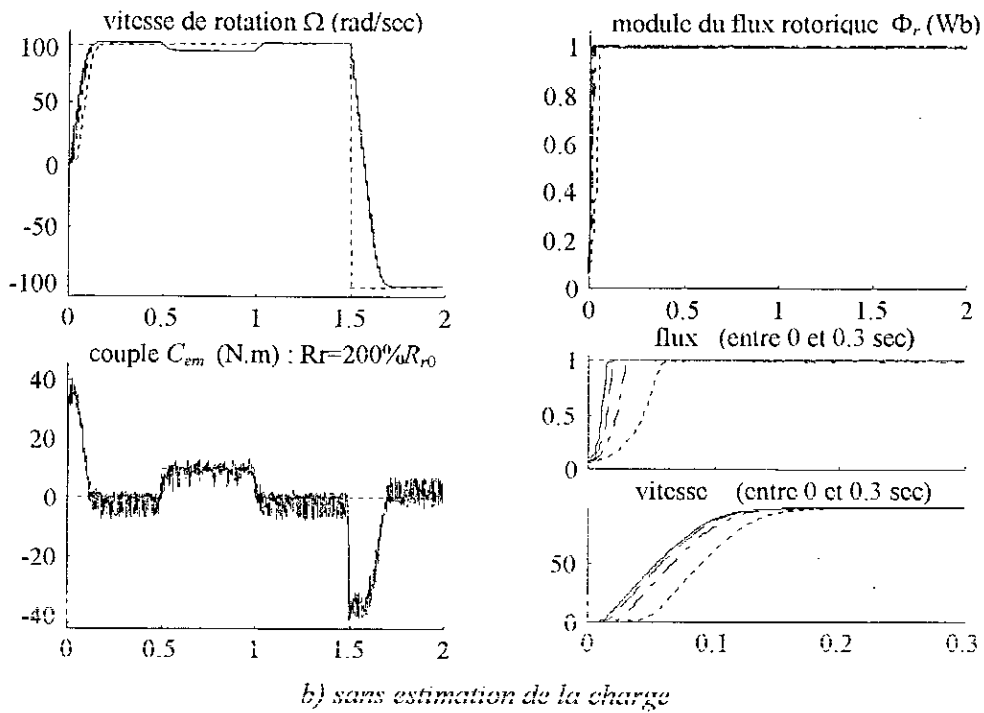
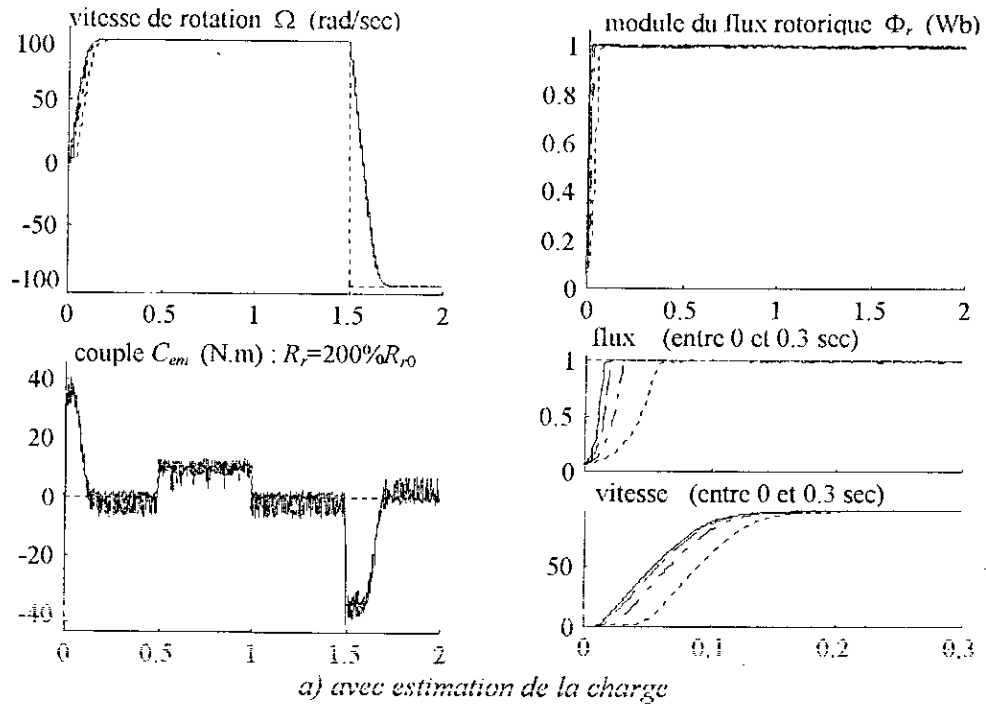
a) avec estimation de la charge



b) sans estimation de la charge

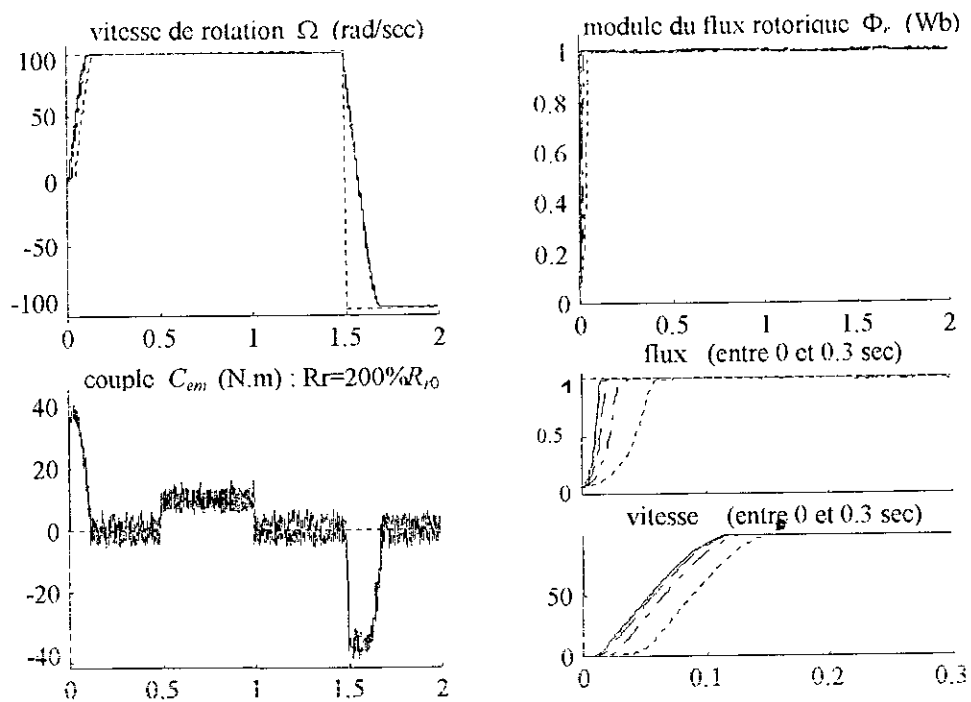
..... référence $R_r = 50\%R_{r0}$ - - - $R_r = 150\%R_{r0}$
 - - - $R_r = 100\%R_{r0}$ ——— $R_r = 200\%R_{r0}$

Figure IV.21: Commande à structure variable à relais ($k=K$). Résultats de simulation pour différentes variations de R_r

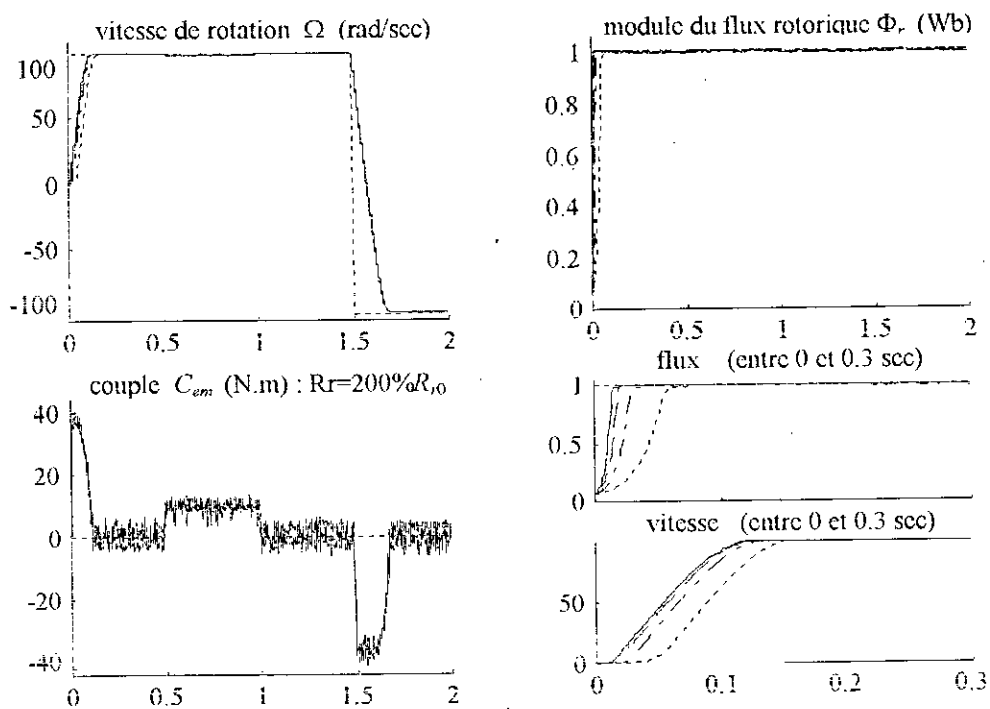


..... référence $R_r = 50\%R_{r0}$ --- $R_r = 150\%R_{r0}$
 --- $R_r = 100\%R_{r0}$ — $R_r = 200\%R_{r0}$

Figure IV.22 : Commande à structure variable adoucie ($k=0$). Résultats de simulation pour différentes variations de R_r



a) avec estimation de la charge



b) sans estimation de la charge

..... référence $R_r = 50\%R_{r0}$ - - - $R_r = 150\%R_{r0}$
 - - - $R_r = 100\%R_{r0}$ - - - $R_r = 200\%R_{r0}$

Figure IV.23 : Commande à structure variable semi-adoucie ($k \neq 0$). Résultats de simulation pour différentes variations de R_r

IV.7.6. Commande robuste floue de Ben-Ghalia de la machine asynchrone

En se basant sur le modèle flou obtenu précédemment il est possible de concevoir une commande robuste floue comme le montre la figure (IV.24).

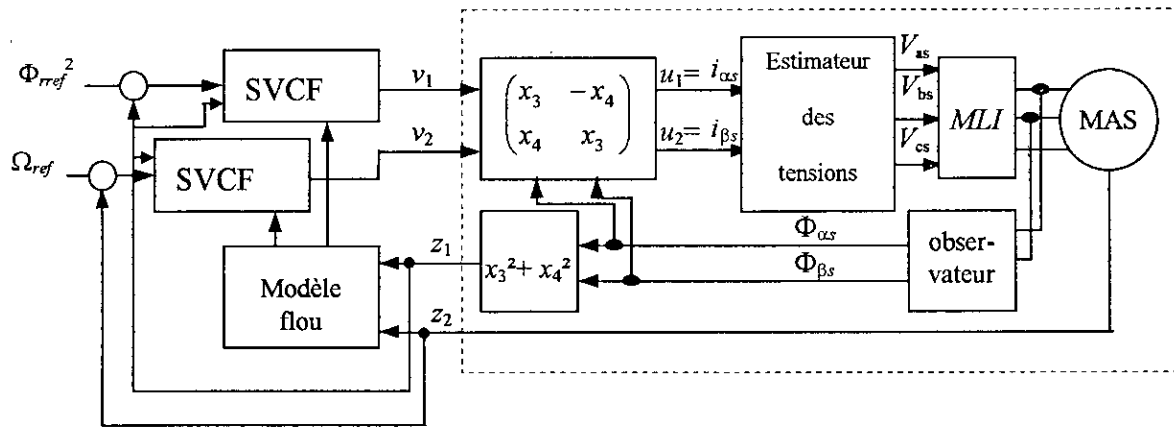


Figure.IV.24 : Schéma de Commande à structure variable floue basée sur le modèle flou du moteur asynchrone

IV.7.6.1. Elaboration de la commande floue robuste

Une fois le modèle flou obtenu, nous pouvons passer à la conception du contrôleur flou. D'une manière générale, plusieurs contrôleurs flous peuvent être élaborés. Etant donné que le schéma de commande est à structures variables, nous avons opté pour la conception d'un contrôleur flou robuste. Pour ce faire, nous procédons de la manière suivante :

Étape 1. Construction des variables d'erreur floues

Etant donné, les variables d'erreur $e_1(t)$ et $e_2(t)$ définies comme suit:

$$e_1 = z_{ref1} - z_1 \quad (IV.119)$$

$$e_2 = z_{ref2} - z_2$$

z_{ref1} et z_{ref2} sont choisies constantes.

Les variables d'erreur floues \tilde{e}_1 et \tilde{e}_2 correspondantes sont alors donné par:

$$\tilde{e}_1 = \tilde{z}_{ref1} - \tilde{z}_1 \quad (IV.120)$$

$$\tilde{e}_2 = \tilde{z}_{ref2} - \tilde{z}_2$$

avec : \tilde{z}_{ref1} et \tilde{z}_{ref2} des singletons flous.

La partition floue $P_F(U_{e_i})$ de l'univers de discours U_{e_i} de chaque variable d'entrée ($i = 1, 2$) est construite comme suit :

$$P_F(U_{e_i}) = \left\{ F_{e_i}^{k_i} = \tilde{z}_{ref i} - F_{z_i}^{k_i} ; k_i = 1, \dots, M_i \right\} \quad (IV.121)$$

où $F_{e_i}^{k_1}$, $F_{e_i}^{k_2}$ ($k_1 = 1, \dots, M_1$; $k_2 = 1, \dots, M_2$) sont des ensembles flous lesquelles dans le présent algorithme peuvent prendre les variables floues \tilde{e}_1 et \tilde{e}_2 .

