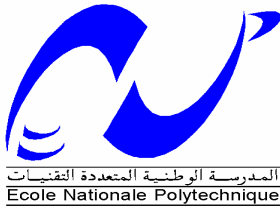


Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département de Génie Electrique
Laboratoire de Commande des Processus



THESE

présentée

en vue de l'obtention du titre de

Docteur d'Etat

En

AUTOMATIQUE

Par

NAIT SEGHIR Amirouche

Magister en Automatique de l'ENP

Intitulée :

CONTRIBUTION A LA COMMANDE ADAPTATIVE ET NEURONALE D'UNE MACHINE SYNCHROME A AIMANTS PERMANENTS

Soutenue publiquement le 16/12/2007 devant le jury composé de:

F. BOUDJEMA	Professeur à l'ENP	Président
M.S. BOUCHERIT	Professeur à l'ENP	Rapporteur
D. BOUKHETALA	Maître de conférences à l'E.N.P	Examineur
L. NEZLI	Maître de conférences à l'E.N.P	Examineur
L. BARAZANE	Maître de conférences à l'USTHB	Examineur
M.E. SIFI	Chargé de Mission audit M.T. SNTF	Invité

CONTRIBUTION A LA COMMANDE ADAPTATIVE ET NEURONALE D'UNE MACHINE SYNCHRONE A AIMANTS PERMANENTS

RESUME

ملخص — في هذا العمل قدمنا تصميم و تطبيق للتحكم التلاؤمي و التحكم بتقنيات الشبكات العصبية الاصطناعية على محرك تزامني ذو مغناطيس دائم. إن قدرات المعدلات الخطية محدودة نتيجة تأثرها بالحالة اللاخطية للمحرك و بتغير المعاملات. استعمل لهذا الغرض المنظم الذاتي الضبط بالطريقة المباشرة و المستعمل لطريقة التعرف عن العوامل. كذلك حلت لهذا المنظم طريقة مبنية على إستراتيجية التحكم بالخطأ التوقعي الأدنى المتزن و أخرى على إستراتيجية الانحراف الأدنى المعمم. في هذه الدراسة، قمنا باستعمال المنظم الذاتي الضبط في مكان المعدل الخطي للسرعة المستعمل في مخطط تقنية الحقل الموجه للمحرك. بعد ذلك قدمنا في الجزء الثاني لهذا العمل، دراسة عن تقنية التحكم بالشبكات العصبية الاصطناعية لسرعة المحرك التزامني ذو مغناطيس دائم. بما أن النموذج الكهربائي للماكينة كثير التغير، اقترحنا تصميم لمخطط جديد للمنظم بتقنية الشبكات العصبية كثير المتغير. نتائج المحاكاة أثبتت ناجعة و صلابة مخططات التحكم المقترحة رغم وجود مختلف إشارات التشويش و التغيرات الطارئة على عوامل الماكينة.

كلمات مفاتيح —، للتحكم التلاؤمي، المنظم الذاتي الضبط، الشبكات العصبية الاصطناعية، محرك تزامني ذو مغناطيس دائم.

Résumé— Dans ce travail, l'étude et l'application de la commande adaptative et la commande par réseau de neurones artificiel (RNA) à la machine synchrone à aimants permanents (MSAP), ont été abordées. Les performances des régulateurs linéaires sont limitées à cause de leur sensibilité aux non linéarités et aux variations paramétriques. Alors, la commande auto-ajustable directe utilisant l'algorithme des moindres carrés récursifs est adoptée. Les stratégies de commande à erreur de prédiction minimale pondérée (EPMP) et à variance minimale généralisée sont considérées. Dans cette étude, nous appliquons la commande adaptative à la place du régulateur PI de la boucle de vitesse du schéma de la commande vectorielle de la MSAP.

La commande en vitesse de la MSAP utilisant les réseaux de neurones artificiels est étudiée dans la deuxième partie de ce travail. Comme le modèle électrique de la machine est non linéaire à deux entrées et deux sorties nécessitant un régulateur multivariable (MIMO), un nouveau régulateur neuronale multi sortie est proposé.

Les résultats obtenus permettent d'illustrer, tant au niveau des performances que de la robustesse, l'apport des commandes proposées dans les entraînements électriques avec machine synchrone à aimants.

Mots Clés — Commande adaptative directe, régulateur auto-ajustable, réseaux de neurones, Machine synchrone à aimants permanents.

Abstract — In this work, the design and the application of adaptive control and Artificial Neural Networks (ANN) control of Permanent Magnet synchronous Motors (PMSM) are presented. The performances derived by the linear controllers are usually limited due to their sensitivity to non linear behaviour of PMSM dynamics, load disturbance, and parameter uncertainties. So a direct self-tuning control strategy based on a recursive least square identification Algorithm is adopted. The weighted minimum prediction error (WMPE) and the generalized minimum variance control techniques are considered. In this study we apply adaptive regulator in place of rotor speed PI controller of the field-oriented control scheme of the PMSM.

ANN for speed control design of PMSM is presented in the second part of the work. Because of the electrical model of motor is a nonlinear two inputs two outputs system which needs a multivariable current controller, a new multi-input multi-output (MIMO) neural network controller is proposed.

The results obtained show the efficiency of the proposed controllers to ensure robustness against load, parameters variations and to achieve the required performances under the various conditions.

Keywords — Direct adaptive control, Self-tuning control, Neural Networks, Permanent Magnet synchronous Motors.

*À ma défunte mère
A mon père
A mes frères et sœurs
A ma femme Azizen
A ma petite Lina Taous Tarwihthe*

AVANT PROPOS

Les travaux de recherche exposés dans cette thèse ont été menés au sein du Laboratoire de Commande des Processus, de l'Ecole Nationale Polytechnique, et au Laboratoire des Systèmes Electriques et Industriels, de la Faculté d'Electronique et Informatique de l'Université des Sciences et de la Technologie Houari Boumediene.

*Je voudrai en premier lieu remercier Monsieur **M. S. BOUCHERIT**, Professeur à l'Ecole Nationale Polytechnique qui a bien voulu diriger les travaux de cette thèse. Son concours éclairé et cordial, son accueil chaleureux, sa patience ainsi que la confiance qu'il m'a témoignée sont inestimables. Qu'il trouve ici l'expression de ma profonde gratitude.*

*Je mesure tout l'honneur qu'a bien voulu me faire Monsieur **F. BOUDJEMAA**, Professeur à l'Ecole Nationale Polytechnique, en acceptant la présidence du jury. Qu'il trouve ici l'expression de mon profond respect.*

*Je tiens également à exprimer ma profonde et respectueuse gratitude à Monsieur **D. BOUKHETALA**, Maître de Conférence à l'Ecole Nationale Polytechnique qui me fait un grand honneur en acceptant d'examiner cette thèse, je lui suis très reconnaissant.*

*Mes sincères remerciements vont aussi à Monsieur **L. NEZLI**, Maître de Conférence à l'Ecole Nationale Polytechnique, pour son souci de voir ce travail achevé, et qui m'a honoré en acceptant d'examiner la thèse.*

*Qu'il me soit également permis de présenter mes remerciements les plus sincères à Mademoiselle **L. BARAZANE**, Maître de Conférence à l'Université des Sciences et de la Technologie Houari Boumediene, qui m'a honoré en acceptant d'examiner cette thèse.*

*Mes remerciements les plus sincères vont également à Monsieur **M.E. SIFI**, Chargé de Mission Audite à la SNTF, d'avoir accepté de faire partie du jury. Qu'il trouve ici l'expression de mon respect.*

Ma gratitude va également à tous les enseignants chercheurs de la Faculté d'Electronique et Informatique et du laboratoire LSEI, pour la bonne humeur et l'agréable cadre de travail qu'ils m'ont fait partager.

Je tiens à exprimer une reconnaissance particulière à mes amis et collègues des labos 47 et 49 pour l'accueil qu'ils m'ont réservé. Je leur voue une énorme reconnaissance pour m'avoir accompagné tout au long de ce travail, en m'apportant soutien moral et scientifique. Qu'ils trouvent ici mes remerciements les plus chaleureux.

*Ma vive reconnaissance va également à Monsieur **H. MEZIANI**, qui n'a pas hésité à mettre à ma disposition les moyens matériels m'ayant permis de finaliser la mise en forme de cette thèse.*

*Enfin, je ne remercierai jamais assez mes **Amis**, chez lesquels j'ai trouvé réconfort et soutien dans les moments les plus pénibles.*

TABLE DES MATIERES

TABLE DES MATIERES

INTRODUCTION GENERALE

CHAPITRE I : MODELISATION DE LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

I.1	INTRODUCTION.....	6
I.2	PRESENTATION DE L'ENSEMBLE CONVERTISSEUR – MACHINE.....	6
I.3	MODELISATION ET MISE EN EQUATION DE LA MSAP.....	7
I.3.1	Equations de la machine dans le repère (a b c).....	7
I.3.2	Equations de la machine dans le repère de Park (d, q).....	8
I.3.3	Modèle linéaire et Fonction de transfert de la MSAP.....	11
I.3.4	Simulation numérique du modèle de la machine.....	11
I.4	MODELISATION DU CONVERTISSEUR.....	13
I.4.1	Modèle de l'onduleur.....	13
I.4.2	Commande de l'onduleur.....	14
I.5	SIMULATION DE L'ENSEMBLE CONVERTISSEUR MACHINE SYNCHRONNE.....	15
I.6	CONCLUSION.....	17

CHAPITRE II : COMMANDE VECTORIELLE DE LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

II.1	INTRODUCTION.....	19
II.2	PRINCIPE DE LA COMMANDE VECTORIELLE.....	19
II.3	COMMANDE VECTORIEL AVEC REGLAGE DU COURANT.....	20
II.3.1	Synthèse des régulateurs de courants.....	20
II.3.2	Synthèse du régulateur de vitesse.....	21
II.4	SIMULATION DE LA COMMANDE EN VITESSE DE LA MACHINE.....	22
II.5	ETUDE DE ROBUSTESSE AUX VARIATIONS PARAMETRIQUES.....	25
II.6	CONCLUSION.....	30

CHAPITRE III : LA COMMANDE ADAPTATIVE

III.1	INTRODUCTION.....	32
III.2	COMMANDE A CONTRE-REACTION ET COMMANDE ADAPTATIVE.....	32
III.3	DIFFERENTES TECHNIQUES DE COMMANDE ADAPTATIVE.....	32
III.3.1	Commande adaptative à modèle de référence.....	33
III.3.2	Commande auto-ajustable.....	34
III.4	METHODES RECURSIVES D'IDENTIFICATION.....	35
III.5	COMMANDE AUTO-AJUSTABLE A VARIANCE MINIMALE.....	35
III.5.1	Principe de la commande.....	35
III.5.2	Synthèse d'un régulateur auto-ajustable à variance minimale généralisée.....	36
III.5.3	Algorithme auto-ajustable implicite.....	38
III.6	COMMANDE ADAPTATIVE A ERREUR DE PREDICTION MINIMALE.....	39
III.6.1	Principe de la commande.....	39
III.6.2	Synthèse de la loi de commande.....	39