Étape 2. Construction des fonctions floues de commutation

Les extensions floues des fonctions de commutation $s_1(e_1)$ et $s_2(e_2)$ quant à elles peuvent être écrites de la manière suivante:

$$\tilde{s}_1(\tilde{e}_1) = \lambda_1 \sim \tilde{e}_1 \quad \text{et} \quad \tilde{s}_2(\tilde{e}_2) = \lambda_2 \sim \tilde{e}_2 \quad (IV.122)$$

Les ensembles flous que peut prendre chacune des fonctions floues de commutation sont alors donné par:

$$\begin{aligned} F_{s_1}^{k_1} &= \lambda_1 \sim F_{e_1}^{k_1} \\ F_{s_2}^{k_2} &= \lambda_2 \sim F_{e_2}^{k_2} \end{aligned} \quad (IV.123)$$

Étape 3. Choix de la dynamique floue désirée

Durant le mode d'atteignabilité, la robustesse est assurée par le choix des deux dynamiques floues désirées \tilde{s}_{d1} et \tilde{s}_{d2} qui sont des extensions des dynamiques désirées s_{di} définies par l'équation (IV.111).

Les ensembles flous $F_{s_i}^{k_i}$ ($i = 1, 2$) correspondant à \tilde{s}_{di} , sont par conséquent, choisis en se référant à l'extension des deux fonctions non-linéaires $M_1(s_1)$ et $M_2(s_2)$.

$$F_{s_i}^{(k_i)} = -\omega_{s_i} \sim F_{M(PF_{s_i})}^{(k_i)} \mp (\omega_{s_i} - 1) \sim F_{M(NF_{s_i})}^{(k_i)} \quad (IV.124)$$

avec :

$$\omega_{s_i} = \begin{cases} 1 & \text{si } s_i \in \text{supp}(F_{s_i}^{k_i}) \cap U_{s_i}^+ \\ 0 & \text{si } s_i \in \text{supp}(F_{s_i}^{k_i}) \cap U_{s_i}^- \end{cases} \quad (IV.125)$$

$U_{s_i}^+ \subset \mathbb{R}^+$ et $U_{s_i}^- \subset \mathbb{R}^-$ sont les parties strictement positive et strictement négative de U_{s_i} .

$PF_{s_i}^{k_i}$ et $NF_{s_i}^{k_i}$ sont respectivement la partie positive et la partie négative de l'ensemble flou $F_{s_i}^{k_i}$.

$F_{M(PF_{s_i})}^{(k_i)}$ les ensembles flous que peut prendre $\tilde{M}_1(\tilde{s}_1)$ lorsque $\tilde{s}_i = PF_{s_i}^{(k_i)}$.

$F_{M(NF_{s_i})}^{(k_i)}$ les ensembles flous que peut prendre $\tilde{M}_2(\tilde{s}_2)$ lorsque $\tilde{s}_i = NF_{s_i}^{(k_i)}$.

Étape 4. Calcul de la loi de commande floue

D'après les deux dernières équations, nous pouvons déduire les lois de commande floues définies comme:

$$\tilde{v}_1 = \left(-\tilde{f}_1(\tilde{z}_1) \tilde{+} \tilde{z}_{ref1} \tilde{+} \frac{1}{\lambda_1} \tilde{\sim} \tilde{s}_{d1}(\tilde{e}_1) \tilde{+} \frac{1}{\lambda_1} \tilde{\sim} \tilde{e}_1 \right) \tilde{\approx} \tilde{g}_1(\tilde{z}_1) \quad (\text{IV.126})$$

$$\tilde{v}_2 = \left(-\tilde{a} \tilde{+} \tilde{z}_{ref2} \tilde{+} \frac{1}{\lambda_2} \tilde{\sim} \tilde{s}_{d2}(\tilde{e}_2) \tilde{+} \frac{1}{\lambda_2} \tilde{\sim} \tilde{e}_2 \right) \tilde{\approx} \tilde{g}_2(\tilde{z}_1)$$

Ce système d'équation nous permet de déduire les ensembles flous $F_{u_i}^{(k_1, k_2)}$, tels que :

$$F_{v_1}^{(k_1, k_2)} = \left(-F_{f_1}^{(k_1)} \tilde{+} \tilde{z}_{ref1} \tilde{+} \frac{1}{\lambda_1} \tilde{\sim} F_{s_{d1}}^{(k_1)} \tilde{+} \frac{1}{\lambda_1} \tilde{\sim} F_{e_1}^{(k_1)} \right) \tilde{\approx} F_{g_1}^{(k_1)} \quad (\text{IV.127})$$

$$F_{v_2}^{(k_1, k_2)} = \left(-\tilde{a} \tilde{+} \tilde{z}_{ref2} \tilde{+} \frac{1}{\lambda_2} \tilde{\sim} F_{s_{d2}}^{(k_2)} \tilde{+} \frac{1}{\lambda_2} \tilde{\sim} F_{e_2}^{(k_2)} \right) \tilde{\approx} F_{g_2}^{(k_1)}$$

et la base de règles correspondant à l'entrée de commande u pour chaque sous-système :

$$\begin{array}{ll} \mathfrak{R}_{v_1}^{(k_1)} : & \text{IF } (z_1, e_1)^t \text{ is } (F_{z_1}^{k_1}, F_{e_1}^{k_1})^t \\ & \text{THEN } v_1 \text{ is } F_{v_1}^{(k_1)} \end{array} \quad (\text{IV.128})$$

$$\begin{array}{ll} \mathfrak{R}_{u_2}^{(k_1, k_2)} : & \text{IF } (z_1, e_2)^t \text{ is } (F_{z_1}^{k_1}, F_{e_2}^{k_2})^t \\ & \text{THEN } v_2 \text{ is } F_{v_2}^{(k_1, k_2)} \end{array} \quad (\text{IV.129})$$

IV.7.6.2. Résultats de simulation

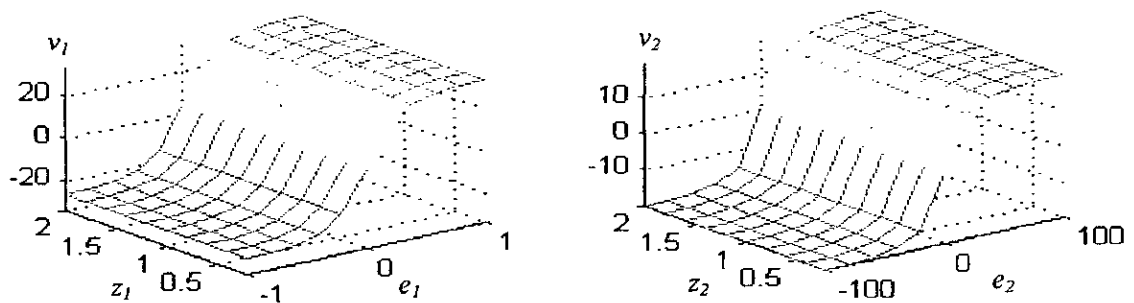
Une simulation numérique de la commande à structure variable floue basée sur le modèle flou linguistique du moteur asynchrone est effectuée. Afin de pouvoir comparer les résultats, les mêmes essais

que ceux de la commande à structure variable réelle ont été réalisés pour différents nombres de partitions (3, 7 et 15).

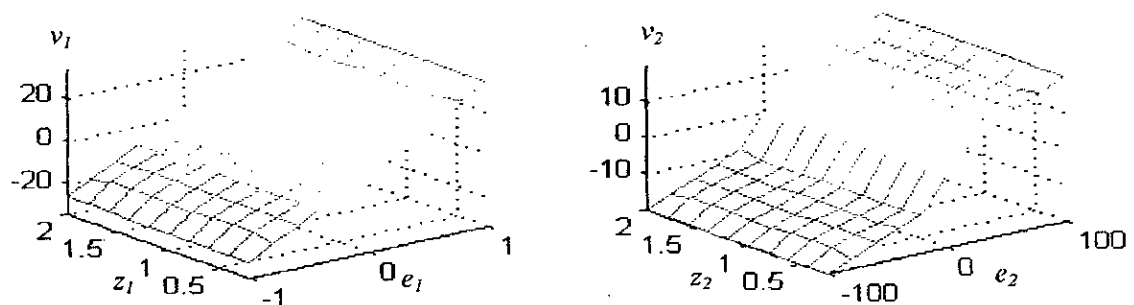
La comparaison entre ces deux commandes est illustrée aux figures (IV.25) et (IV.26). Il est important de spécifier que la plage d'incertitudes n'est pas considérée dans l'élaboration du modèle flou. Ce test montre qu'effectivement l'approximation est d'autant plus précise que le nombre de partitions de l'espace d'entrée augmente (l'approximation est meilleure à partir d'un partitionnement de 7 ensembles flous).

Afin de montrer l'effet de cette précision sur le comportement du système, nous avons effectué une simulation de la commande relative à deux partitions différentes $M_i = 3$ puis 7, tout en gardant les mêmes conditions d'essai. Les résultats obtenus sont présentés dans la figure (IV.27). Nous remarquons, dans ce cas aussi, que plus le nombre de partitions augmente, plus la commande robuste floue est susceptible d'approximer la commande à structure variable conventionnelle.

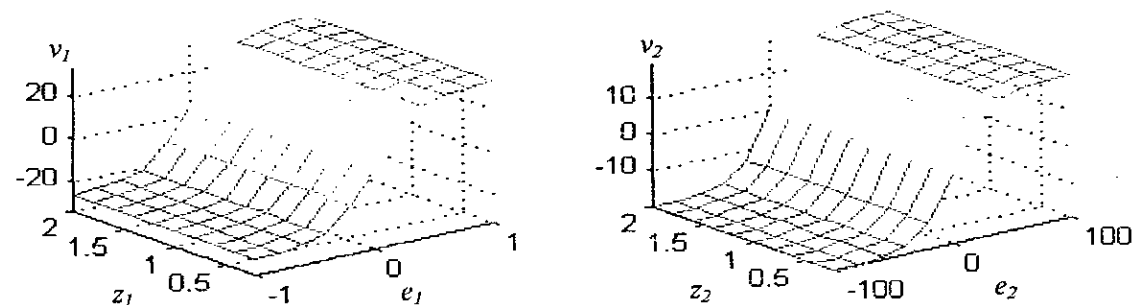
De la même façon que pour le cas réel, un test de robustesse a été effectué pour un nombre de partitions égal à 3, 7 et 15, pour une variation de la résistance rotorique de 50% jusqu'à 200% par rapport à la résistance nominale, et pour un couple de charge non estimé. La plage des incertitudes est prise en compte par le modèle flou. Les réponses obtenues sont représentées par les figures (IV.28-IV.29). Les résultats obtenus mettent en évidence la robustesse de la commande proposée vis à vis des variations paramétriques avec en plus une nette réduction des oscillations (chattering), ainsi qu'une meilleure poursuite des références par rapport à la commande à structures variables (Fig.IV.21). Par ailleurs, ils montrent également que le contrôleur utilisant le modèle flou avec un partitionnement de 7 ensembles flous de l'univers de discours des variables d'entrée est robuste, malgré la mauvaise approximation du modèle réel original. Ce qui prouve que le modèle proposé dans sa forme canonique est efficace et que la commande à structure variable floue a bien pris en compte la plage des incertitudes.



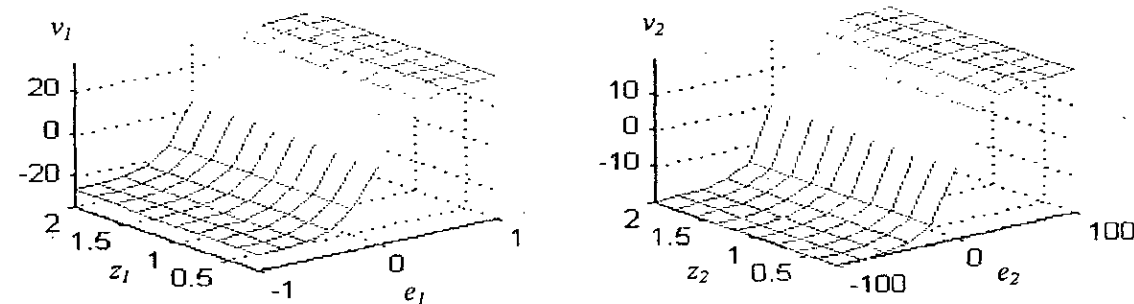
a) sorties de commande pour le cas réel



b) sortie de commande floue pour $M_i = 3$

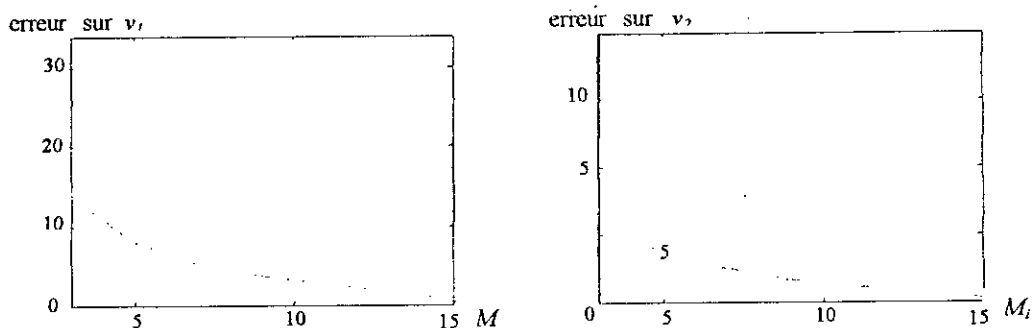


c) sortie de commande floue pour $M_i = 7$

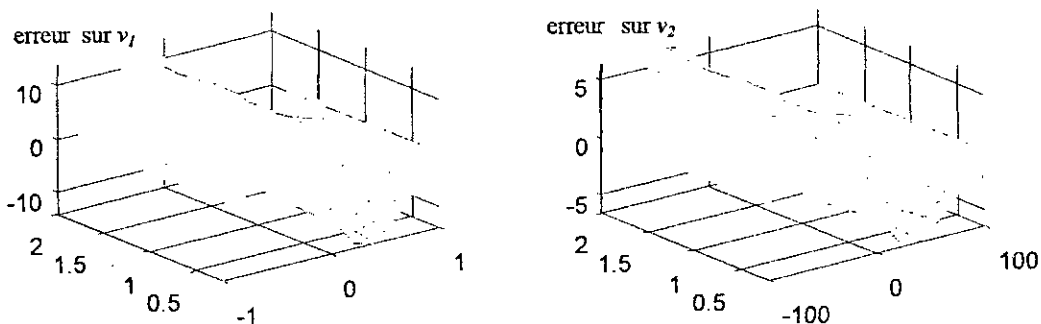


d) sortie de commande floue pour $M_i = 15$

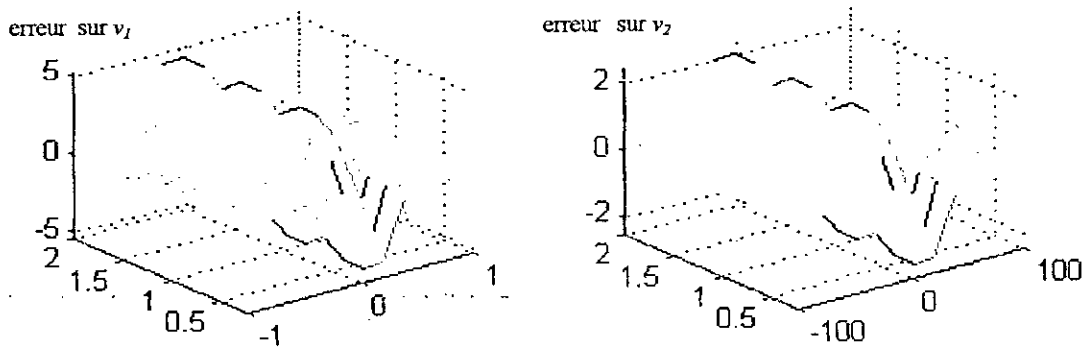
Figure IV.25 : Commande à structure variable floue. Comparaison entre les commandes réelle et floue pour différentes valeurs de M_i



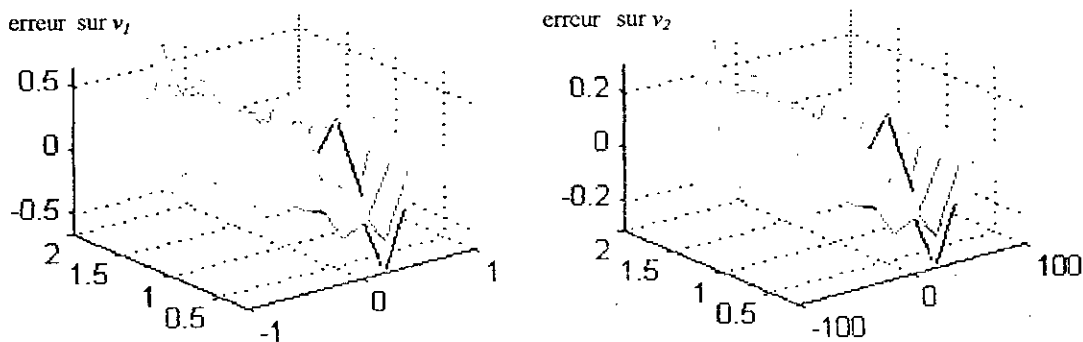
a) erreurs maximales en fonction de M_i



b) erreur entre la commande réelle et floue pour $M_i = 3$



c) erreur entre la commande réelle et floue pour $M = 7$



d) erreur entre la commande réelle et floue pour $M = 15$

Figure IV.26 : Commande à structure variable floue. Comparaison entre les commandes floue et réelle pour différentes valeurs de M_i

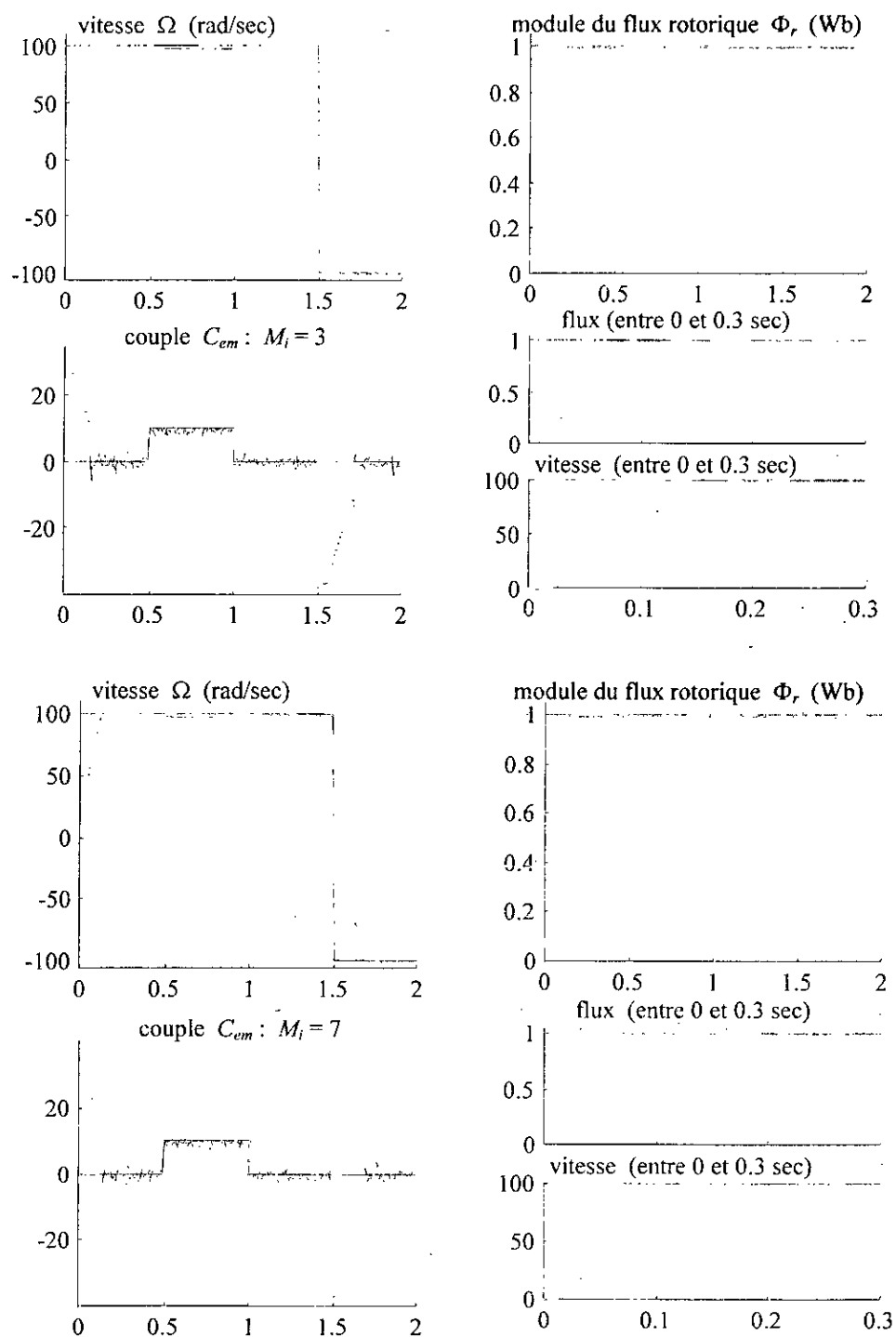
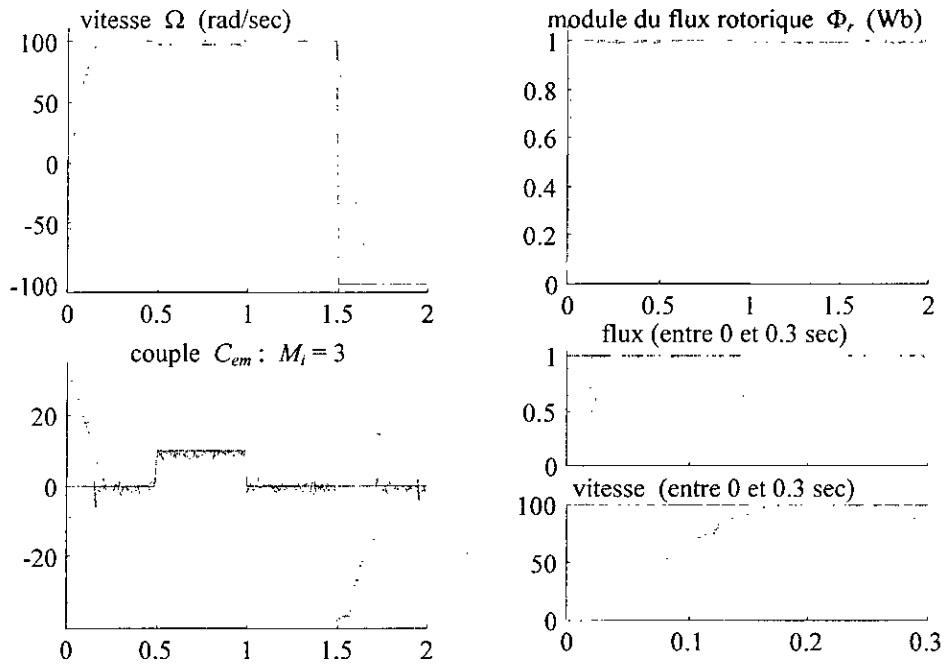
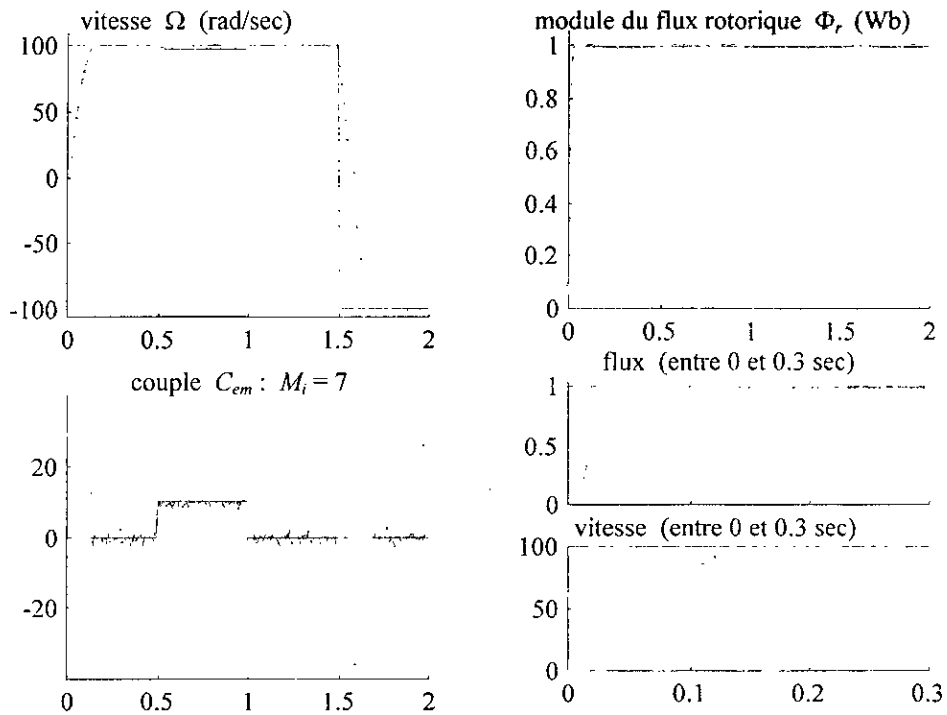


Figure IV.27 : Résultats de simulation pour $M_i = 3$ et 7 de la commande à structure variable floue (cas idéal).



a) avec un partitionnement de l'espace d'entrée en 3 ensembles flous



b) avec un partitionnement de l'espace d'entrée en 7 ensembles flous

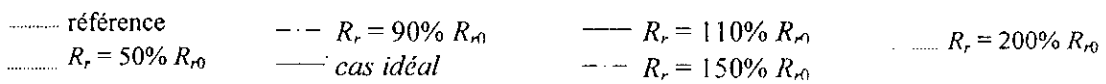


Figure IV.28 : Commande à structure variable floue. Résultats de simulation pour différentes variations de R_r (sans estimation de charge et avec une variation de 5% sur les inductances)