III.7	COMMANDE A ERREUR DE PREDICTION MINIMALE PONDEREE	41
III.7.1	Principe de la commande	41
III.7.2	Synthèse de la loi de commande	41
III.8	CONCLUSION	43
CHAPITRE IV : APPLICATION DE LA COMMANDE ADAPTATIVE A LA MACHINE SYNCHRONE A AIMANTS PERMANENTS		
IV.1	INTRODUCTION.....	45
IV.2	APPLICATION DE LA COMMANDE ADAPTATIVE A LA MSAP.....	45
IV.3	RESULTATS DE SIMULATION ET DISCUSSION.....	46
IV.3.1	Commande à EPM et à EPMP de la MSAP.....	46
IV.3.2	Application de la commande à VMG.....	50
IV.4	CONCLUSION	53
CHAPITRE V : LES RESEAUX DE NEURONES ARTIFICIELS		
V.1	INTRODUCTION.....	55
V.2	HISTORIQUE SUR LES RNA.....	55
V.3	LE NEURONE BIOLOGIQUE	56
V.4	LE NEURONE FORMEL.....	56
V.4.1	Modèle mathématique non-linéaire.....	57
V.4.2	Modèle mathématique étendu	58
V.4.3	Les fonctions de transfert	58
V.5	LE PERCEPTRON	59
V.6	NEURONE A BASE RADIALE	60
V.7	LES RESEAUX DE NEURONES FORMELS OU RESEAU MULTICOUCHE.....	60
V.7.1	Les réseaux de neurones non bouclés	60
V.7.2	Les réseaux de neurones bouclés	61
V.7.3	Le choix du nombre d'unités cachées	62
V.8	APPRENTISSAGE DES RESEAUX DE NEURONES	62
V.8.1	L'apprentissage non supervisé	63
V.8.2	L'apprentissage supervisé	63
V.9	REGLES ET METHODES D'APPRENTISSAGE	63
V.9.1	La règle de Hebb	63
V.9.2	La règle de Widrow-Hoff.....	64
V.9.3	Méthode de la rétropropagation du gradient (backpropagation).....	64
V.9.4	Méthode de backpropagation avec momentum.....	66
V.9.5	Méthode du gradient conjugué.....	66
V.9.6	Algorithmes d'optimisation.....	67
V.10	LES ETAPES DE LA CONCEPTION D'UN RESEAU.....	67
V.10.1	Choix des échantillons	68
V.10.2	Normalisation des données.....	68
V.10.3	Elaboration de la structure du réseau	68
V.10.4	Apprentissage	68
V.10.5	Validation	68

TABLE DES MATIERES

V.11 APPLICATION DES RN A LA COMMANDE DES PROCESSUS.....	68
V.11.1 Système de commande par simple bouclage.....	69
V.11.2 Système de commande avec modèle interne.....	69
V.12 CONCLUSION	70

CHAPITRE IV : APPLICATION DES RESEAUX DE NEURONES ARTIFICIELS POUR LA COMMANDE DE LA MSAP

VI.1 INTRODUCTION.....	72
VI.2 CONCEPTION DU REGULATEUR NEURONAL	72
VI.2.1 Création de la base d'apprentissage	72
VI.2.2 Procédures d'apprentissage	73
VI.3 REGULATEUR NEURONAL DE VITESSE.....	74
VI.3.1 Réseau à structure monocouche	74
VI.3.2 Réseau à structure multicouche.....	75
VI.3.3 Test et validation	76
VI.4 RESULTATS ET DISCUSSION.....	76
VI.5 APPLICATION D'UN REGULATEUR NEURONAL MULTIVARIABLE (MIMO) 80	
VI.5.1 Architecture du réseau de neurones	81
VI.5.2 Apprentissage du RN	81
VI.5.3 Résultats et discussion.....	82
VI.6 CONCLUSION	84

CONCLUSION GENERALE

BIBLIOGRAPHIE

ANNEXES :

A: ALGORITHME D'ADAPTATION PARAMÉTRIQUE	94
A.1 ALGORITHME D'ADAPTATION PARAMETRIQUE.....	944
A.1.1 Algorithme du gradient simple (ou de projection).....	944
A.1.2 Algorithme des moindres carrés récursif.....	955
A.2 ESTIMATION DANS LE CAS STOCHASTIQUE:.....	977
A.3 APPLICATION AU MODELE DE LA MSAP	977
B: FORME PREDICTIVE.....	99
C: TRANSFORMATION DE PARK.....	100
D: PARAMETRES DES MACHINES UTILISEES.....	101

LISTE DES FIGURES

Figure I.1 : Structure générale de l'ensemble convertisseur-commande-machine	6
Figure I.2 : Représentation du schéma équivalent de la MSAP	7
Figure I.3 : Représentation équivalente dans le repère de Park	9
Figure I.4 : Schéma bloc de la machine synchrone à aimants permanents	10
Figure I.5 : Représentation d'état de la MSAP.	11
Figure I.6 : Simulation du fonctionnement de la machine à vide et en boucle ouverte	12
Figure I.7 : Simulation de la machine en boucle ouverte avec application de la charge à $t=3$ sec.....	12
Figure I.8 : Structure générale de l'alimentation de la MSAP	13
Figure I.9 : Schéma du modèle de l'onduleur	14
Figure I.10 : association onduleur MLI machine synchrone.....	15
Figure I.11 : Simulation en BO de l'ensemble convertisseur machine commande, à vide.....	16
Figure I.12 : Simulation en BO de l'ensemble convertisseur machine commande en charge	16
Figure II.1 : Représentation du modèle découplé de la MSAP	20
Figure II.2 : Schéma de la commande en vitesse de la MSAP.....	20
Figure II.3 : Boucle de réglage du courant.....	21
Figure II.4 : Boucle de régulation de la vitesse.....	22
Figure II.5 : Commande en vitesse de la MSAP avec application d'un couple résistant.....	23
Figure II.6 : Réponse de la commande pour une variation de consigne	23
Figure II.7 : Commande en vitesse avec application du couple résistant puis variation consigne.....	24
Figure II.8 : Commande en vitesse de la MSAP avec variation de consigne en régulation.....	24
Figure II.9 : Robustesse de la commande pour une variation de 100% de la résistance R_s en régulation ..	26
Figure II.10 : Simulation pour une variation de 100% de la résistance R_s en poursuite de la consigne	26
Figure II.11 : Test de variation de 100% de la résistance R_s en poursuite et régulation	27
Figure II.12 : Test de variation de 100% de l'inductance L_q en régulation	27
Figure II.13 : Test de variation de 100% l'inductance L_q en poursuite de la variation de consigne	28
Figure II.14 : Test de variation de 100% l'inductance L_q avec poursuite et régulation	28
Figure II.15 : Robustesse de la commande pour une variation de 100% du J_m en régulation.....	29
Figure II.16 : Robustesse du régulateur PI pour une variation de 100% du J_m en poursuite	29
Figure II.17 : Robustesse pour une variation de 100% du J_m en poursuite et régulation.....	30
Figure III.1 : Structure de base d'une commande adaptative	33
Figure III.2 : Commande adaptative à modèle de référence.....	33
Figure III.3 : Commande adaptative à modèle de référence implicite.....	34
Figure III.4 : Commande auto-ajustable schéma de base.....	35
Figure III.5 : Cycle auto-ajustable implicite.....	36
Figure III.6 : Structure du modèle ARMAX.....	36
Figure III.7 : Structure du modèle du procédé avec le modèle auxiliaire	37
Figure III.8 : Structure de commande adaptative à EPM.....	39
Figure III.9 : Structure de commande adaptative à EPM.....	41
Figure IV.1 : Schéma bloc de la commande adaptative à EPM de la MSAP.....	45
Figure IV.2 : Démarrage en charge avec un couple résistant de 0.05N.m	47
Figure VI.3 : Perturbation de charge ($\lambda=0.03$).....	47
Figure VI.4 : Perturbation de charge ($\lambda=0.8$).....	47
Figure IV.5 : Test de Variation de J	48
Figure IV.6 : Test de poursuite de trajectoire avec variation de R	48
Figure IV.7 : Poursuite de trajectoire de vitesse de référence.....	49
Figure IV.8 : Application d'un échelon de couple résistant entre 1s et 2s	49
Figure IV.9 : rejet de perturbation de couple avec une erreur +50% of résistance statorique R_s	49
Figure IV.10 : Application du couple résistant avec une variation de +300% de J_m	50
Figure IV.11 : Robustesse de la commande pour les faibles vitesses (+50% de R_s).....	50
Figure IV.12 : Robustesse de la commande pour les faibles vitesses (+300% de J_m).....	50
Figure IV.13 : Test de poursuite ($\lambda=0,03$).....	51
Figure IV.14 : Test de poursuite ($\lambda=0,2$).....	51
Figure IV.15 : Application d'un couple de charge.....	51
Figure IV.16 : Variation de la résistance R en poursuite.....	52

Figure IV.17: Variation de l'inertie J en régulation.....	52
Figure V.1: Neurone biologique.....	56
Figure V.2: Modèle Non-Linéaire du Neurone formel.....	57
Figure V.3: Modèle Etendu du Neurone formel.....	58
Figure V.4: Les formes usuelles de la fonction d'activation.....	59
Figure V.5: Structure d'un réseau multicouche (feedforward).....	60
Figure V.6: Réseau multicouche à connexions locales.....	61
Figure V.7: Réseau à connexions récurrentes (ou feedback).....	61
Figure V.8: Réseau à connexion complexe.....	61
Figure V.9: Forme canonique d'un réseau de neurones bouclé.....	62
Figure V.10: Perceptron Multi Couches.....	64
Figure V.11: Méthode de la rétropropagation du gradient.....	66
Figure V.12: Système de commande par simple bouclage.....	69
Figure V.13: Système de commande avec modèle interne.....	69
Figure VI.1: Schéma de sauvegarde des données d'entraînement.....	73
Figure VI.2: l'Architecture du Réseau.....	75
Figure VI.3: processus d'apprentissage.....	76
Figure VI.4: Résultat de l'Apprentissage du Réseau de Neurones.....	76
Figure VI.5: Schéma de la commande en vitesse de la MSAP avec régulateur neuronal.....	77
Figure VI.6: Poursuite de la trajectoire de référence en présence de la charge utilisant le PI.....	78
Figure VI.7: Robustesse du PI vis à vis d'une erreur dans paramètre R , ($R=5R_n$).....	78
Figure VI.8: Poursuite de la trajectoire en présence de la perturbation de charge utilisant le RNA.....	79
Figure VI.9: Robustesse du RNA vis à vis d'une erreur dans paramètre R ($R=5R_n$).....	79
Figure VI.10: Comparaison entre la Robustesse du PI et du RNA à une erreur paramétrique ($R=5R$)....	80
Figure VI.11: Comparaison entre la Robustesse du PI et du RNA à une erreur paramétrique ($J_m=3J_m$)....	80
Figure VI.12: Structure multivariable de la commande de la MSAP.....	81
Figure VI.13: Architecture du Contrôleur Neuronale Multivariable.....	81
Figure VI.14: Poursuite de la trajectoire avec couple de charge utilisant le RNA MIMO.....	82
Figure VI.15: Test de poursuite sur plusieurs niveaux de vitesse utilisant le RNA MIMO.....	82
Figure VI.16: Test de Robustesse du RNA MIMO vis à vis d'une variation de $3J_m$	83
Figure VI.17: Robustesse du RNA MIMO vis à vis de variations paramétriques.....	83

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Le domaine de l'entraînement électrique à vitesse variable a connu un essor considérable. Grâce à cette évolution, la commande des machines électriques a fait des progrès énormes. En effet, les exigences accrues de l'industrie sont à l'origine de l'utilisation de techniques de réglage et de commande des machines de plus en plus performantes. Ceci est dû d'une part, à l'évolution de l'Automatique par l'introduction de méthodes intelligentes de commande des systèmes de plus en plus complexes, et d'autre part, aux progrès enregistrés dans le domaine de la microélectronique et de l'électronique de puissance. Ce qui a permis la conception des convertisseurs statiques très rapides qui permettent la mise en pratique de ces techniques [PIL89] [NAH01] [SEI88] [BOU99] [FU 91].

Grâce à la simplicité de la commande du flux et du couple, la machine à courant continu reste toujours très utilisée dans les domaines nécessitant des entraînements à vitesse variables. Cependant, la présence du système balais-collecteur est d'un handicap considérable. Les machines à courant alternatif possèdent de nombreux avantages. L'absence de collecteur leur permet d'avoir un encombrement minimal, une fiabilité accrue, un coût de construction plus réduit et une vitesse de fonctionnement élevée. Parmi ces machines, La machine synchrone à aimants permanents (MSAP) est un actionneur qui s'impose dans les applications nécessitant des performances dynamiques et statiques très élevées, et plus particulièrement, dans les systèmes embarqués en raison de sa puissance massique élevée. Néanmoins, son utilisation exige des structures internes et des stratégies de commande plus complexes du fait de sa forte non linéarité et du couplage entre le flux et le couple [ARA98] [ROB95] [FU 95].

Des performances semblables à celle du moteur à courant continu à excitation séparée, sont obtenues pour ces actionneurs par l'utilisation de méthodes de commande faisant appel à la commande vectorielle. La stratégie consiste à maintenir le courant statorique en quadrature avec le flux rotorique (flux des aimants) c'est à dire de le réduire à la seule composante i_q . Le flux et le couple sont alors pilotés par deux courants indépendants [BER02] [BOU99] [NAH01].

Les régulateurs conventionnels type PI largement utilisés dans ces structures de commande, sont insuffisants pour répondre aux performances exigées. Ils sont sujets à une détérioration des performances en présence des perturbations de charge et des variations paramétriques. Pour palier ces dégradations, l'utilisation des commandes modernes et intelligentes est plus que nécessaire. Un certain nombre de ces commandes ont été déjà appliquées à la machine synchrone à aimants permanents telles que la commande adaptative, les réseaux de neurones, la commande floue ... Les travaux publiés dans la littérature ont montré l'efficacité de ces méthodes à conduire les systèmes particulièrement complexes et à maintenir un niveau optimal de l'indice de performance [IBR02] [BAG99] [BOU95] [KIM03] [LAN86] [LIN98] [MIL92] [MIN04] [PAJ05] [SHI03] [CIR02].

Dans le présent travail, une approche de la commande adaptative, basée sur les régulateurs auto-ajustables, a été analysée et appliquée à la commande en vitesse de la machine synchrone à aimants permanents. Deux stratégies de commande sont explorées : la commande à erreur de prédiction minimale pondérée (EPMP) et la commande à variance minimale généralisée. L'approche directe du régulateur auto-ajustable a été considérée. Les paramètres de

la loi de commande sont directement estimés. La première commande est une extension de la technique à erreur de prédiction minimale simple (EPM) qui nécessite des hypothèses pour garantir la stabilité de la boucle fermée. L'application de cette nouvelle stratégie couvre tous les types de systèmes grâce à l'introduction dans la structure du critère à minimiser un certain nombre de degrés de liberté pour assurer la stabilité de la boucle fermée. La difficulté de cette stratégie réside dans le choix des coefficients des polynômes de pondération [ÅST73] [ÅST89] [GOO84].

La deuxième technique est introduite pour prendre en charge les bruits qui peuvent entacher les grandeurs mesurées. Ces lois de commande sont associées à un algorithme d'estimation paramétrique, lequel est basé sur les moindres carrés récurrents, pour donner lieu à un régulateur auto-ajustable [ÅST73] [ÅST89] [CLA75] [CLA79]. Différents tests de simulation, montrant les performances de cette commande, sont présentés et discutés.

Dans une seconde étape, l'utilisation des réseaux de neurones artificiels (RNA) pour la commande a été explorée. L'emploi des réseaux de neurones plutôt que des techniques classiques pour commander des systèmes complexes peut se justifier par la simplicité de mise en œuvre (peu d'analyse mathématique préliminaire), par leur capacité d'approximation universelle, par la possibilité de considérer le processus comme une boîte noire et par la possibilité de débiter la conduite avec un minimum d'informations sur le processus. L'utilisation alors des réseaux de neurones pour l'asservissement du moteur synchrone à aimants permanents présente alors un intérêt considérable [KUK99] [LIU98] [PAJ05] [SAL98] [SHE99] [CIR02].

Dans cet objectif, une structure de réseau de neurones multicouches statiques, a été analysée et appliquée à la commande en vitesse de la MSAP. Pour simplifier la synthèse de la commande, nous avons adopté la stratégie d'apprentissage supervisée. La difficulté de l'utilisation de ce régulateur réside dans le choix de sa structure [DAV90] [HAR98] [RAH98] [SOL00] [YAN03].

Des tests de simulation effectués ont montré de bonnes performances en termes de poursuite de la trajectoire de référence et de rejet de la perturbation de charge ainsi qu'une robustesse du régulateur neuronal vis-à-vis des variations paramétriques subies par le variateur.

La thèse s'articule donc autour des chapitres suivants:

Dans le premier chapitre, nous présenterons le modèle mathématique de la machine synchrone à aimants permanents (MSAP) permettant l'étude de son comportement dynamique. Le modèle adopté est basé sur la transformation de Park. Le principe de fonctionnement et la modélisation du convertisseur constitué par un onduleur à modulation de largeur d'impulsions sont aussi présentés.

Le deuxième chapitre porte sur la commande vectorielle de la MSAP. La régulation de la vitesse (boucle externe) et des courants du modèle de Park (boucle interne) de la MSAP sont assurées par des régulateurs classiques type proportionnel intégral (PI). Nous allons montrer la non robustesse de régulateur classique utilisé et sa faiblesse lors des variations paramétriques supposées du modèle de la machine.

Dans le troisième chapitre, nous nous proposons de concevoir des contrôleurs adaptatifs qui tiennent compte des incertitudes dans les paramètres du système commandé. Il s'agit de la

résistance statorique et le couple de charge d'un côté et de l'inductance et du moment d'inertie de l'autre côté. Deux stratégies de commande sont explorées : la commande à erreur de prédiction minimale pondérée (EPMP) et la commande à variance minimale généralisée. Une étude de la robustesse des commandes utilisées et leurs efficacités lors des variations paramétriques du modèle de la machine sont mises en évidence. Les performances de l'ensemble des contrôleurs adaptatifs présentés sont discutées.

Le chapitre quatre est consacré à la présentation des réseaux de neurones artificiels et des différentes configurations qui se présentent pour les problèmes de modélisation, d'identification et de commande de processus.

Finalement, le cinquième chapitre est destiné à la conception et à l'application de contrôleur neuronal à la commande en vitesse de la machine. En premier lieu, l'architecture du réseau correcteur retenu est appliquée à la boucle de vitesse de la machine. Les performances du contrôleur sont testées en tenant compte des variations paramétriques supposées de notre système et du couple de charge. En second lieu, un régulateur multivariable (MIMO) est élaboré pour prendre en charge les deux boucles de régulation. Les performances du réseau obtenu, sont comparées à celles de contrôleur neuronal de vitesse.

Une conclusion générale vient clôturer les six chapitres de cette thèse.

CHAPITRE

I

MODELISATION DE LA MACHINE
SYNCHRONNE A AIMANTS
PERMANENTS

I MODELISATION DE LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

I.1 INTRODUCTION

L'étude de la commande d'un système est un travail qui nécessite une bonne connaissance de son modèle dynamique afin de bien prédire son comportement dans les différents modes de fonctionnement envisagés.

Les moteurs à courant continu ont assuré pendant longtemps le fonctionnement de la plupart d'équipements industriels (robots et machines outils). Cependant, le collecteur mécanique reste leur principal défaut que l'on tolère mal dans certaines applications. C'est pour cette raison que l'industrie s'est orientée vers l'utilisation des machines à courant alternatif afin de profiter de leurs avantages tels que, la flexibilité de variation de vitesse et la stabilité de fonctionnement. Parmi les machines à courant alternatif utilisés dans les entraînements, la machine synchrone à aimants permanents (MSAP) qui présente un certain nombre d'avantages, à savoir les pertes au rotor nulle, une grande capacité de surcharge et surtout à cause de son couple massique élevé comparé aux machines asynchrone et synchrone à rotor bobiné [BER02], [ARA98].

Cet avantage lui donne la supériorité aux autres types de machines. Elle est souvent appelée machine à courant continu sans balais car lorsqu'elle est autopilotée, ses caractéristiques coïncident avec celles d'un moteur à courant continu à excitation shunt. Par ailleurs, les travaux effectués sur cette ont montré la similitude de ses caractéristiques avec celles moteur à courant continu et que les lois de commande s'appliquent facilement sur cette machine. Cependant, l'autopilotage nécessite l'utilisation d'un onduleur et un capteur de position.

Dans ce chapitre, la modélisation de la machine synchrone à aimants permanents est présentée. A cet effet, le modèle triphasé ainsi que le modèle obtenu à l'aide de la transformation de Park sont abordés.

I.2 PRESENTATION DE L'ENSEMBLE CONVERTISSEUR – MACHINE

Le système étudié est représenté dans la figure I.1. Il est composé d'une machine synchrone à aimants permanents à pôles lisses, sans amortisseurs ni pièces polaires, associée à un dispositif de commande à vitesse variable. La fréquence des tensions ou des courants d'alimentation, délivrées par un convertisseur statique, est asservie à la position du rotor. Cette procédure de fonctionnement autopiloté permet d'obtenir la variation de vitesse [BOU99].

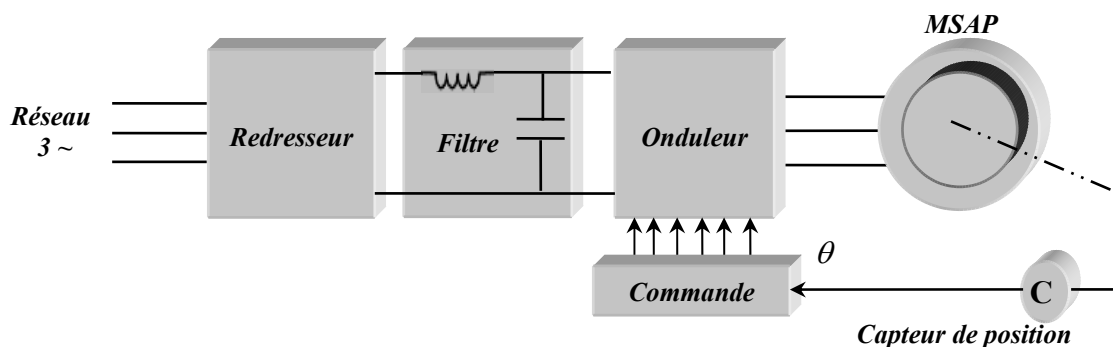


Figure I.1 : Structure générale de l'ensemble convertisseur-commande-machine

I.3 MODELISATION ET MISE EN EQUATION DE LA MSAP

Pour effectuer l'analyse d'un système, il est nécessaire de connaître les relations entre ses grandeurs d'entrées et de sorties. L'ensemble de ses relations constitue son modèle mathématique. L'actionneur qui fera l'objet de cette étude est une machine synchrone à aimants permanents à distribution sinusoïdale, à pôles lisses, l'inducteur est animé d'une vitesse de rotation ω [GRE94], [KAD00].

Les hypothèses simplificatrices usuelles adoptées dans la modélisation de la machine, données dans la majorité des références, sont:

- ☐ Les circuits magnétiques ne sont pas saturés, ce qui permet d'exprimer le flux comme fonction linéaire des courants.
- ☐ Les pertes par courants de Foucault et par hystérésis sont négligées.
- ☐ Les couplages capacitifs entre les enroulements ainsi que l'effet de peau sont négligés.
- ☐ La distribution de la force magnétomotrice créée par les enroulements au stator est sinusoïdale.
- ☐ Il n'existe pas d'enroulement amortisseur au rotor.
- ☐ Le système de tension est équilibré (composante homopolaire nulle).