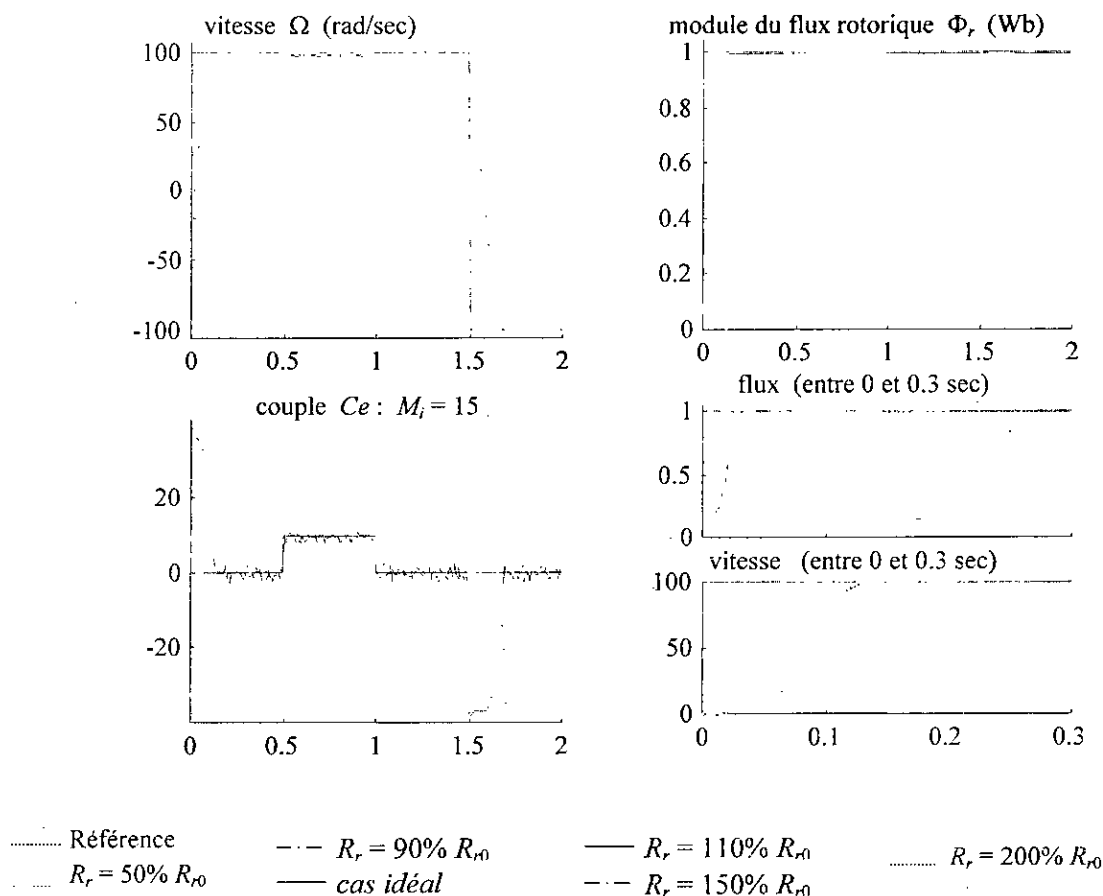


Figure IV.29 : Commande à structure variable floue. Résultats de simulation pour différentes variations de R_r (avec un partitionnement de l'espace d'entrée de 15 ensembles flous, sans estimation de charge et avec une variation de 5% sur les inductances)

IV.8. CONCLUSION

Dans ce chapitre, nous avons présenté une façon d'introduire la modélisation floue dans la commande d'un moteur asynchrone. La modélisation floue adoptée est de type linguistique, elle exploite la connaissance préalable du modèle analytique décrivant grossièrement le comportement du système incertain.

Cette méthode concerne l'obtention du modèle flou linguistique du moteur asynchrone présenté dans le référentiel statorique sous forme canonique. Ce modèle prend en charge toutes les incertitudes (paramétriques, non-paramétriques et structurelles), afin d'assurer une bonne robustesse lors de son introduction dans la conception de la commande floue robuste.

La commande à structure variable floue basée sur le modèle flou du système a été présentée pour une large gamme de modèles, puis pour les systèmes mis sous forme canonique, avant de l'appliquer dans le cas du moteur asynchrone pour le réglage du flux rotorique et de la vitesse. Cette technique a donné de très bons résultats comparés à ceux obtenus dans le cas de la commande par structures variables, du point de vue robustesse vis à vis des variations paramétriques et perturbations externes, poursuite des références imposées et surtout une nette réduction du phénomène de chattering.

Conclusion Générale

Conclusion générale

Les divers travaux de recherche entrepris jusqu'à nos jours, ont montré que la difficulté de la commande du moteur asynchrone réside essentiellement dans son modèle mathématique caractérisé par :

- une forte non-linéarité et un couplage étroit entre les variables internes (couple et flux).
- la présence des incertitudes et la variation des paramètres dues à l'effet thermique et au vieillissement de la machine.
- la difficulté d'accès aux variables d'état (flux rotorique).

Pour trouver des solutions à ces contraintes, deux stratégies de commande classiques du moteur asynchrone ont été d'abord présentées, à savoir : la commande par flux orienté et la commande par placement de pôles basée sur un retour d'état linéarisant du modèle. Elles ont été prises comme base pour la conception d'une autre commande dite hybride.

En ce qui concerne la commande par flux orienté celle-ci a montré son efficacité du point de vue découplage des deux principales grandeurs *Couple-Flux*, dans le régime permanent, à condition de maintenir le flux constant. D'autre part, comme le retour d'état linéarisant est connu pour la réalisation d'un découplage indépendamment du régime de fonctionnement, l'idée d'une commande hybride combinant la commande vectorielle et celle linéarisante a été présentée. En plus du découplage réalisé, et du point de vue implémentation, le schéma de cette structure de réglage est plus simple à mettre en œuvre. Néanmoins, l'inconvénient majeur déjà rencontré dans les deux commandes d'origine et qui concerne la sensibilité vis-à-vis des incertitudes inhérentes au système et des perturbations externes subsiste toujours.

Dans ce contexte, et comme plusieurs approches de commande robustes intègrent lors de leur élaboration, les fluctuations paramétriques et les perturbations de charge injectées dans le processus de commande, nous avons envisagé d'intégrer l'une de ces techniques dites à structures variables dans la structure de réglage hybride pour la conception des régulateurs de vitesse et du flux rotorique.

A cet effet, l'objectif de notre travail consistait à trouver un meilleur moyen pour synthétiser et concevoir ces régulateurs tout en gardant les mêmes, voire de meilleures performances dynamiques que celles obtenues avec les techniques conventionnelles précédentes.

La commande à structure variable apporte de meilleures performances, principalement en ce qui concerne la robustesse. D'autre part, le temps de réponse ainsi que le temps de calcul sont également améliorés. Un autre avantage appréciable apporté par cette technique est qu'elle ne nécessite, au préalable, ni l'estimation de la résistance rotorique, ni celle du couple résistant. Pour son élaboration, les surfaces de glissements, ainsi que les lois de commande associées correspondants à chaque régulateur ont

été synthétisées. Les lois de commande discontinues adoptées dépendent chacune de deux paramètres qui sont fonctions intégrantes des variations paramétriques et des perturbations externes. Toutefois, ces paramètres ne peuvent être connus qu'après une identification adéquate, ce qui constitue un inconvénient majeur. Un mauvais choix de ces paramètres peut se traduire par l'apparition du phénomène de chattering, néfaste pour le bon fonctionnement du système. Pour le réduire, il est nécessaire de faire une adaptation du gain de la commande discontinue de la logique de commutation. Dans ce sens, nous avons proposé une solution qui repose sur la variation de la valeur de la loi de commande discontinue u_n en fonction de la distance entre la variable d'état et la surface de glissement. Celle-ci consiste à encadrer la surface par une bande avec deux seuils en utilisant une forme non-linéaire plus adoucie que la fonction relais, plus précisément une forme plus ou moins exponentielle. Les résultats de simulation numérique obtenus permettent de voir que l'objectif que nous nous sommes fixé et qui consiste en la réduction du chattering est atteint, mais se fait au prix d'une difficulté dans la mise en œuvre du schéma de commande. De plus, pour avoir de bonnes performances, une connaissance précise du flux est exigée, d'où la contrainte de la nécessité d'avoir recours à un observateur de flux fiable.

Pour contourner ces restrictions, et comme la majorité des théories de la commande robustes de la machine asynchrone autre que la commande à structures variables élaborent une loi de commande à partir d'une connaissance exacte et précise des états du système et de son modèle, ce qui représente un sérieux problème quant à la détermination de la loi de commande par ces méthodes, nous avons envisagé d'intégrer dans notre processus de commande les techniques de l'intelligence artificielles entre autres: les réseaux de neurones.

Dans ce contexte, les propriétés offertes par les réseaux de neurones ont été mises à profit pour approximer les lois de commande générées par les deux régulateurs de la commande précédente, permettant ainsi de surmonter les restrictions liées à la reconstruction en temps réel la dynamique non-linéaire de forme plus ou moins exponentielle de la commande discontinue u_n proposée.

Une fois, l'architecture des réseaux de neurones adoptée, ces derniers sont entraînés hors ligne en utilisant l'algorithme de la rétro-propagation avec momentum avant d'être mis en ligne pour remplacer les régulateurs à structures variables. A cet effet, lors de cette phase d'apprentissage, les erreurs de poursuite des grandeurs commandées sont prises en compte comme indice de performance pour optimiser le fonctionnement de chaque régulateur neuronal. Les paramètres du réseau sont ajustés par le biais des règles d'adaptation non-linéaires afin de ramener les erreurs de poursuite à des valeurs infiniment petites.

Il a été signalé que, les réseaux de neurones artificiels statiques à mode d'apprentissage supervisé utilisant un ensemble d'entrées/sorties peuvent apporter quelques légères améliorations concernant la réduction du chattering. Toutefois, ils se trouvent être limités par la nécessité d'avoir un meilleur pilote "teacher" (base de données adéquates) pour obtenir un bon apprentissage, et ne peuvent être élaborés qu'après l'établissement du schéma de commande. Lors du fonctionnement et une fois entraînés, les poids de ce type de réseaux sont constants et ne font que reproduire les exemples d'apprentissage avec une puissance de généralisation limitée. Ce qui risque de ne pas leur permettre de s'adapter complètement avec les différents régimes de fonctionnement du système à commander. D'autre part, la phase d'apprentissage de ces réseaux est très lourde, ce qui nous a conduit à orienter notre travail vers

l'application d'une autre technique émergente à savoir : la commande robuste floue basée sur un modèle flou.

A cet effet, nous avons étudié le concept de la modélisation floue d'une manière générale. Par la suite, une approche de modélisation floue introduite par **Ben-Ghalia** applicable sur les systèmes incertains a été présentée. Cette approche permet de convertir une représentation analytique du système incertain en une représentation basée sur des règles linguistiques de type "If-Then". Elle offre également l'avantage d'être adapté à la conception des systèmes de commande robustes du fait qu'elle tient compte des plages de variation des paramètres internes de la machine.

Pour bénéficier des avantages que procure cette technique, et vu la complexité, la non-linéarité et les incertitudes du modèle du moteur, le fruit de notre travail de doctorat s'est concrétisé par une contribution à la modélisation floue et par l'élaboration d'une commande à structure variable floue que nous introduisons pour la première fois dans la commande des machines électriques et ce, afin de concevoir un modèle flou susceptible d'approximer le modèle réel et de fournir une commande robuste floue. A cet effet, nous avons envisagé d'adapter cette technique à notre système, sachant qu'elle consiste en une nouvelle approche de commande robuste floue basée sur un modèle flou, ayant donné des résultats très intéressants dans un cas relativement simple de commande d'un bras manipulateur. Ainsi, cette technique de modélisation floue sert à approximer le système par une représentation floue tout en tenant compte de la description analytique grossière du modèle considéré et des incertitudes qui pourraient survenir. En premier lieu, une reformulation du modèle du moteur permettant une meilleure introduction des systèmes approximatifs, en particulier des modèles flous, se trouvait être nécessaire, nous incitant à en proposer une. Cette nouvelle reformulation permet en plus de simplifier au maximum le schéma de commande, tout en gardant une forme canonique du système global. Par la suite, nous avons utilisé la technique de modélisation floue de **Ben-Ghalia** pour concevoir un modèle flou susceptible d'avoir le même comportement dynamique que le système de base. Il est important de noter que cette technique est introduite pour la première fois dans le domaine de la commande des machines électriques. Une fois le modèle flou obtenu, nous l'avons utilisé pour pouvoir concevoir un contrôleur robuste flou qui prend en charge toutes les incertitudes paramétriques et non paramétriques, sans avoir à considérer leur nature et leur répartition spectrale ; ce qui n'est pas le cas dans plusieurs stratégies de commande des systèmes probabilistes. La commande ainsi obtenue n'est pas constituée d'une combinaison entre des expressions réelles et floues, mais elle est constituée intégralement à base de règles floues. La robustesse pour ce genre de commandes est toujours assurée. D'autre part, ceci nous a permis d'éviter le calcul préalable des paramètres de la commande (par exemple le gain de la commande discontinue dans le cas de la commande à structures variables) en fonction de ces incertitudes.

L'approche de la modélisation floue offre aussi l'avantage, d'une part, de conserver les performances de la commande indépendamment de la complexité du modèle en faisant seulement un bon choix de partitionnement et des formes des ensembles flous des entrées / sorties, et d'autre part, d'assurer une simplicité d'implémentation, de rapidité de calcul et de réduire la capacité mémoire. Plusieurs points intéressants de l'approche proposée ont été notés et concrétisés par des résultats communiqués à la communauté scientifique. Les résultats de simulation effectués ont montré que l'approximation du modèle réel du moteur par le modèle flou est d'autant plus précise que le nombre des ensembles flous

correspondants aux variables d'entrée augmente. De plus, l'approche proposée permet d'améliorer les performances du moteur asynchrone en particulier et de solutionner le problème de "Chattering".

Suggestions et perspectives

La méthode proposée peut être appliquée sur une large classe de systèmes incertains. Son choix repose sur la présence d'un modèle analytique, qui peut être non-linéaire et de représentation complexe.

Le nombre d'entrées de commande, par contre, doit être le moins élevé possible. Dans le cas contraire, il est fortement recommandé de décomposer le modèle d'origine en sous-systèmes, et de réduire la base de règles du modèle flou approprié [SET95], [SET98a], [SET98b].

Dans le cas d'un système couplé, la décomposition par retour d'état permettant de le mettre sous une forme canonique de la même manière que celle présentée auparavant est suggérée.