Le schéma équivalent de cette machine est représenté par la figure I.2.

I.3.1 Equations de la machine dans le repère (a b c)

La machine peut être modélisée par les équations électromécaniques, exprimées dans le repère (a b c).

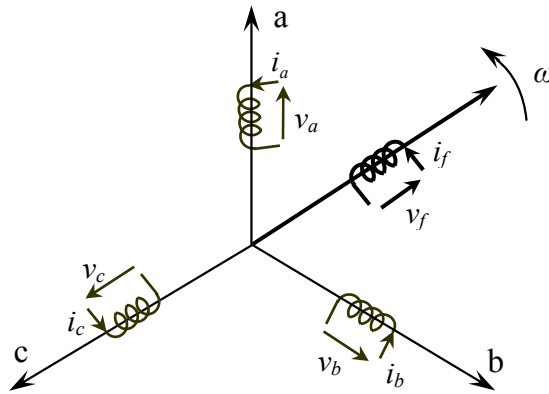


Figure I.2 : Représentation du schéma équivalent de la MSAP

- Les équations aux tensions de la machine synchrone choisie sont :

Pour le stator :

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \quad (I.1)$$

La machine étant triphasée équilibrée symétrique, la résistance des phases statoriques : $R_s = R_a = R_b = R_c$

Le système sous forme matricielle :

$$[V] = [R_s][i] + \frac{d}{dt}[\psi] \quad (I.2)$$

Ou : $[V] = [v_a \ v_b \ v_c]^T$ est le vecteur des tensions des phases statoriques.

$[i] = [i_a \ i_b \ i_c]^T$: Vecteur des courants des phases statoriques.

$[\psi] = [\psi_a \ \psi_b \ \psi_c]^T$: Vecteur des flux totaux à travers les bobines statoriques.

Pour le rotor :

$$v_f = R_f i_f \quad (I.3)$$

v_f , R_f et i_f : sont la tension, la résistance et le courant rotoriques respectivement.

La relation flux courants est donnée par:

$$[\psi] = [L][i] + [\psi_f] \quad (I.4)$$

Soit :

$$[V] = [R_s][i] + \frac{d}{dt}([L][i] + [\psi_f]) \quad (I.5)$$

$[L]$: Matrice des inductances propres et mutuelle statoriques, indépendante de θ , puisque quelque soit la position du rotor, la machine est géométriquement symétrique [BOU99].

Le flux d'excitation des aimants permanents est donné par :

$\psi_f = M_f i_f$, où M_f et i_f sont des grandeurs fictives constantes.

L'équation (I.4) devient :

$$[\psi] = [L][i] + [M_f]i_f \quad (I.6)$$

L'aimant permanent est représenté par un enroulement fictif, de ce fait l'inductance est constante.

L'expression du couple électromagnétique est :

$$\Gamma_{em} = \frac{1}{2}[i]^T \left(\frac{d}{d\theta}[L] \right) [i] \quad (I.7)$$

L'équation mécanique :

La conversion de l'énergie électrique en énergie mécanique est régie par l'équation mécanique suivante :

$$J_m \frac{d\omega_r}{dt} = \Gamma_{em} - \Gamma_r - f_r \omega_r \quad (I.8)$$

Γ_r : le couple résistant.

J_m : le moment d'inertie des parties tournantes de l'ensemble machine charge.

f_r : le coefficient de frottement.

ω_r : la vitesse de rotation du rotor.

I.3.2 Equations de la machine dans le repère de Park (d, q)

Les équations obtenues dans le repère abc sont fortement non-linéaires et couplées. Elles sont fonctions de la position du rotor θ . Ceci fait apparaître des difficultés pour la résolution du système. Pour simplifier ce problème, la plupart des travaux dans la Littérature font appel à l'utilisation de la transformation de Park [LOU99] [PIL89]. Cette transformation, appliquée aux variables réelles (tensions, courants et flux), permet d'obtenir des variables fictives appelées les composantes d - q ou de Park. Ceci peut être interprété comme étant une substitution des enroulement de phases du système réel (a , b , c) en enroulements orthogonaux d'axes (d q)

tournant à une vitesse ω par rapport au stator (Figure I.3). Ce changement de repère rend les équations dynamiques de la machine plus simples ce qui facilite leur étude et leur analyse [FU95].

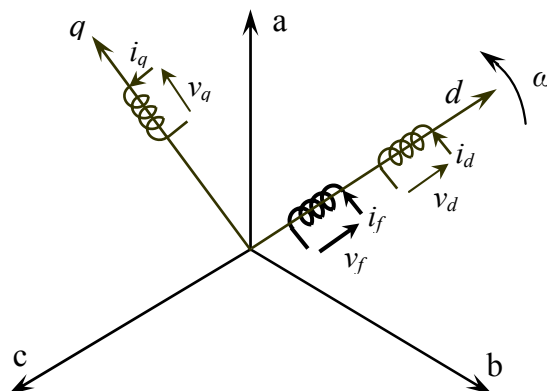


Figure I.3 : Représentation équivalente dans le repère de Park

Nous pouvons alors effectuer le changement de variables comme suit :

$$[X_{dqo}] = [P(\theta)][X_{abc}] \quad (\text{I.9})$$

X_{dqo} et X_{abc} représentent les vecteurs courant, tension ou flux, dans le repère $(d q)$ et $(a b c)$ respectivement.

$P(\theta)$ représente la matrice de la transformation de *Park*.

Comme les enroulements statoriques sont montés en étoile à neutre isolé, la somme instantanée des courants statoriques est nulle, de sorte qu'il n'y a pas de composante homopolaire.

L'équation (I.9) devient :

$$[X_{dq}] = [P(\theta)][X_{abc}] \quad (\text{I.10})$$

D'où les transformations inverses :

$$[X_{abc}] = [P(\theta)^{-1}][X_{dq}] \quad (\text{I.11})$$

La matrice et son inverse, assurant cette transformation sont décrits en (Annexe C).

En appliquant la transformation de *Park* sur le système d'équations (I.1), et tenant compte de (I.10 et 11), nous exprimons tous les vecteurs dans un repère lié au rotor. Si θ est l'angle électrique entre le rotor et le stator alors:

$$\begin{aligned} v_d &= R_s i_d + \frac{d\psi_d}{dt} - p \omega_r \psi_q \\ v_q &= R_s i_q + \frac{d\psi_q}{dt} + p \omega_r \psi_d \end{aligned} \quad (\text{I.12})$$

où $\omega_s = p \omega_r$ est la vitesse électrique, p étant le nombre de paires de pôles.

-Les relations flux courants :

$$\begin{aligned} \psi_d &= L_d i_d + \psi_f \\ \psi_q &= L_q i_q \end{aligned} \quad (\text{I.13})$$

L_d, L_q : inductances d'axes direct et en quadrature ou :

$$L_d = L_s - M_f + 3/2 L'_s$$

$$L_q = L_s - M_f - 3/2 L'_s$$

Dans les machines synchrones à aimants permanents, les aimants engendrent un flux constant dans le référentiel choisi et sachant que i_f est constant l'équation (I.13) devient [NAH01]:

$$\begin{aligned} v_d &= R_s i_d + L_d \frac{di_d}{dt} - p\omega_r L_q i_q \\ v_q &= R_s i_q + L_q \frac{di_q}{dt} + p\omega_r L_d i_d + p\omega_r \psi_f \end{aligned} \quad (\text{I.14})$$

Le couple électromagnétique développé par la MSAP s'écrit dans le repère (d,q) de Park selon :

$$\Gamma_{em} = p \cdot (\psi_d i_q - \psi_q i_d) = p [(L_d - L_q) i_d + \psi_f] i_q \quad (\text{I.15})$$

$(L_d - L_q) i_d i_q$: couple dû à la saillance des pôles.

Le rotor étant lisse ($L_d=L_q$) l'équation se simplifie alors à :

$$\Gamma_{em} = p\psi_f i_q \quad (\text{I.16})$$

Les équations de la machine synchrone à aimants permanents dans le repère de Park sont regroupées dans le système suivant ;

$$\begin{cases} v_d = R_s i_d + L_d \frac{di_d}{dt} - p\omega_r L_q i_q \\ v_q = R_s i_q + L_q \frac{di_q}{dt} + p\omega_r L_d i_d + p\omega_r \psi_f \\ J_m \frac{d\omega_r}{dt} = \Gamma_{em} - \Gamma_r - f_r \omega \\ \omega_r = \frac{d\theta}{dt} \\ \Gamma_{em} = p \cdot (\psi_d i_q - \psi_q i_d) = p [(L_d - L_q) i_d + \psi_f] i_q \end{cases} \quad (\text{I.17})$$

Le schéma bloc de la machine issu du modèle donné en (I.17) est représenté par la figure I.4.

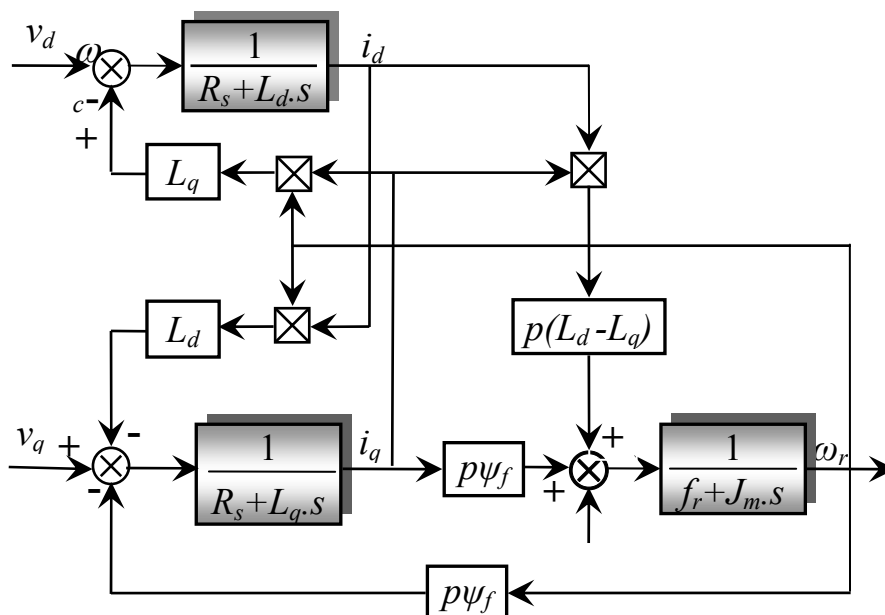


Figure I.4 : Schéma bloc de la machine synchrone à aimants permanents

Ce schéma est composé de plusieurs blocs linéaires illustrant la relation qui existe entre les entrées (commandes), les états et les sorties. Les non-linéarités sont représentées par les blocs de multiplication qui caractérisent le couplage entre les axes d et q . Pour commander le couple de la MSAP, il sera nécessaire de contrôler le vecteur courant par l'intermédiaire de ses composantes suivant l'axe direct i_d et l'axe en quadrature i_q . De plus, les dynamiques du système sont représentées par trois fonctions de transfert. Deux fonctions donnant la dynamique rapide (courants) et une fonction qui définit la dynamique lente (vitesse).