Une fois le modèle (ou le contrôleur) flou réalisé, un réglage postérieur peut être effectué par des algorithmes d'optimisation dédiés pour les systèmes flous [ISA92], tels que la programmation évolutionnaire (en particulier les algorithmes génétiques [KIM95]) et les algorithmes adoptés dans les systèmes neuronaux (qui nécessitent de convertir le système flou en une architecture neuro-floue [JAN92]).

Concernant l'estimation des composantes du flux, des travaux de recherche ont été consacrés aux observateurs non-linéaires du moteur asynchrone. Nous citons [BEN99a] et [BEN99b] où la technique des modes de glissement permet une bonne observation des composantes du flux dans le référentiel fixe, sans nulle nécessité d'une connaissance exacte des paramètres du moteur.

Annexe

Annexe

ANNEXE A. PARAMETRES DU MOTEUR ASYNCHRONE

La simulation est effectuée sur une machine définie par les paramètres suivants :

Puissance nominale : 1,5 kW

Tension nominale : 220 / 380 V

Rendement nominal : 0,78

Facteur de puissance nominal : 0,8

Vitesse nominale : 1420 tr/min

Fréquence nominale : 50 Hz

Résistance rotorique : 3,805 Ω

Résistance statorique : 4,85 Ω

Inductance cyclique du stator : 0,274 H

Inductance cyclique du rotor : 0,274 H

Inductance mutuelle cyclique : 0,258 H

Nombre de paires de pôles : 2

Moment d'inertie : 0,031 Nm·s²/rad

Coefficient de frottement : 0,008 N·m·s/rad

$$\text{trig}_A(x; [a \ x_0]) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi(x - x_0)}{2a} \right) \right) ; \quad x_0 - 2a \leq x \leq x_0 + 2a \quad (\text{B.6})$$

– La fonction *sigmoïdale* :

$$\text{sigmf}_A(x; [a \ x_0]) = \frac{1}{1 + \exp(a(x - x_0))} \quad (\text{B.7})$$

La fonction sigmoïdale est souvent utilisée comme fonction d'activation dans les systèmes neuronaux artificiels [JAN92].

– Les fonctions *en forme de S* :

$$\text{smf}_A(x; [x_0 \ x_1]) = \begin{cases} 2 \left(\frac{x - x_0}{x_1 - x_0} \right)^2 & x_0 < x < \frac{x_0 + x_1}{2} \\ 1 - 2 \left(\frac{x_1 - x}{x_1 - x_0} \right)^2 & \frac{x_0 + x_1}{2} < x < x_1 \\ 0 & x \leq x_0 \\ 1 & x \geq x_1 \end{cases} \quad (\text{B.8})$$

où x_0 et x_1 déterminent les points d'arrêt de la fonction d'appartenance.

– Les fonctions *en forme de Z* :

$$\text{zmf}_A(x; [x_0 \ x_1]) = \begin{cases} 1 - 2 \left(\frac{x - x_0}{x_1 - x_0} \right)^2 & x_0 < x < \frac{x_0 + x_1}{2} \\ 2 \left(\frac{x_1 - x}{x_1 - x_0} \right)^2 & \frac{x_0 + x_1}{2} < x < x_1 \\ 1 & x \leq x_0 \\ 0 & x \geq x_1 \end{cases} \quad (\text{B.9})$$

– La fonction π caractérisée par le produit ou le minimum entre les fonctions en forme de s et en forme de z .

$$\pi \text{mf}_A(x; [x_{10} \ x_{11} \ x_{20} \ x_{21}]) = \text{smf}_A(x; [x_{10} \ x_{11}]) \cdot \text{zmf}_A(x; [x_{20} \ x_{21}]) \quad (\text{B.10})$$

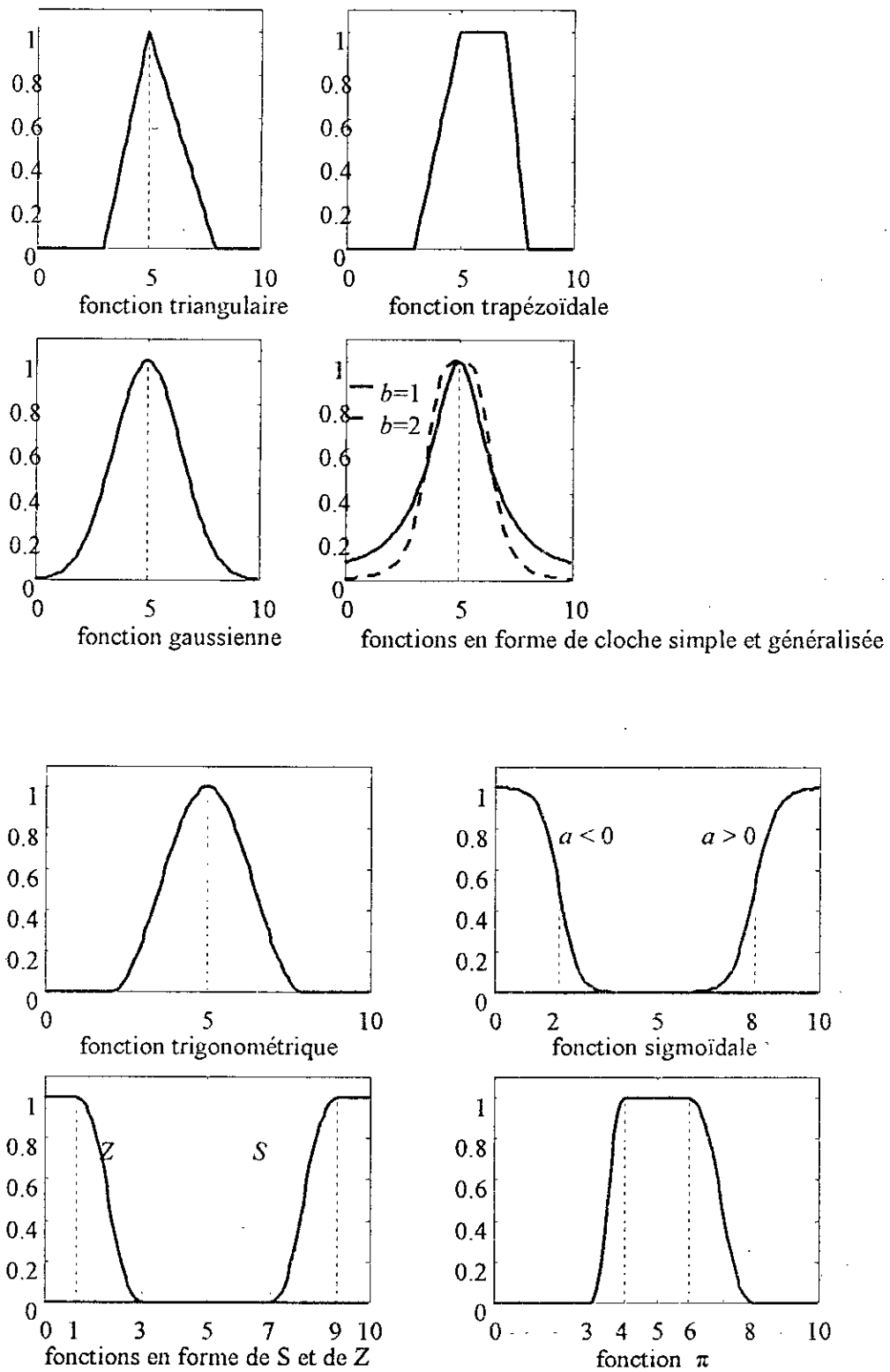


Figure B.1 : Formes usuelles des fonctions d'appartenance

II.2 Opérateurs usuels de conjonction, disjonction et de complément

Soient A et B deux ensembles flous dans X définis par leurs fonctions d'appartenance respectives $\mu_A(x)$ et $\mu_B(x)$.

– Les degrés d'appartenance de l'intersection et de l'union des ensembles flous A et B sont représentés par une opération de conjonction $T(\cdot, \cdot)$ et de disjonction $S(\cdot, \cdot)$ respectivement :

$$\begin{aligned}\mu_{A \cap B}(x) &= T(\mu_A(x), \mu_B(x)) \\ \mu_{A \cup B}(x) &= S(\mu_A(x), \mu_B(x))\end{aligned}\tag{B.11}$$

avec $T(\cdot, \cdot)$ satisfait les conditions d'une T -norme [JAN92] définies par :

$$\begin{aligned}T(0,0) &= 0 \\ T(a,1) &= T(1,a) = a \\ T(a,b) &\leq T(c,d) \text{ si } a \leq c \text{ et } b \leq d \quad (\text{monotonie}) \\ T(a,b) &= T(b,a) \quad (\text{commutativité}) \\ T(a,T(b,c)) &= T(T(a,b),c) \quad (\text{associativité})\end{aligned}\tag{B.12}$$

et $S(\cdot, \cdot)$ satisfait les conditions d'une T -conorme définies par :

$$\begin{aligned}S(1,1) &= 1 \\ S(a,0) &= S(0,a) = a \\ S(a,b) &\leq S(c,d) \text{ si } a \leq c \text{ et } b \leq d \quad (\text{monotonie}) \\ S(a,b) &= S(b,a) \quad (\text{commutativité}) \\ S(a,S(b,c)) &= S(S(a,b),c) \quad (\text{associativité})\end{aligned}\tag{B.13}$$

– Le degré d'appartenance de \bar{A} (la négation de A) est spécifié par un opérateur de négation $N(\cdot)$ satisfaisant la condition :

$$\begin{aligned}N(0) &= 1 \text{ et } N(1) = 0 \\ N(a) &\leq N(b) \text{ si } a \geq b \quad (\text{monotonie})\end{aligned}\tag{B.14}$$

Remarque :

La T-norme $T(\cdot, \cdot)$ et la T-conorme $S(\cdot, \cdot)$ doivent vérifier la généralisation de la loi de De Morgan, qui est donnée par [KEL97] :

<i>loi de De Morgan généralisée</i>	<i>Loi de De Morgan</i>	(B.15)
$S(a,b) = N[T(N(a), N(b))]$	$A \cup B = \overline{\overline{A} \cap \overline{B}}$	
$T(a,b) = N[S(N(a), N(b))]$	$A \cap B = \overline{\overline{A} \cup \overline{B}}$	

II.3 Quelques exemples des opérateurs de T-norme, de T-conorme et de complément

Le tableau B.1 montre quelques opérateurs de T-norme, de son conjoint T-conorme et du complément, utilisés dans la littérature [BOU95].

<i>Nom</i>	<i>T-norme</i> $T(a,b)$	<i>T-conorme</i> $S(a,b)$	<i>Complément</i> $N(a)$
Zadeh	$\min(a, b)$	$\text{Max}(a, b)$	$1 - a$
Probabiliste	$a \cdot b$	$a + b - a b$	$1 - a$
Lukasiewicz	$\max(a + b - 1, 0)$	$\min(a + b, 1)$	$1 - a$
Hamacher ($\gamma > 0$)	$\frac{a \cdot b}{\gamma + (1 - \gamma)(a + b - ab)}$	$\frac{a + b + a \cdot b - (1 - \gamma) \cdot ab}{1 - (1 - \gamma) \cdot ab}$	$1 - a$
Yager ($p > 0$)	$\max\left(1 - \left((1 - a)^p + (1 - b)^p\right)^{1/p}, 0\right)$	$\min\left(\left(a^p + b^p\right)^{1/p}, 1\right)$	$1 - a$
Weber ($\lambda > -1$)	$\max\left(\frac{a + b - 1 + \lambda \cdot ab}{1 + \lambda}, 0\right)$	$\min(a + b + \lambda \cdot ab, 1)$	$\frac{1 - a}{1 + \lambda a}$
Drastique	a si $b = 1$ b si $a = 1$ 0 sinon	a si $b = 0$ b si $a = 0$ 1 sinon	$1 - a$

Tableau B.1 : Exemple d'opérateurs de conjonction, disjonction et de complément
(Cf. [BOU95], p. 41)

ANNEXE C.

➤ DEMONSTRATION DU LEMME (4.1)

Soit \tilde{N} est un ensemble flou dont le support est strictement négatif, c'est à dire :

$$\forall z \in \mathbb{R}^+ : \mu_{\tilde{N}}(z) = 0 \quad (\text{C.16})$$

\tilde{e} le vecteur d'erreur flou, $\tilde{s}(\tilde{e})$ le vecteur flou de commutation défini par :

$$\tilde{s}(\tilde{e}) = \begin{pmatrix} \tilde{s}_1(\tilde{e}) \\ \dots \\ \tilde{s}_m(\tilde{e}) \end{pmatrix} \quad (\text{C.17})$$

Lemme : La condition d'atteignabilité de la dynamique correspondant à la $i^{\text{ème}}$ entrée, définie par :

$$\exists \tilde{N} \text{ tel que } \tilde{s}_i(\tilde{e}) \cdot \tilde{s}_i(\tilde{e}) \subset \tilde{N} ; i = 1, \dots, m \quad (\text{C.18})$$

est équivalente à la condition (C.19) définie par :

$$\forall s_i(e) \in \text{Supp}(\tilde{s}_i(\tilde{e})) - \{0\} \text{ et } \forall \dot{s}_i(e) \in \text{Supp}(\tilde{\dot{s}}_i(\tilde{e})) - \{0\} \quad (\text{C.19})$$

$$\text{Alors : } s_i(e) \cdot \dot{s}_i(e) < 0$$

Preuve : [GHA97]

Soit un ensemble flou $F_{s\dot{s}}$ défini par :

$$F_{s\dot{s}} = \tilde{s}_i(\tilde{e}) \tilde{\sim} \tilde{\dot{s}}_i(\tilde{e}) \quad (\text{C.20})$$

avec : $\tilde{\sim}$ l'extension de l'opérateur de multiplication sur les ensembles flous.

Le support de $F_{s\dot{s}}$ sera donc décrit par l'expression :

$$\text{supp}(F_{s\dot{s}}) = \left\{ s_i(e) \cdot \dot{s}_i(e) \in \mathbb{R} \mid s_i(e) \in \text{supp}(\tilde{s}_i(\tilde{e})) \text{ et } \dot{s}_i(e) \in \text{supp}(\tilde{\dot{s}}_i(\tilde{e})) \right\} \quad (\text{C.21})$$

Utilisant l'équation (C.20), la condition (C.18) peut être écrite par :

$$F_{s\dot{s}} \subset \tilde{N} \quad (\text{C.22})$$

Utilisant la définition de l'inclusion des ensembles flous [ZAD65], la condition (C.22) devient :

$$\forall z \in \mathbb{R} : \mu_{F_{s\dot{s}}}(z) < \mu_{\tilde{N}}(z) \quad (\text{C.23})$$

Si on prend $z_i = s_i(e) \cdot \dot{s}_i(e) \in \text{supp}(F_{s\dot{s}})$ et $z \geq 0$, les équations (C.23) et (C.16) conduit à :

$$\mu_{F_{ss}}(z_i = s_i(e) \cdot \dot{s}_i(e)) < 0 \quad (\text{C.24})$$

Ce qui est une contradiction, car les degrés d'appartenance appartiennent à l'intervalle $[0,1]$. Donc, nous pouvons conclure que :

$$\forall z_i = s_i(e) \cdot \dot{s}_i(e) \in \text{supp}(F_{ss}) : s_i(e) \cdot \dot{s}_i(e) < 0 \quad (\text{C.25})$$

D'autre part, pour que la condition (C.25) soit vérifiée, il faut qu'en plus les vecteur $s_i(e)$ et $\dot{s}_i(e)$ ne soient pas nuls.

Cependant, à partir des équations (C.21), (C.24) et (C.25), nous pouvons conclure que pour tout $s_i(e) \in \text{supp}(\tilde{s}_i(\tilde{e})) - \{0\}$ et $\dot{s}_i(e) \in \text{supp}(\tilde{\dot{s}}_i(\tilde{e})) - \{0\}$ nous avons $s_i(e) \cdot \dot{s}_i(e) < 0$.

➤ LES FONCTIONS DE CARATHEODORY

Avant de présenter les fonctions de Carathéodory, quelques définitions de base sont données au préalable [CHA81][UNI98].

C.1.1. Espace topologique

– *Espace* : ensemble muni d'une structure.

– *Espace topologique* : ensemble muni d'une structure topologique.

– *Topologie* (ou structure topologique) sur un ensemble E est un ensemble \mathfrak{G} de parties de E satisfaisant aux propriétés suivantes :

a) la réunion de toute famille d'éléments de \mathfrak{G} appartient à \mathfrak{G}

b) l'intersection de toute famille finie d'éléments de \mathfrak{G} appartient à \mathfrak{G}

En particulier, la partie vide et la partie pleine de E appartiennent à \mathfrak{G} .

– On appelle *partie ouverte* (ou un *ouvert*) de l'espace topologique E toute partie de E appartenant à \mathfrak{G} , et étant un voisinage de chacun de ses points.

– On appelle *partie fermée* (ou un *fermé*) toute partie de E dont le complémentaire est un ouvert.

– Si E est un espace discret, toute partie de E est à la fois ouverte et fermée.

C.1.2. Espace métrique

– L'*espace métrique* est un ensemble E muni d'une *distance* d [CHA81].

– On appelle *distance* sur un ensemble E une application d de $E \times E$ dans l'ensemble \mathbb{R}_+ (des nombres réels positifs ou nul) telle que, quels que soient les éléments x , y et z de E , on ait :

$$d(x, y) = 0 \Leftrightarrow x = y \quad (\text{C.26a})$$

$$d(x, y) = d(y, x) \quad (\text{C.26b})$$

$$d(x, z) \leq d(x, y) + d(y, z) \quad (\text{C.26c})$$

La dernière condition (C.26c) est appelée *inégalité triangulaire* car elle est la généralisation de l'inégalité classique entre les longueurs des côtés d'un triangle.

Tout sous-ensemble A de E est un espace métrique, dit sous-espace métrique de E pour la distance d' définie par $d'(x, y) = d(x, y)$, $\forall x, y \in A$.

Exemple :

\mathbb{R}^2 est un espace métrique muni d'une distance d qui peut prendre au moins l'une des formes suivantes :

$\forall x = (x_0, x_1)$ et $y = (y_0, y_1)$ de \mathbb{R}^2 :

$$\begin{aligned} d(x, y) &= |x_0 - y_0| + |x_1 - y_1| \\ d(x, y) &= (x_0 - y_0)^2 + (x_1 - y_1)^2 \\ d(x, y) &= [(x_0 - y_0)^2 + (x_1 - y_1)^2]^{1/2} \end{aligned} \quad (\text{C.27})$$

C.1.3. Espace topologique localement compact

– Un espace topologique E est dit *séparé* si, pour tout couple (x, y) de points distincts de E , il existe un voisinage de x et un voisinage de y disjoints. Cette condition est appelée *axiome de Hausdorff*.

– Tout espace métrique est séparé (en particulier \mathbb{R}^n).

– Un *recouvrement* d'une partie P d'un ensemble E est une famille $(P_i)_{i \in I}$ de parties de E dont la réunion contient P .

– Lorsqu'une sous-famille $(P_i)_{i \in J}$ de la famille $(P_i)_{i \in I}$ est un recouvrement de P , on dit que le recouvrement $(P_i)_{i \in J}$ est *extrait* du recouvrement $(P_i)_{i \in I}$.

– Un recouvrement $(P_i)_{i \in I}$ d'une partie P d'un espace topologique E est dit *ouvert* (resp. *fermée*) si, pour tout élément i de I , P_i est un *ouvert* (resp. *fermé*) de E .

– Un *espace topologique* E est *compact* s'il est séparé, et si de tout recouvrement ouvert de E on peut extraire un recouvrement fini.

– Pour qu'un sous-espace d'un espace compact E soit compact, il faut et il suffit qu'il soit fermé dans E .

- Un espace topologique séparé E est dit *localement compact* si tout point de E possède un voisinage compact.
- Tout espace compact est localement compact.
- Les parties ouvertes et les parties fermées d'un espace localement compact sont localement compactes.
- Le produit d'une famille finie d'espaces localement compact est localement compact.

C.1.4. Fonction mesurable

- Soit $P(E)$ ensemble des parties de E . Une *tribu* de parties d'un ensemble E est toute partie non vide de $P(E)$ tels que : la réunion de toute famille d'éléments de $P(E)$ appartient à $P(E)$, et le complément de chaque élément de $P(E)$ appartient à $P(E)$. [CHA81]

- *Espace mesurable* : couple constitué d'un ensemble E et d'une tribu $P(E)$ de parties de E . On définit dans $P(E)$ une application $m : P(E) \rightarrow \mathbb{R}_+$ (appelée *une mesure*) vérifiant la condition [RUD64] [UNI98] :

$$\forall A, B \in P(E) \text{ alors } A, B \text{ disjoints} \Rightarrow m(A \cup B) = m(A) + m(B) \quad (\text{C.28})$$

- Les parties de E appartenant à une tribu $P(E)$ sont dites des *ensembles mesurables*.

- *fonction mesurable* : Soit f une fonction à valeurs réelles définie sur un espace mesuré E . La fonction f est dite *mesurable* si l'ensemble $\{x \mid f(x) > a\}$ est mesurable pour tout réel a [RUD64].

C.1.5. Mesure au sens de Lebesgue

L'application $f \rightarrow \mu(f) = \int f$ est une mesure de Radon positive sur \mathbb{R} , appelée *mesure de Lebesgue* et notée m_x [CHA81].

On appelle mesure de Lebesgue sur \mathbb{R}^n la mesure de Radon :

$$m_x = m_{x_1} \otimes m_{x_2} \otimes \dots \otimes m_{x_n} \quad (\text{C.29})$$

où, pour tout élément i de $[1, n]$, m_{x_i} désigne la mesure de Lebesgue sur \mathbb{R} .

C.1.6. Mesure au sens de Radon

Soit : E un espace topologique localement compact dénombrable à l'infini (exemple \mathbb{R}^n), et $KC(E)$ l'espace vectoriel des fonctions à valeurs réelles continues sur E à support compact.

On appelle mesure de Radon sur E une forme linéaire $\mu : KC(E) \rightarrow \mathbb{C}$ satisfaisant à la condition suivante :

$$\begin{aligned} &\text{Pour toute fonction positive } f \text{ appartenant à } KC(\mathbb{R}), \\ &\text{il existe un nombre réel positif } M \text{ tel que,} \\ &\forall g \text{ de } KC(E) : |g| < f \Rightarrow |\mu(g)| \leq M. \end{aligned} \tag{C.30}$$

Toute forme linéaire positive sur $KC(E)$ est une mesure de Radon, dite positive.

C.1.7. Fonction fortement Carathéodory

Une fonction $f(\cdot) : U_x \times \mathbb{R} \rightarrow \mathbb{R}^n$ est une *fonction fortement Carathéodory* [GHA97] si, et seulement si, pour tout $t \in \mathbb{R}$, la fonction $f(\cdot, t)$ est continue ;
pour tout $x \in U_x$, $f(x, \cdot)$ est mesurable au sens de Lebesgue ;
et pour tout ensemble compact Σ de $U_x \times \mathbb{R}$, il existe une constante réelle C_Σ telle que, pour tout (x, t) de Σ on a :

$$\|f(x, t)\| \leq C_\Sigma \tag{C.31}$$

Bibliographie

Bibliographie

- [BAB98] R. Babuska, "Fuzzy modeling: Principles, methods and applications.", In C. Bonivento, C. Fantuzzi, R. Rovatti, "Fuzzy Logic Control : Advances in Methodology", World Scientific, Singapore, pp.187-220, 1998.
- [BAR00a] L. Barazane, Y. Sellami, C. Larbes, R. Ouiguini, "Commande hybride, linéarisante et à structures variables d'un moteur asynchrone", *First International Conference on Electrical Engineering (ICEE'2000)*, pp. 45-51, Boumerdes, Algérie, 2000.
- [BAR00b] L. Barazane, Y. Sellami, S. Boukhalfa, "Speed Controller of Field Oriented Control of an Induction Motor Using an Artificial Neural Network", *5^{ème} Colloque Africain sur le Recherche en Informatique (INRIA'2000)*, pp. 81-89, Antanarivo, Madagascar 2000.
- [BAR00c] L. Barazane, B. Hemici, M.O. Mahmoudi, Y. Amara, R. Benamar, "Étude de l'influence de la constante de temps rotorique et de la saturation sur la commande vectorielle de la machine asynchrone alimentée en courant", *Troisième Conférence Internationale sur les Mathématiques Appliquées et les Sciences de l'Ingénieur (CIMASI'2000)*, pp. 121-127, Casablanca, Maroc 2000.
- [BAR00d] L. Barazane, Y. Sellami, C. Larbes, R. Ouiguini, "Commande hybride, linéarisante et à structures variables d'un moteur asynchrone alimenté par un onduleur MLI à hystérésis des courants redressés", *Conférence Internationale sur l'Électrotechnique (ICEL'2000)*, pp. 190-195, Oran, Algérie, 2000.
- [BAR01] L. Barazane, Y. Sellami, R. Ouiguini, M.S. Boucherit, C. Larbes, "An approach to Field-Oriented Control Using a Cascade Sliding Mode Controller of an Induction Motor", *Séminaire National sur le Contrôle et les Signaux (SNCS'2001)*, pp.1-6, Djelfa, Algérie, 2001.
- [BAR02a] L. Barazane, C. Larbes, R. Ouiguini, M.S. Boucherit, "An approach to Field-Oriented Control of an Induction Motor Using Cascade Sliding Mode Controllers", *The 2nd international conference (JTEA'02)*, pp. 99-106, Sousse Nord, Tunisie, 2002.
- [BAR02b] L. Barazane, Y. Sellami, R. Ouiguini, M.S. Boucherit, "Fuzzy Sliding Mode Controllers to Reduce the Chattering in Vector Controlled Induction Motor Drives", *4th Conférence internationale sur les Mathématiques Appliquées et les Sciences de l'Ingénieur, (CIMASI'2002)*, pp.65-70, Casablanca, Maroc, 2002.
- [BAR03a] L. Barazane, Y. Sellami, R. Ouiguini, C. Larbes, M.S. Boucherit, "A New Fuzzy Model Representation of Induction Motor", *2nd Intern. Conf. on Signals, Systems, Decision and Information Technology (SSD'03)*, pp.98:1-6, Sousse, Tunisie, 2003.
- [BAR03b] L. Barazane, C. Larbes, R. Ouiguini, M.S. Boucherit, "Robust Hybrid Control Based on Feedback Linearization and Field Oriented Control of an Induction Motor", *Archives of electrical engineering*, N°2, Juillet 2003 (à paraître).