I.3.3 Modèle linéaire et Fonction de transfert de la MSAP

Dans la plupart des cas, la dynamique mécanique de la machine est nettement plus lente que celle électrique et nous pouvons donc faire l'hypothèse que la vitesse angulaire est considérée constante dans les équations électriques de la machine [SEI88]. Nous avons alors le modèle linéaire suivant :

$$\frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \frac{L_q}{L_d} \omega_r \\ -\frac{L_d}{L_q} \omega_r & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \left(\begin{bmatrix} v_d \\ v_q \end{bmatrix} - \begin{bmatrix} e_d \\ e_q \end{bmatrix} \right) \quad (\text{I.18})$$

e_d et e_q sont les composantes directe et en quadrature de f.e.m. données par :

$$\begin{cases} e_d = 0 \\ e_q = \omega_r \psi_f \end{cases}$$

La matrice de transfert du système $Y(s)$ est définie par :

$$Y(s) = \frac{1/R_s}{P(s)} \begin{bmatrix} 1 + \tau_q s & \omega_r \tau_q \\ -\omega_r \tau_d & 1 + \tau_d s \end{bmatrix} \quad (\text{I.19})$$

avec $P(s) = (1 + \tau_d s)(1 + \tau_q s) + \omega^2 \tau_d \tau_q$

où $\tau_d = L_d / R_s$ et $\tau_q = L_q / R_s$ sont les constantes de temps électriques.

La machine peut être schématisée par un bloc de représentation d'état mettant en évidence les entrées et sorties du système (Figure I.5).

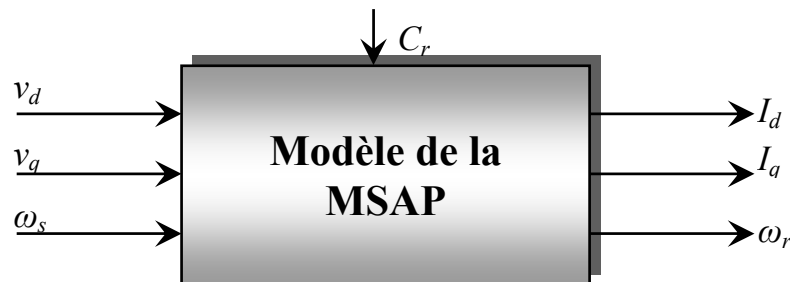


Figure I.5: Représentation d'état de la MSAP.

I.3.4 Simulation numérique du modèle de la machine

La simulation du comportement de la machine en boucle ouverte. Les paramètres de la machine M1 utilisés sont donnés en Annexe D. Les figures I.6 et I.7 montrent un essai à vide et un essai en charge en utilisant une source d'alimentation sinusoïdale.

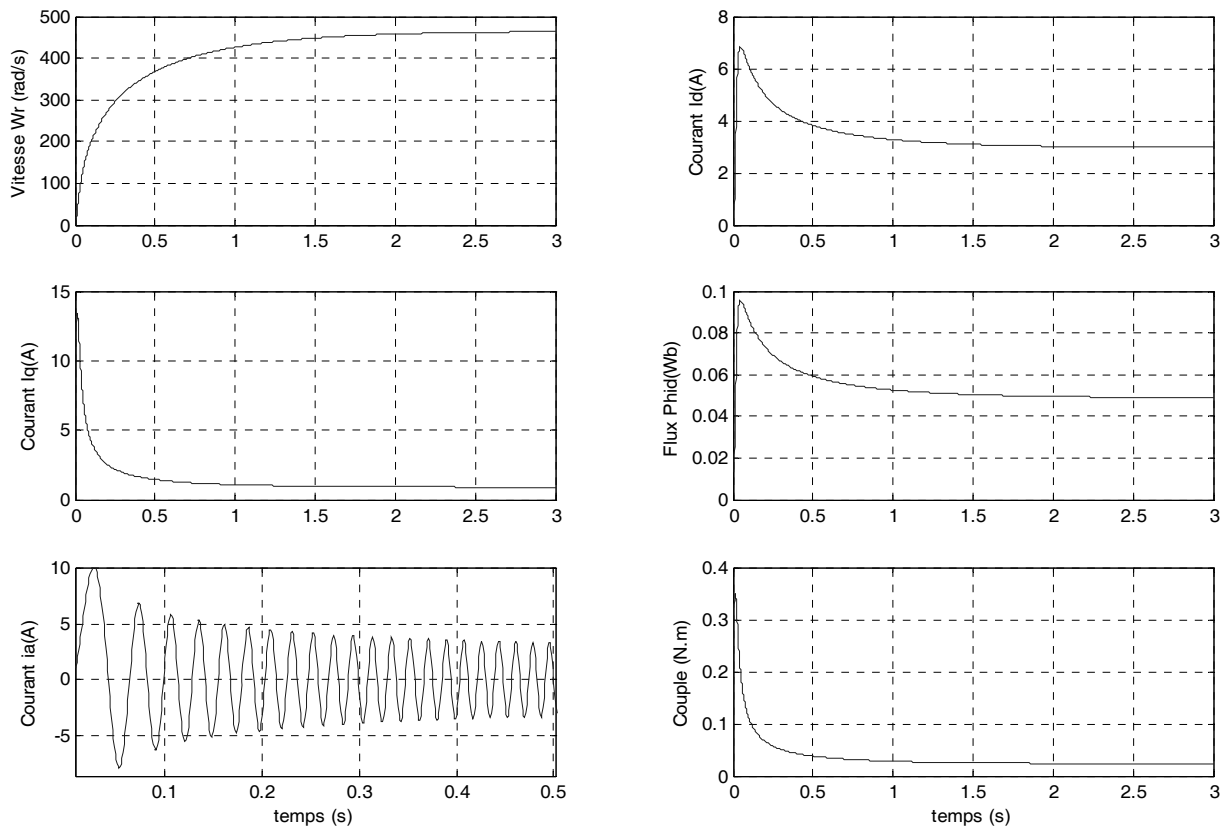


Figure I.6 : Simulation du fonctionnement de la machine à vide et en boucle ouverte

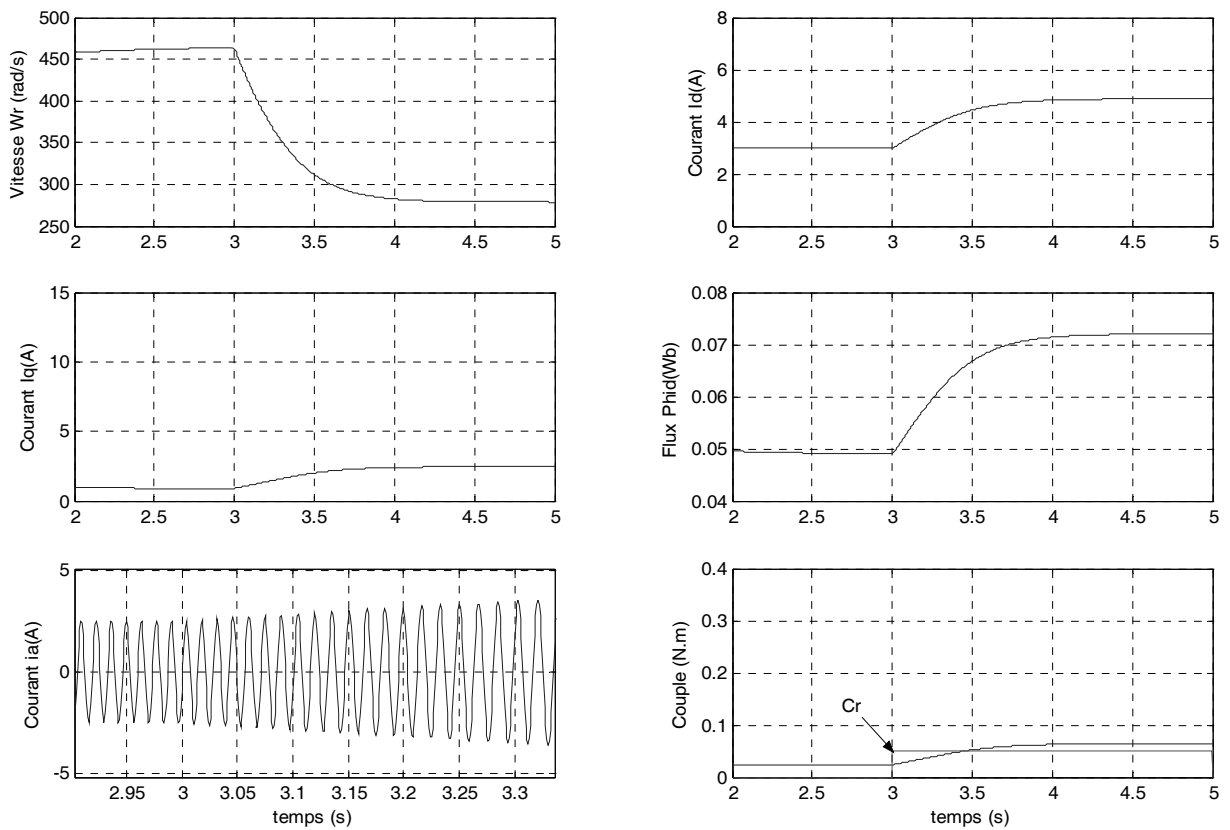


Figure I.7 : Simulation du fonctionnement de la machine en boucle ouverte avec démarrage à vide puis application de la charge à $t=3$ sec

I.4 MODELISATION DU CONVERTISSEUR

Pour assurer un fonctionnement à vitesse variable de la machine synchrone, il est nécessaire d'alimenter la machine à fréquence variable. Cette alimentation se fait à l'aide d'un convertisseur statique. A la sortie du convertisseur, les tensions d'alimentation statoriques sont contrôlées en amplitude et leur fréquence est proportionnelle à la vitesse de rotation de la machine.

Les onduleurs de tension alimentent les machines à courant alternatif à partir d'une source de tension continue. Ils permettent d'imposer aux bornes de la machine des tensions d'amplitude et de fréquence réglables par la commande [LAJ99] [NAH01].

Dans notre étude la machine est alimentée par un onduleur de tension. Ce dernier est constitué de trois bras utilisant des interrupteurs à trois segments, bidirectionnels en courant et commandés à l'ouverture et à la fermeture. Les interrupteurs peuvent être, selon la puissance mise en jeu, des transistors MOS ou bipolaire, des IGBT ou des GTO associés à une diode de récupération en antiparallèle.

Vu de l'extérieur, ce dispositif est analogue à une machine à courant continu alimentée par un hacheur. On parle alors de moteur à courant continu sans balais (Brushless Motor) pour désigner l'ensemble convertisseur-moteur synchrone autopiloté. La structure générale de l'alimentation est représentée par la figure I.8 [LAJ99].