- [BAR86] A. Ba-Razzouk, A. Cheriti, G. Olivier, "Artificial neural networks rotor time constant adaptation indirect field oriented control drives", *IEEE PESC*, Baveno, pp 701-707, 1986.
- [BAR91] L. Barazane, B. Hemici, "Commande d'un moteur Asynchrone à flux orienté par onduleur de courant", *Journées d'Étude sur l'Électronique de Puissance et la Commande des Machines Électriques (JEPCM'91)*, pp. 121-126, École Nationale Polytechnique, Alger, Algérie, 1991.
- [BAR93] L. Barazane, "Commande vectorielle d'une machine asynchrone alimentée en courant", *Thèse de Magister, ENP, Alger*, 1993.
- [BAR97] A. Ba-Razzouk, A. Cheriti, G. Olivier, P. Sicard, "Field-oriented control induction motors using neural networks decouplers", *IEEE Trans. Power Electronics.*, vol.12, N°4, pp.752-763, 1997.
- [BAR98] G. Bartolini, M. Marchesoni, P. Pisu, E. Usai, "Chattering reduction and robust position control in induction motor with second-order VSS". *Int. Journal of Systems Science*, vol.29, n°1, pp.1-12, 1998.
- [BAR99] L. Barazane, B. Hemici, S. Boukhalfa, "Feed-Forward Field-Orientation Control of an Induction Motor Using an Artificial Neural Network", *Conférence sur les réseaux de neurones, la logique floue, les algorithmes génétiques et leurs applications (CSCA '99)*, pp. 160-164, Alger, Algérie, 1999.
- [BEL86] A. Bellini, G. Figalli, G. Ulivi, "A microcomputer based optimal control system to reduce the effects of the parameter variations and speed measurements errors in induction motor drives", *IEEE Trans. Ind. Applict*, vol.IA-2, N°.1, pp.160-173, 1986.
- [BEN99a] A. Benchaib, A. Rachid, E. Audrezet, M. Tadjine, "Real-time sliding-mode observer and control of an induction motor", *IEEE Trans. Ind. Electronics.*, vol.46, n°1, pp.128-138, Feb. 1999.
- [BEN99b] A. Benchaib, M. Tadjine, A. Rachid, "Sliding mode control of an induction motor with unknown load: application on a DSP-based system", *Int. Journal of Systems Science*, vol.30, n°8, pp.849-863, 1999.
- [BLA71] F. Blaschke, "A new method for the structure decoupling of AC induction machines", *Proc. 2nd IFAC Symp. Multivariable Techn. Control Systems, Part 3*, pp.11-13, 1971.
- [BLA72] F. Blaschke, "The principle of field orientation as applied to the new transvector", *Siemens review*, vol. 34, pp.217-222, 1972.
- [BOD94] M. Bodson, J. Chiason, R. Novotnak, "High performance of induction motor control via input-output linearization", *IEEE Control System*, pp.25-33, 1994.
- [BOR96] P. Bortolet, S. Boverie, A. Titli, "Modélisation floue et contrôle de systèmes non linéaires", *LFA '96-Nancy (France)*, pp.106-133, 1996.
- [BOU95] B. Bouchon-Meunier, "La logique floue et ses applications", *Ed. Addison-Wesley*, France, 1995.
- [BRA96] P.J. Costa Branco, N. Lori, J.A. Dente, "New approaches on structure identification of fuzzy models: case study in an electro-mechanical system", in "Fuzzy Logic, Neural Networks, and Evolutionary Computation", Eds. T. Furuhashi and Y. Uchikawa, pp. 104-143, Springer-Verlag, Berlin, 1996.
- [BRE99] S. Breton, "Une Approche neuronale du contrôle robotique utilisant la vision binoéculaire par reconstruction tridimensionnelle", *Thèse de Doctorat, Univ. De Haute Alsace*, 1999.

- [CAB97] L.A. Cabrera, M.E. Elbuluk, I. Hussain, "Tuning the stator resistance of induction motors using artificial neural networks", *IEEE Trans. Power Electro.*, vol.12, N°5, pp.78-89, 1997.
- [CAR88] R.A. DeCarlo, S.H. Zak, G.P. Matthews, "Variable structure control of nonlinear multivariable systems: A tutorial", *Proceedings of the IEEE*, vol.76, N°3, pp.212-232, 1988.
- [CHA81] L. Chambadal, "Dictionnaire de mathématiques", Hachette, Paris, France, 1981.
- [CHA83] J. Chatelin, "Machines Electriques", *Tomel, Dunod presses Polytechniques Romandes*, 1983.
- [CHA90] C.C. Chan, W.S. Leung, C.W. NG, "Adapting decoupling control of induction motor drives", *IEEE Trans. Ind. Electronics*, vol.37, N°1, pp.81-96, 1990.
- [CHU95a] S.K. Chung, J.H. Lee, J.S. Ko, M.J. Youn, "Current control of voltage-fed PWM inverter for AC machine integral variable structure control", *Proc. of the 1995 IEEE IECON 21st Inter. Conf. on Ind. Electronics*, vol.1, pp.668-673, 1995.
- [CHU95b] S.K. Chung, J.H. Lee, J.S. Ko, M.J. Youn, "Robust speed control of brushless direct-drive motor using integral variable structure control", *IEE Proc. Electric Power Applications*, vol.142-6, pp.361-370, 1995.
- [COS90] R.R. Costa, L. Hsu, "Robustness of VS-MRAC with respect to unmodeled dynamics and external disturbances", *Proceedings of the 29th on Decision and Control Honolulu*, pp.3208-3213, 1990
- [DAV93] E. Davalo, P. Naim, "Des Réseaux de Neurones", *Edition Eyrolles*, 1993.
- [DEC88] R.A. De Carlo, S.H. Zak, G.P. Matthews, "Variable structure control of nonlinear multivariable systems: a tutorial", *Proc. IEEE*, vol.76, n°3, pp.212-232, 1988.
- [DEL87] A.De-Luca, G. Ulivi, "Full linearisation of induction motors via non-linear state-feedback", *In Proc. 26th Conf. Decision and Control*, Los Angeles, pp.1630-1645, 1987.
- [DEL89] A.De-Luca, G. Ulivi, "Dynamic decoupling of voltage frequency controlled induction motors", *Lecture notes in Control and Information sciences of 8th Int. Conf. Analysis and optimisation of systems*, pp.16-23, France, 1989.
- [EME62] T. Emelyanov, "Sur une classe de systèmes de régulation automatique à structure variable", *Journal de l'académie des sciences d'URSS, Énergétique et automatique* N°3, 1962.
- [FIL60] A. Filippov, "Equations différentielles à second membre discontinu", *Journal de mathématiques*, Tome 51, n°1, 1960.
- [FU91] Y. Fu, "Commandes découplées et adaptatives de machines asynchrones 3~", *Thèse de doctorat, Univ. Montpellier II*, 1991.
- [FUN89] K. Funahashi, "On the approximate realization of continuous mapping by neural networks", *Neural networks*, vol.2, pp.157-165, 1989.
- [GAO93] W. GAO, J.C. Hung, "Variable structure control of nonlinear systems: A new approach", *IEEE Trans. Ind. Electronics*, vol.40, N°1, pp.45-55, 1993.
- [GHA94] M. Ben-Ghali, A.T. Alouani, "Fuzzy modelling of dynamic systems", *Proc. of the 26th Southern Symposium on System Theory*, pp.168-171, 1994.
- [GHA95a] M. Ben-Ghali, A.T. Alouani, "Knowledge-base robust control of complex systems", *Intel. Systems for the 21st Century, IEEE Inter. Conf. on Syst. Man. & Cyb.*, vol.2, pp.977-982, 1995.

- [GHA95b] M. Ben-Ghalia, A.T. Alouani, "A fuzzy variable structure approach to feedback regulation of uncertain dynamical systems, with application to robotics", *Information Sciences*, Elsevier Science, vol.85, pp.241-273, 1995.
- [GHA96a] M. Ben-Ghalia, "Nonlinear modeling of uncertain dynamical systems using fuzzy set concepts", *Proc. of the 35th IEEE Conf. on Decision and Control*, Kobe, Japan, pp.418-423, 1996.
- [GHA96b] M. Ben-Ghalia, "Robust model-based control of uncertain dynamical systems : a fuzzy set theory based approach", *Proc. of the 35th Conf. on Decision and Control*, vol.1, Kobe, Japan, pp.807-812, 1996.
- [GHA96c] M. Ben-Ghalia, "Aspects of the effects of the defuzzification stage on the stability and performance of fuzzy model-based control systems", *Proc. of the 5th Int. Conf. on Fuzzy Systems*, vol.1, pp.475-478, 1996.
- [GHA97] M. Ben-Ghalia, "Modelling and robust control of uncertain dynamical systems using fuzzy set theory", *Int. J. Control*, vol.68, n°6, pp. 1367-1395, 1997.
- [HAL69] J.L. Hale, "Ordinary differential equations", Wiley-Interscience, New York, 1969.
- [HAR86] F. Harashima, H. Hashimoto, K. Maruyama, "Practical robust control of robot arm using variable structure systems", *Proc. of IEEE, Int. Conf. on robotics and automation*, San Francisco, pp.532-538, 1986.
- [HAS85] H. Hashimoto, T. Ichiyama, "Tracking control of robot manipulator using sliding mode", *Int Symposium on Ind. Robot*, pp.11-13, Tokyo, 1985.
- [HAY94] S. Haykin, "Neural Networks", *A comprehensive Foundation*, MC Millan College publishing, 1994.
- [HEC90] R. Hecht-Nielsen, "Neurocomputing", *Addition Wesley*, 1990.
- [HER91] J. Hertz, A. Krogh, R.G. Palmer, "Introduction to the theory of Neural computation", *Santa Fe Institute Addition Wesley*, 1991.
- [HER94] J. Héroult, C. Jutten, "Réseaux neuronaux et traitement du signal", *Edition Hermès*, 1994.
- [HO95] E. HO, P.C. Sen, "High performance decoupling control techniques for various rotating field machines", *IEEE Trans. Ind. Electronics*, vol.42, N°1, pp.40-49, 1995.
- [HOL94] J. Holtz, "The Induction motor- A dynamic system", *IECON'94*, pp.1-6, Italie, 1994.
- [HSU89] L. Hsu, R.R. Costa, "Variable structure model reference adaptive control using only input and output measurements", *Part1 Int Journal of Control*, vol.49, N°2, pp.419-437, 1989.
- [HUN91] J.Y. Hung, R.M. Nelms, "Using a boundary layer technique to reduce chatter in sliding mode controllers", *Proc.6th IEEE Applied Power Conference*, pp.195-201, 1991.
- [IOA83] P.A. Ioannou, P.V. Kokotovic, "Adaptive systems with reduced models", *Springer-verlag*, New-York, 1983.
- [ISA92] S. Isaka, A.V. Sebald, "An optimization approach for fuzzy controller design", *IEEE Trans. Syst. Man & Cyb.*, vol.22, n°6, pp.1469-1473, 1992.
- [ISA92] S. Isaka, A.V. Sebald, "An optimization approach for fuzzy controller design", *IEEE Trans. Syst. Man & Cyb.*, vol.22, n°6, pp.1469-1473, Nov./Dec.1992.

- [ISH93] A. Ishigame, T. Furukawa, S. Kawamoto, T. Taniguchi, "Sliding mode controller design based on fuzzy inference for nonlinear systems", *IEEE Trans. Ind. Electronics*, vol.40, n°1, pp.64-70, 1993.
- [ISI89] A. Isidori, "Nonlinear Control Systems - An Introduction", *Springer Verlag*, Berlin, 1989.
- [IWA93] M. Iwasaki, N. Matsui, "Robust speed control of induction motor with torque feedforward control", *IEEE Trans. Ind. Electronics*, vol.40, N°6, pp.553-560, 1993.
- [JAG96] S. Jagannathan, F.L. Lewis, "Discrete- Time neural net controller for a class of nonlinear dynamical systems", *IEEE Trans. Automatic Control*, vol.41, N°11, pp.1693-1699, 1996.
- [JAN92] J.S.R. Jang, "Neuro-Fuzzy Modeling: Architectures, Analyses, and Applications", *Ph.D. Dissertation, EECS Department, Univ. of California at Berkeley*, 1992.
- [JAN95] J.S.R. Jang, C-T. Sun, "Neuro-Fuzzy Modeling and Control", *The Proc. of the IEEE*, vol. 83, pp. 378-406, 1995.
- [JOD94] J.F. Jodouin, C. Jutten, "Les réseaux neuromimétiques", *Collection Informatique, Edition Hermès*, 1994.
- [KAN91] I. Kanellakopoulos, P.V. Kokotovic, A.S. Morse, "Systematic design of adaptive controllers for feedback linearisable systems.", *IEEE Trans. Automatic Control*, vol.36, N°6, pp.1076-1088, 1991.
- [KAZ96] M.P. Kazmierkowski, D.L. Sobczuk, "Sliding mode feedback linearized control of PWM inverter fed induction motor", *IEEE IECON'96*, vol.1, pp.244-249, 1996.
- [KEL97] W.E. Kelly, "Dimensionality in fuzzy systems", *Dissertation, Texas A&M university*, Aug. 1997.
- [KHA96] H.K. Khalil, "Nonlinear systems", *Prentice Hall*, 1996.
- [KIM95] M.H. Kim, J.C. Hung, "Vector Control System of induction motor without speed sensor at very low speed", *IECON'95 Proc. on Ind. Electronics Control and Instrumentation*, 1995.
- [LAL99] D. Lalili, "Sur les commandes non-linéaires adaptatives de la machine asynchrone". *Thèse de Magister, ENP, Alger*, 1999.
- [LEE90] C.C. Lee, "Fuzzy logic in control systems: fuzzy logic controller-part 1 (and part 2)", *IEEE Trans. Systems, Man, and Cybernetics*, vol.20, n°2, pp.404-418, 1990.
- [LEO85] W. Leonhard, "Control of electrical drives", *Springer-Verlag*, New York, 1985.
- [LIA88] C.M. Liaw, C.T. Pan, Y.C. Chen, "An adaptive controller for current-fed induction motors", *IEEE Trans. On Aerosp. Elec. Syst.*, vol.24, N°3, pp.231-242, 1988.
- [LIA94] C.M. Liaw, F.J. Lin "A robust speed controller for induction motor drives", *IEEE Trans. Ind. Electronics*, vol.41, N°3, pp.308-315, 1994.
- [LIN93] F.J. Lin, C.M. Liaw, "Control of indirect field-oriented induction motor drives considering the effects dead time and parameter variations", *IEEE Trans. Ind. Electronics*, vol.40, N°5, pp.486-495, 1993.
- [LIN94] F.J. Lin, R.J. Wai, "Hybrid controller using a neural network for a PM synchronous servo-motor drive", *IEEE Trans. Automatic Control*, vol.39, N°6, pp.15-23, 1994.
- [LU98] Y. Lu, T.Q. Chen, "Fuzzy diagnostic model and its application in automotive engineering diagnosis", *Applied Intelligence*, Kluwer Academic Publishers, vol.9, pp.231-243, 1998.