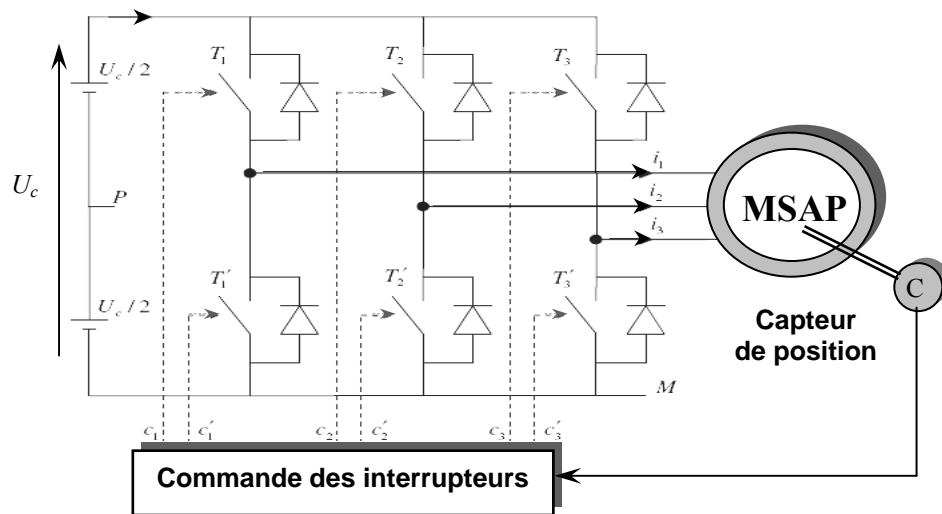


Figure I.8 : Structure générale de l'alimentation de la MSAP

I.4.1 Modèle de l'onduleur

Dans l'étude de l'ensemble complet de commande convertisseur machine, nous nous intéressons davantage au comportement de la machine qu'à celui du convertisseur. De ce fait, nous considérons que les composants semi conducteurs sont modélisés par des interrupteurs idéaux 'Ki' (figure I.9), à chaque bras est associé une fonction logique Fi, tel que :

$$F_i = \begin{cases} 1 & \text{si l'interrupteur est fermé} \\ 0 & \text{si l'interrupteur est ouvert} \end{cases}$$

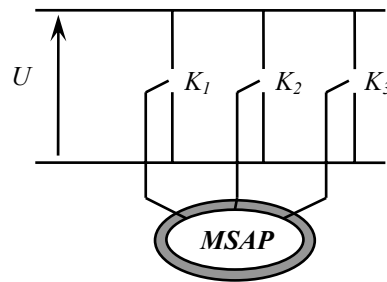


Figure I.9 : Schéma du modèle de l'onduleur

U est la tension continue alimentant l'onduleur de tension.

Les tensions de lignes sont :

$$U_{ab} = U(F_1 - F_2)$$

$$U_{bc} = U(F_2 - F_3)$$

$$U_{ca} = U(F_3 - F_1)$$

Les tensions simples v_a , v_b et v_c s'expriment comme suit :

$$v_a = \frac{1}{3}(U_{ab} - U_{cb})$$

$$v_b = \frac{1}{3}(U_{bc} - U_{ca})$$

$$v_c = \frac{1}{3}(U_{cd} - U_{bc})$$

Sous forme matricielle, nous aboutissons donc à :

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} U \quad (\text{I.20})$$

A l'entrée de l'onduleur le courant s'écrit :

$$i = F_1 i_a + F_2 i_b + F_3 i_c \quad (\text{I.21})$$

En fonction du type de stratégie de commande adoptée, les fonctions F_i sont déterminées.

I.4.2 Commande de l'onduleur

Sachant que dans notre cas, l'onduleur est alimenté en tension, et régulé en courant avec un régulateur classique (PI), les grandeurs de sortie sont donc des tensions désirées aux bornes de la machine. La commande de l'onduleur par la technique de modulation de largeur d'impulsion (MLI), permet de reconstituer ces grandeurs à partir d'une source à fréquence et à tension fixe. Cette stratégie permet aussi de repousser les fréquences élevées des harmoniques de la tension de sortie. Le réglage est effectué par la durée d'ouverture et de fermeture des interrupteurs [ARA98] [BAG99].

Parmi les nombreuses lois de commande permettant d'élaborer une onde MLI, la technique la plus simple et la plus répandue est la triangulo-sinusoïdale. Pour cette stratégie, la détermination des instants d'échantillonnage ainsi que la largeur d'impulsion s'effectue en comparant une tension de référence modulante sinusoïdale, de fréquence f , à une porteuse

d'amplitude fixe et de fréquence f_p nettement supérieur à f . Cette stratégie est caractérisée par deux paramètres : l'indice de modulation m et le taux de modulation r [SEI88].

$$m = f_p / f \quad ; \quad r = V_r / V_p .$$

La figure I.10 représente l'association onduleur à MLI machine synchrone, où les tensions de référence sinusoïdales $v_a^* v_b^* v_c^*$ sont comparées au signal de la porteuse v_p , afin de déduire les instants de commutation des interrupteurs.

Les tensions de référence sont données par la relation :

$$v_j^* = r \sin(2\pi f t - 2(j-1)\pi/3) \quad (I.22)$$

Avec : $j=1, 2, 3$.

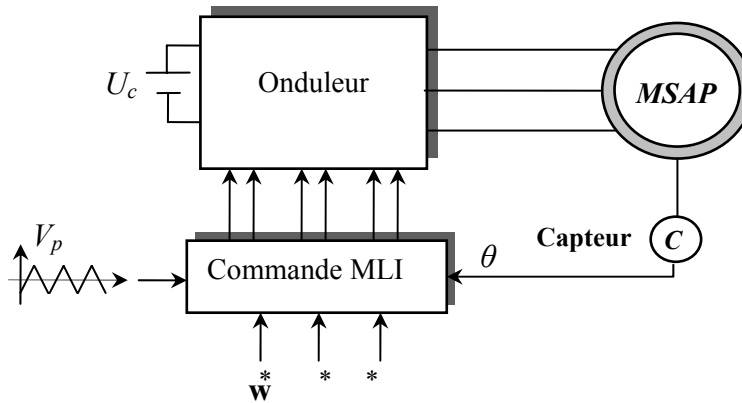


Figure I.10 : association onduleur MLI machine synchrone

La porteuse est de la forme suivante :

$$V_p(t) = \begin{cases} \frac{4t}{T_p}(4n+1) & \text{si } t \in [nT_p, \frac{n+1}{2}T_p] \\ \frac{4t}{T_p}(4n+3) & \text{si } t \in [\frac{n+1}{2}T_p, (n+1)T_p] \end{cases} \quad (I.23)$$

T_p est la période du signal de la porteuse V_p .

En supposant que la commande de l'onduleur de tension est quasi-parfaite (absence de retard, de chute de tension, et de pertes en commutation), celui-ci peut être modélisé par un simple gain. La commande du système se déduit alors du modèle le représentant.

I.5 SIMULATION DE L'ENSEMBLE CONVERTISSEUR MACHINE SYNCHROME

Le système constitué de l'ensemble convertisseur machine synchrone à aimants permanents modélisé précédemment, est simulé et représenté dans cette partie par les figures I.11 et I.12. Nous effectuons un démarrage à vide, et un test avec application de couple de charge. Nous constatons des ondulations autour des grandeurs de la machine qui sont dues à la présence de l'onduleur.

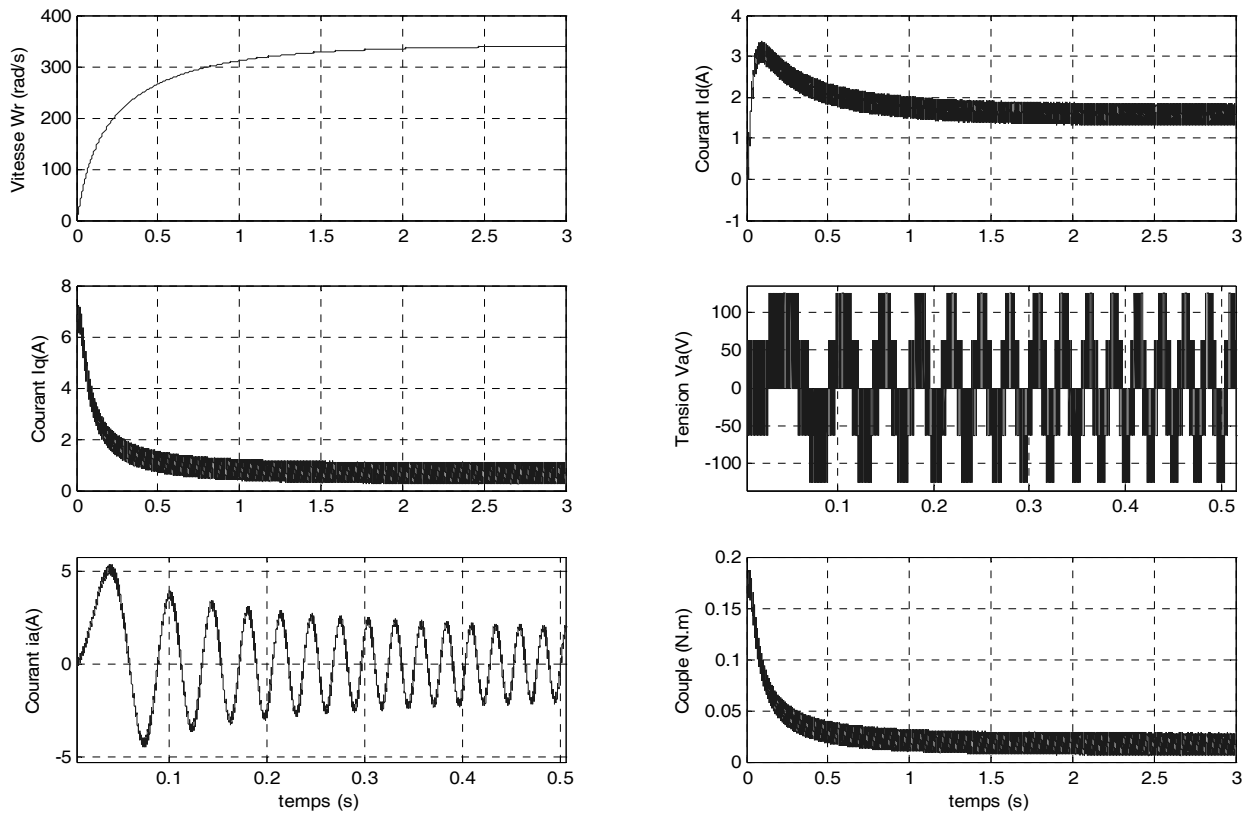


Figure I.11 : Simulation en BO de l'ensemble convertisseur machine commande, à vide

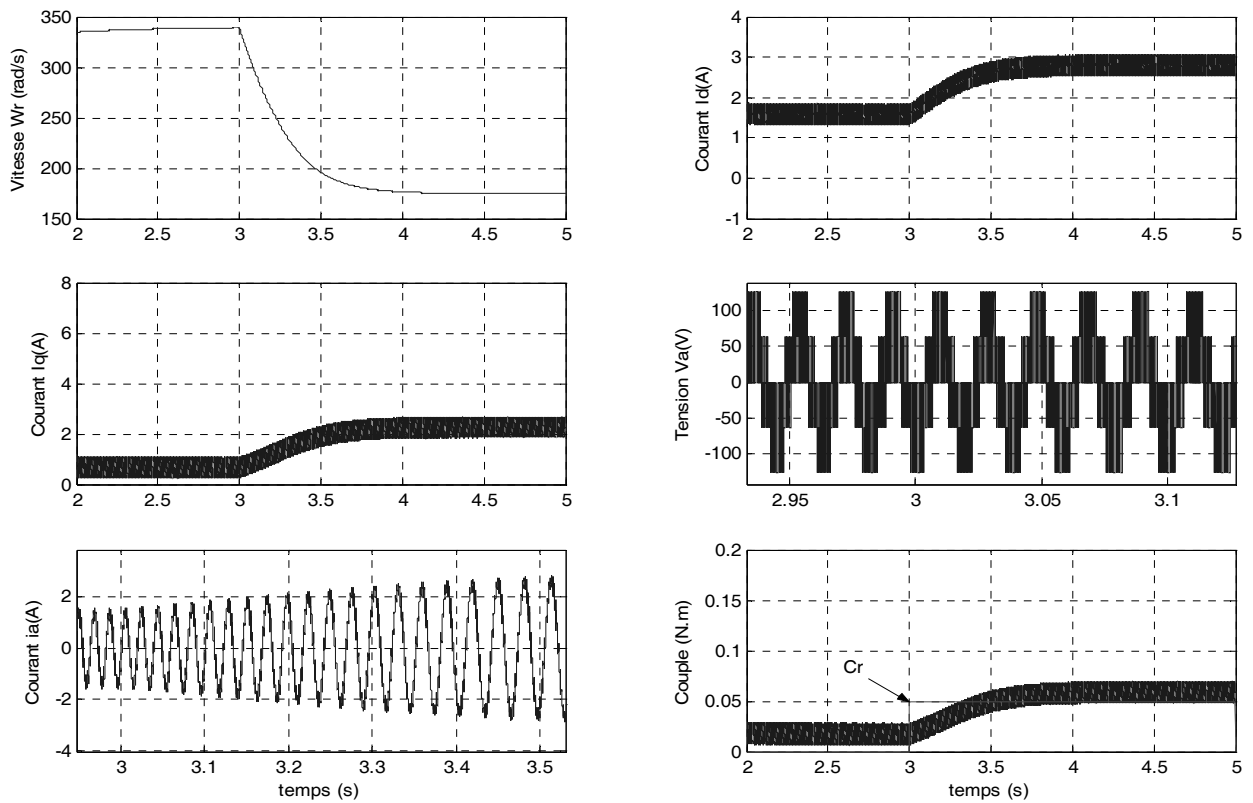


Figure I.12 : Simulation en BO de l'ensemble convertisseur machine commande en charge