- [MAH99] M.O. Mahmoudi, N. Madani, N.F. Benkhoris, F. Boudjema, "Cascade sliding mode control of field-oriented induction machine drive", *Eur. Phys. J. AP7*, pp.217-225, 1999.
- [MAM75] E.H. Mamdani, S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *Int. J. of Man-Machine Studies*, vol.7, n°1, pp.1-13, 1975.
- [MAN99] M. Männle, "Identifying Rule-Base TSK fuzzy models", *Proc. EUFIT*, Aachen, Germany, pp.286-299, 1999.
- [MAR91] R. Marino and P. Valigi, "Nonlinear control of induction motors : a simulation study", *In European Control Conference*, Grenoble, France, pp.1057-1062, 1991.
- [MAT95] R. Mathew, D. Houghton, W. Oghanna "Vector control techniques for induction motors", *IEEE International Conference on power Electronics and drive systems*, vol.2, pp.813-818, 1995.
- [MAT99] "Fuzzy logic Toolbox for use with Matlab. User's guide", Version 2, *The Math Works*, 1999.
- [NAR90] K.S. Narendra, K.Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. Neural networks*, vol.1, pp 4-27, 1990.
- [NAR90] K.S. Narendra, K. Pathasarathy, " Identification and control of dynamical systems using neural networks", *IEEE Trans. Neural networks*, vol.1, N°1, pp.503-509, 1990.
- [PAS87] D. Psalti, A. Sideris, A. Yamamura, "Neural controllers", *proceedings of 1st International conference on neural networks*, vol 4, pp.551-558, San Diego USA, 1987.
- [PED97] W. Pedrycz, M. Reformat, "Rule-Based Modeling of Nonlinear Relationships", *IEEE Trans. Fuzzy Syst.*, vol.5, n°2, 1997.
- [RAC96] A. Rachid, "Systèmes de régulation", *Edition Masson*, New york, 1996.
- [RUM86] D.E. Rumelhart, G.E. Hinton., R.J. Williams, "Learning internal representation by error propagation", *Parallel Distributed Processing*, vol 1, Chap.8, 1986.
- [SAB81] A. Sabanovic, D.B. Izosimov, "Application of sliding mode to induction motor control", *IEEE Trans Ind. Application*, vol IA 17, n°1, pp.41-49, 1981.
- [SAB89] A. Sabanovic, F. Bilaloviv, "Sliding mode control of AC drives", *IEEE Trans Ind. Application*, vol 25, n°1, pp.70-75, 1989.
- [SAS89] S.S. Sastry, A. Isisori, "Adaptive control of linearizable systems", *IEEE Trans. Automatic and control*, vol.34, N°11, pp.1123-1131, 1989.
- [SET95] M. Setnes, "Fuzzy rule-base simplification using similarity measures", *M.Sc. Thesis, Delft University of Technology*, Delft, Netherlands, 1995.
- [SET95] M. Setnes, "Fuzzy rule-base simplification using similarity measures", *M.Sc. Thesis, Delft University of Technology*, Delft, Netherlands, July 1995.
- [SET98a] M. Setnes, R. Babusuka, U. Kaymak, H.R. Nauta Lemke, "Similarity measures in fuzzy rule base simplification", *IEEE Trans. Syst. Man. & Cyb., Part-B*, vol.28, n°3, pp.376-386, 1998.
- [SET98a] M. Setnes, R. Babusuka, U. Kaymak, H.R. Nauta Lemke, "Similarity measures in fuzzy rule base simplification", *IEEE Trans. Syst. Man. & Cyb., Part-B*, vol.28, n°3, pp.376-386, June 1998.
- [SET98b] M. Setnes, R. Babuska, H.B. Verbruggen, "Complexity reduction in fuzzy modeling", *Mathematics & Computers in Simulation*, vol.46, pp.507-516, 1998.

- [SET98b] M. Setnes, R. Babuska, H.B. Verbruggen, "Complexity reduction in fuzzy modeling", *Mathematics & Computers in Simulation*, vol.46, pp.507-516, 1998.
- [SIM95] M.G. Simões, B.K. Bose, "Neural networks based estimation of feedback signals for a vector controlled induction motor drive", *IEEE Trans. Industry Application*, vol.31, N°3, pp.630-639, 1995.
- [SIR89] H. Sira Ramirez, "Nonlinear variable structure systems in sliding mode: The general case", *IEEE Trans. Automatic and Control*, vol.34, N°11, pp.1186-1188, 1989.
- [SLO84] J.J. Slotine, "Sliding controller design for nonlinear systems", *IGC*, vol.40, n°2, pp.421-434, 1984.
- [SLO91] J.J. Slotine, W. Li, "Applied nonlinear control", *Englewood Cliffs, NJ: Prentice Hall*, 1991.
- [SOB96] M.P. Kazmierkowski, D.L. Sobczuk, "Sliding mode feedback linearized control of PWM inverter fed induction motor". *IEEE IECON, 22nd Int. Conf. on Ind. Electronics, Control and Instrumentation*, vol.1, pp.244-249, 1996.
- [SON98] Q. Song, "Robust training algorithm of multi-layered neural networks for identification of non-linear dynamic system", *IEE Proc. Control Theory Application*, vol.145, N°1, pp.52-67, 1998.
- [SUG88] M. Sugeno, G.T. Kang, "Structure identification of fuzzy model", *Fuzzy Sets and Systems*, North Holland, vol.28, pp.15-33, 1988.
- [SUG93] M. Sugeno, T. Yasukawa, "A fuzzy logic based approach to qualitative modeling", *IEEE Trans. Fuzzy Syst.*, vol.1, pp.7-31, Feb.1993.
- [TAK85] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", *IEEE Trans. Syst., Man, and Cybernetics*, vol.15, pp. 116-132, 1985.
- [TAY93] J.G. Taylor, "The promise of neural networks", *Springer-Verlag*, New York, 1983.
- [TLE99] A. Tlemçani, "Sur la commande adaptative par les systèmes flous: application à la machine synchrone à aimants permanents", *Thèse de Magister, ENP, Alger*, 1999.
- [UNI98] "Encyclopédie universalis", 1998.
- [UTK77] V.I. Utkin, "Variable structure systems with sliding mode", *IEEE Trans. Automatic Control*, vol.ac-22, N°2, pp.212-222, 1977.
- [UTK92] V.I. Utkin, "Sliding mode control and optimization", *Springer-Verlag*, Berlin, 1992.
- [UTK93] V.I. Utkin, "Sliding mode control design principles and applications to electric drives", *IEEE Trans. Ind. Electronics*, vol 40, n°1, pp.23-36, 1993.
- [UTK99] V.I. Utkin, A. Sabanovic, "Sliding modes applications in power electronics and motion control systems", *IEEE Conf., ISIE'99*, Bled, Slovenia, pp.TU22-TU31, 1999.
- [YAS95] A. Yasildirek, F.L. Lewis, "Feedback linearization using neural networks", *Automatica*, vol.31, N°11, pp.1659-1664, 1995.
- [YOU78] K.-K.D. Young, "Controller design for a manipulator using theory for variable structure systems", *IEEE Trans. Syst. Man. Cybernetics*, vol.8, pp.210-218, 1978.
- [YOU96] K.D. Young, V.I. Utkin, U.Ozguner, "A control engineers guide to sliding mode control", *IEEE Workshop on variable structure systems*, pp.1-14, 1996
- [YOU98] T. Youcef, "Commande à structure variable généralisée appliquée à un bras manipulateur", *Thèse de Magister, INHC Boumerdes, Algérie*, 1998.

- [YOU99] K.D. Young, V.I. Utkin, Ü. Özgüner, "A control engineer's guide to sliding mode control", *IEEE Trans. Control Systems Technology*, vol.7, n°3, 1999.
- [ZAD65] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, vol.8, pp.338-353, 1965.
- [ZHA99] T. Zhan, S.S. GE, C.C. Hang, "Design and performance analysis of a direct adaptive controller for nonlinear systems", *Automatica*, vol.35, N°5, pp.1809-1817, 1999.

Publications Soumises

- L.Barazane, Y.Sellami, R.Ouiguini, C.Larbes, M.S.Boucherit, E.M. Berkouk, "Cascade Fuzzy Variable Structure Control of Induction Motor Based on the Approach of Fuzzy Modelling", *International Journal of Systems Science (IJSS)*.
- L.Barazane, Y.Sellami, R.Ouiguini, C. Larbes, M.S.Boucherit, E.M. Berkouk, "New Approach of Linguistic Fuzzy Modelling of Induction Motor", *Archives of electrical engineering*.
- L.Barazane, Y.Sellami, R.Ouiguini, M.S.Boucherit, E.M. Berkouk, "An Approach to the Fuzzy Variable Structure Control of Induction Motor", *ELECTROMOTION*.

ملخص:

- أحد وجهتنا عملنا حول تقنيات التحكم الحديثة وهي تلك التي تركز على شبكات الخلايا العصبية الاصطناعية والأتمتاج الغامضة:
- أتمتج تجاوز بعض مشاكل التحكم في المحرك اللامتزامن، مثل ربط متغيرات الوضعية، الديناميكيات، مرور الخدلية، الإزديلات الوسيطية وغير الوسيطية المتعلقة بالنظام.
 - بهدف تقليص ظاهرة "الشاترينغ" (chattering) في عملية التحكم ذات الهياكل المتغيرة.

وبالتالي تجسدت نتيجة جهدنا في هذا العمل من خلال مساهمتين هما:

1. مخطط تحكم موثوق قائم على شبكات الخلايا العصبية الاصطناعية، موجه لتقريب العبارات غير الخطية لقانون التحكم غير المستمر المقترح، والمنجز مسبقاً بمعدلات الهياكل المتغيرة لعملية التحكم المختلط الواقع عليه الاختيار.
 2. من جهة أخرى، ومن أجل تقادي بعض حدود التقنية السالفة، وجهنا عملنا نحو التشكيل الغامض للتملاج بغرض إعداد تحكم جديد ذات هياكل متغيرة غامضة تركز على نموذج غامض للمحرك اللامتزامن، والذي تم تطبيقه للمرة الأولى في تحكم المحركات الكهربائية. تسمح هذه التقنية بمحاكاة النظام عن طريق تمثيل غامض أخذاً بعين الإعتبار الوصف التحازلي العام وإزديلات النظام.
- المفاتيح: محرك لامتزامن، تحكم شعاعي، تحكم تخطيطي، تحكم مختلط، تحكم ذات هياكل متغيرة، مُدترينغ، تحكم موثوق مختلط بشبكات الخلايا العصبية الاصطناعية، تشكيل غامض للتملاج، تحكم غامض ذات هياكل متغيرة قائم على نموذج غامض.

Résumé :

Nous avons dirigé notre travail sur les techniques de commande émergentes à savoir celles basées sur les réseaux de neurones artificiels (RNA) et les modèles flous :

- Dans un effort visant à résoudre certains problèmes de commande du moteur asynchrone, tels que le couplage des variables d'état, les dynamiques non-linéaires, les incertitudes paramétriques et non paramétriques inhérents au système,
- et dans le but de réduire le phénomène de chattering dans le processus de commande à structures variables.

Ainsi, le fruit de notre travail de doctorat s'est concrétisé par deux contributions:

1. Un schéma de commande robuste basé sur les RNAs destinés à approximer les termes non-linéaires de la loi de commande discontinue proposée et qui est réalisée au préalable par les régulateurs à structures variables du processus de commande hybride adoptée.
2. D'autre part et pour contourner certaines limitations causées par la technique précédente, nous avons orienter notre travail vers la modélisation floue dans le but d'élaborer une nouvelle commande à structure variable floue basée sur un modèle flou de la machine asynchrone et qui est appliquée pour la première fois dans la commande des machines électriques. Cette technique sert à imiter le système par une représentation floue tout en tenant compte de la description analytique grossière et des incertitudes du système.

Mots-clés : Moteur asynchrone, control vectoriel, commande linéarisante, commande hybride, commande à structures variables, chattering, commande robuste hybrid par réseaux de neurones artificiels, modélisation floue, commande à structures variables floue basée sur un model flou.

Abstract :

This research work is based on the application of artificial neural networks (ANRs) and fuzzy models:

- In order to solve some control problems of the asynchronous motor such as, state variables coupling, dynamic nonlinearities, parametric and nonparametric uncertainties inherent to the system.
- And in the objective to reduce the chattering phenomenon in variable structure control system.

So, the result of this reaserch consists on:

1. a new robust control scheme based on ANRs aimed to approximate the nonlinear terms of the proposed discontinuous control law which were previously implemented by variable structure regulators in the adopted hybrid control.
2. in the other hand, and in order to overcome the limitations of the previous technique, fuzzy modelling was proposed to elaborate a new approach to fuzzy variable structure control based in this case on the fuzzy model of the asynchronous motor which is applied for the first time in electrical motors control. This technique help to imitate the system model by a linguistic fuzzy mle base representation, with taking into account the mathematical description and the boundary uncertainties of the system.

Key words : Induction motor, vector control, feedback linearisation control, hybrid control, variable structure control, chattering, robust hybrid control based on artificial neural networks, fuzzy modelling, fuzzy variable structure control based on fuzzy model.