I.6 CONCLUSION

Dans ce chapitre, nous avons élaboré un modèle permettant la commande à vitesse variable de la machine synchrone à aimant permanent. L'utilisation de la transformation de *Park* nous a permis d'aboutir à un modèle simplifié facilitant ainsi l'application des commandes réservée pour les systèmes linéaires. L'association machine synchrone-convertisseur statique est aujourd'hui utilisée dans le domaine des fortes puissances (Traction électrique, propulsion marine). C'est une structure qui a permis à la machine synchrone de conquérir les entraînements réglés par moteur à courant continu dont la puissance était limitée par la présence du collecteur mécanique.

L'association machine synchrone à aimant permanent - onduleur de tension MLI est largement utilisée dans les entraînements de petites puissances et notamment en robotique. A couple électromagnétique égal, ces machines grâce à l'absence de collecteur mécanique ont un moment d'inertie du rotor plus faible, et permettent des vitesses de rotations plus élevées que les moteurs à courant continu. Ceci ne peut que contribuer à améliorer les performances dynamiques des systèmes entraînés. La commande vectorielle appliquée à la MSAP fera l'objet du chapitre suivant.

CHAPITRE

-II-

COMMANDE VECTORIELLE DE LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

II COMMANDE VECTORIELLE DE LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

II.1 INTRODUCTION

Le développement de nouvelles technologies de semi-conducteurs, et de méthodes de calculs, a permis l'application de nouvelles commandes, tel que principalement la commande par l'orientation du flux ou la commande vectorielle, qui réalise le découplage entre flux et courant [BAG99]. Cette dernière a rendu possible l'utilisation des machines à courant alternatifs dans les secteurs occupés jusque la par les machines à courant continu tels que la robotique, l'usinage de précision ou le spatial [IBR02].

la commande vectorielle basée sur une régulation classique Proportionnel-Intégral (PI), associe dans sa structure des termes de compensation qui permettent de découpler l'axe d qui sera utilisé réglage du flux, de l'axe q au réglage du couple. Cette configuration permet de réaliser des systèmes d'actionnement électriques ayant les performances exigées par les domaines d'application [BOU99].

Ce chapitre présente la commande vectorielle appliquée à la MSAP. La boucle de courant et la boucle de vitesse sont régulées à l'aide de correcteurs PI. Des résultats de simulation sont présentés pour montrer les performances de la régulation proposée. L'étude de la robustesse de ce contrôle sera examinée en dernier lieu.

II.2 PRINCIPE DE LA COMMANDE VECTORIELLE

La commande vectorielle, consiste à régler le flux par une composante du courant et le couple par l'autre composante. Il faut donc, choisir un système d'axe d, q et une loi de commande qui assure le découplage du flux et du couple

La stratégie de commande la plus souvent utilisée est celle qui consiste à maintenir le courant i_d à une valeur nulle et à réguler la vitesse par le courant i_q via la tension v_q . Cette stratégie permet de simplifier la commande du couple par la linéarisation de la relation entre le couple et le courant, de travailler à couple maximal et de minimiser les pertes [SEI88] [NAH01].

Si le courant i_d est maintenu nul, physiquement le flux de réaction est en quadrature avec le flux rotorique produit par les aimants permanents et $\psi_d = \psi_f$.

L'expression du couple donnée par la relation (I.15), devient :

$$\Gamma_{em} = p \psi_f i_q \quad (\text{II.1})$$

Les équations de tension (I.14) de la machine, deviennent quand à elles :

$$\begin{aligned} v_d &= -p\omega_r L_q i_q \\ v_q &= R_s i_q + L_q \frac{di_q}{dt} + p\omega_r \psi_f \end{aligned} \quad (\text{II.2})$$

La variation du couple est proportionnel à celle du courant, donc le modèle de la machine se réduit à celui d'une machine à courant continu à excitation indépendante [Fu95] (figure II.2).

Lorsque le courant i_d est nul, la vitesse est contrôlée par la composante i_q , ce qui a pour effet de diminuer les pertes joules et de travailler à couple maximum.

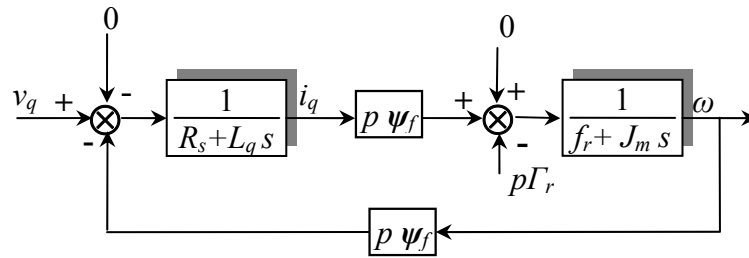


Figure II.1: Représentation du modèle découplé de la MSAP

Le contrôle de la vitesse de la machine, demande alors une commande simultanée de deux variables i_d et i_q . Le système comporte donc une boucle de régulation de vitesse, imposant la référence de courant i_q et le courant i_d est imposé égale à 0 par des régulateurs PI (proportionnel intégral) afin d’avoir une erreur statique nulle dans le régime permanent. L’entraînement à vitesse variable en appliquant la commande vectorielle est représentée par la figure II.2.

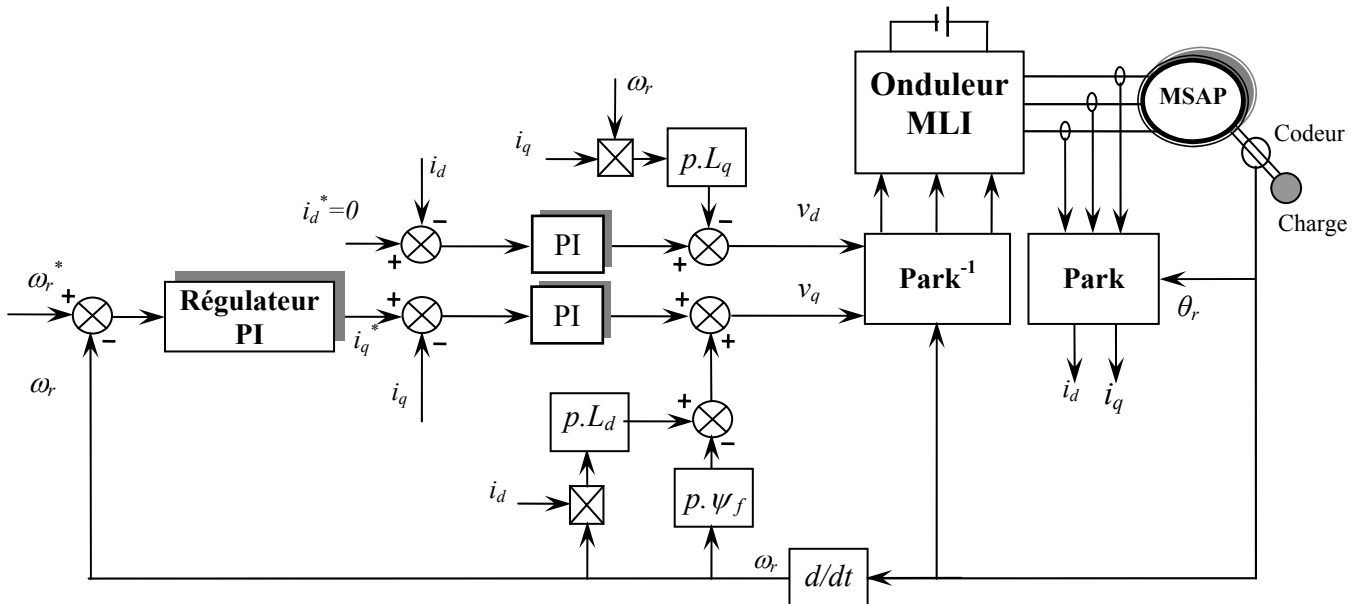


Figure II.2 : Schéma de la commande en vitesse de la MSAP

II.3 COMMANDE VECTORIEL AVEC REGLAGE DU COURANT

Le rôle des régulateurs est de maintenir une grandeur de sortie égale à la grandeur de référence imposé malgré la présence des perturbations internes ou externes. Les régulateurs de courant et de vitesse peuvent être synthétisés au moyen des techniques classiques développées pour les systèmes linéaires. [BER02]

II.3.1 Synthèse des régulateurs de courants

Connaissant le modèle de la machine dans le référentiel de *Park*, nous pouvons réaliser le réglage des courants i_d et i_q , en introduisant le régulateur proportionnel intégral (PI). Le schéma de la boucle de courant est alors représenté par la figure II. 3.

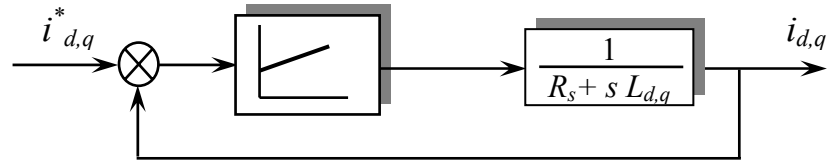


Figure II.3 : Boucle de réglage du courant

La fonction de transfert du régulateur est donnée, sous forme générale, par l'équation (II.3) pour les deux axes d et q :

$$F_i(S) = K_p + \frac{K_i}{S} \quad (\text{II.3})$$

En boucle fermée la fonction de transfert est donnée par :

$$F_i(S) = \frac{i(S)}{i^*(S)} = \frac{K_p S + K_i}{L S^2 + (R_s + K_p)S + K_i} \quad (\text{II.4})$$

Le polynôme caractéristique est donné par :

$$p(S) = L S^2 + (R_s + K_p)S + K_i \quad (\text{II.5})$$

Deux pôles complexes conjugués sont généralement imposés : $S_{1,2} = \rho(-1 \pm j)$;

Ce qui conduit aux paramètres du régulateur :

$$\begin{aligned} K_p &= 2L \rho - R_s \\ K_i &= 2L \rho^2 \end{aligned} \quad (\text{II.6})$$

II.3.2 Synthèse du régulateur de vitesse

La boucle de régulation de la vitesse est représentée sur la figure II.4. La fonction de transfert du régulateur en boucle fermée est calculée par :

$$F_f(S) = \frac{F_r(S) F_o(S)}{1 + F_r(S) F_o(S)} \quad (\text{II.7})$$

$F_r(S)$ est la fonction de transfert du régulateur de la vitesse est donnée par :

$$F_r(S) = K_{p\Omega} + \frac{K_{i\Omega}}{S} \quad (\text{II.8})$$

La fonction de transfert en boucle ouverte est exprimée par :

$$F_o(S) = \frac{p \psi_d}{f_c (1 + S \tau_q) (1 + S \tau_m)} \quad (\text{II.9})$$

τ_q est la constante de temps électrique de la machine et est égale à L_q / R_s

τ_m est la constante de temps mécanique de la machine égale à J / f_c

La fonction de transfert trouvée est de la forme suivante :

$$F_r(S) = \frac{p \psi_f (S K_{p\Omega} + K_{i\Omega})}{J \tau_q S^3 + (J + f_c \tau_q) S^2 + (p \psi_f K_{p\Omega} + f_c) S + p \psi_f K_{i\Omega}} \quad (\text{II.10})$$

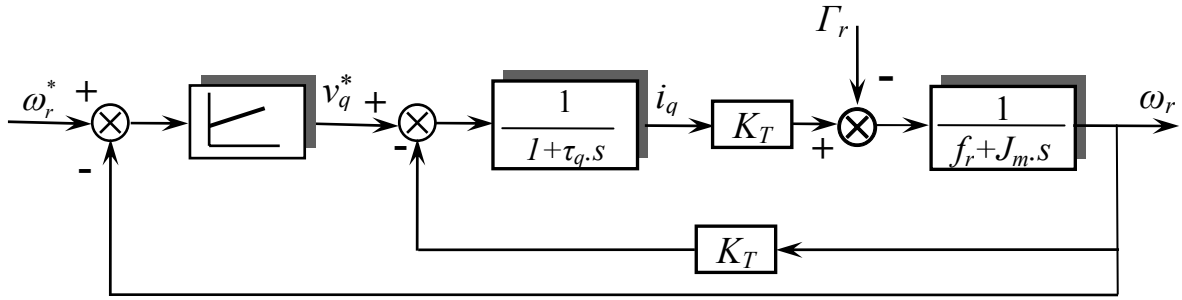


Figure II.4 : Boucle de régulation de la vitesse

En négligeant les termes $J\tau_q$ et $f_c\tau_q$ devant J , le polynôme caractéristique de cette fonction devient :

$$P(s) = J_m S^2 + (p\psi_f K_{p\Omega} + f_r)S + p\psi_f K_{i\Omega} \quad (II.11)$$

Pour obtenir les solutions de cette fonction, nous lui imposons deux pôles complexes conjuguée à son polynôme caractéristique $S_{1,2} = \rho(-1 \pm j)$, ce qui donne :

$$p(S) = S^2 + 2\rho S + 2\rho^2 \quad (II.12)$$

Par identification terme à terme, à partir des deux dernières équations, nous obtenons les paramètres du régulateur:

$$K_{p\Omega} = \frac{2\rho J - f_c}{p\psi_f} \quad \text{et} \quad K_{i\Omega} = \frac{2\rho^2 J}{p\psi_f} \quad (II.13)$$

II.4 SIMULATION DE LA COMMANDE EN VITESSE DE LA MACHINE

Après réalisation du découplage et synthèse des régulateurs, nous effectuons la simulation de la commande en vitesse par régulateur PI de la machine M1. Le système est simulé sous Matlab-Simulink, en prenant un simple gain comme modèle de l'onduleur.

Dans la figure II.5 le test de simulation présente une application d'un couple résistant entre les instants 0,7 et 1,4 secondes. Nous constatons l'établissement de la vitesse à sa valeur de référence en un temps de réponse satisfaisant et sans dépassement avec une erreur nulle au régime permanent. Le test montre un rejet de la perturbation sans dépassement avec maintien de la vitesse de référence. Le découplage est parfaitement réalisé.

Nous avons simulé le démarrage de la machine, avec poursuite de variation de la consigne en figure II.6. Nous remarquons un établissement de la vitesse à la valeur désirée en un faible temps de réponse et sans dépassement, une très bonne poursuite pour l'établissement de la vitesse à la valeur de référence, l'erreur en régime permanent est nulle.

Enfin, la variation de la consigne avec application d'un couple résistant, est représentée dans les figure II.7 et II.8. Ces tests montrent l'établissement sans dépassement de la vitesse de référence et le rejet de la perturbation. La poursuite de la variation de la consigne se fait avec des temps de réponse rapides et sans dépassement grâce à l'action rapide du PI.

Nous déduisons suite aux tests effectués, que le régulateur PI utilisé est bien paramétré, du fait qu'il donne de bonnes performances de poursuite et de rejet rapide de la perturbation de charge.

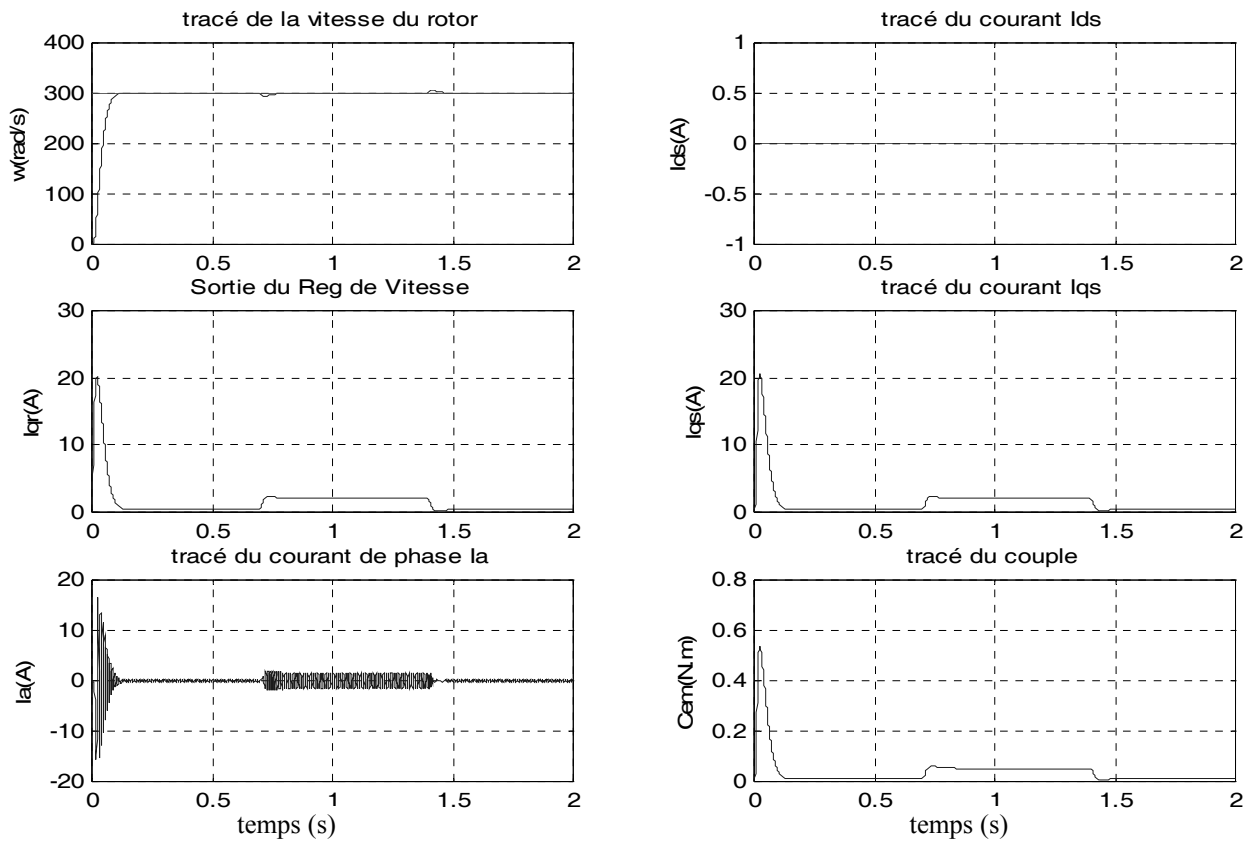


Figure II.5: Commande en vitesse de la MSAP avec application d'un couple résistant

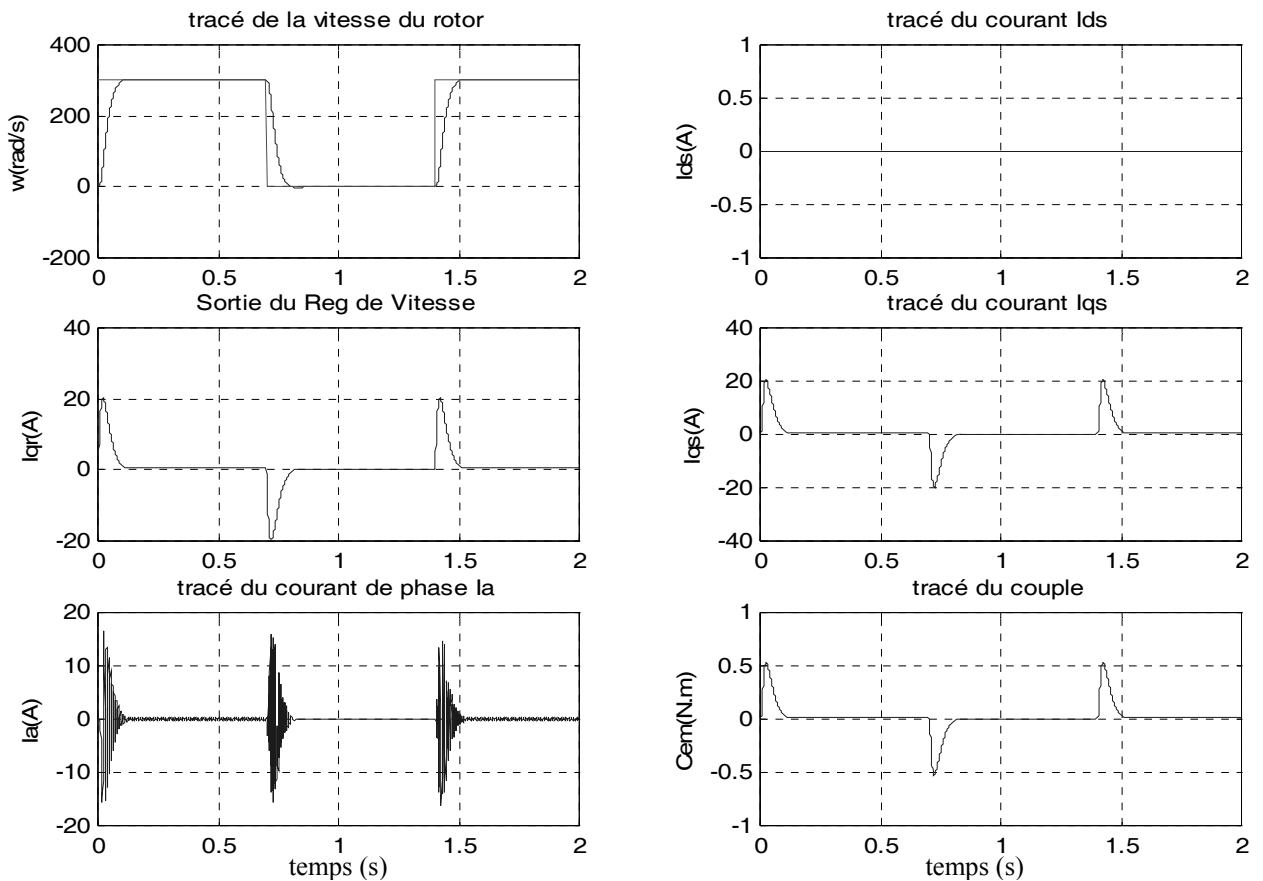


Figure II.6: Réponse de la commande pour une variation de consigne

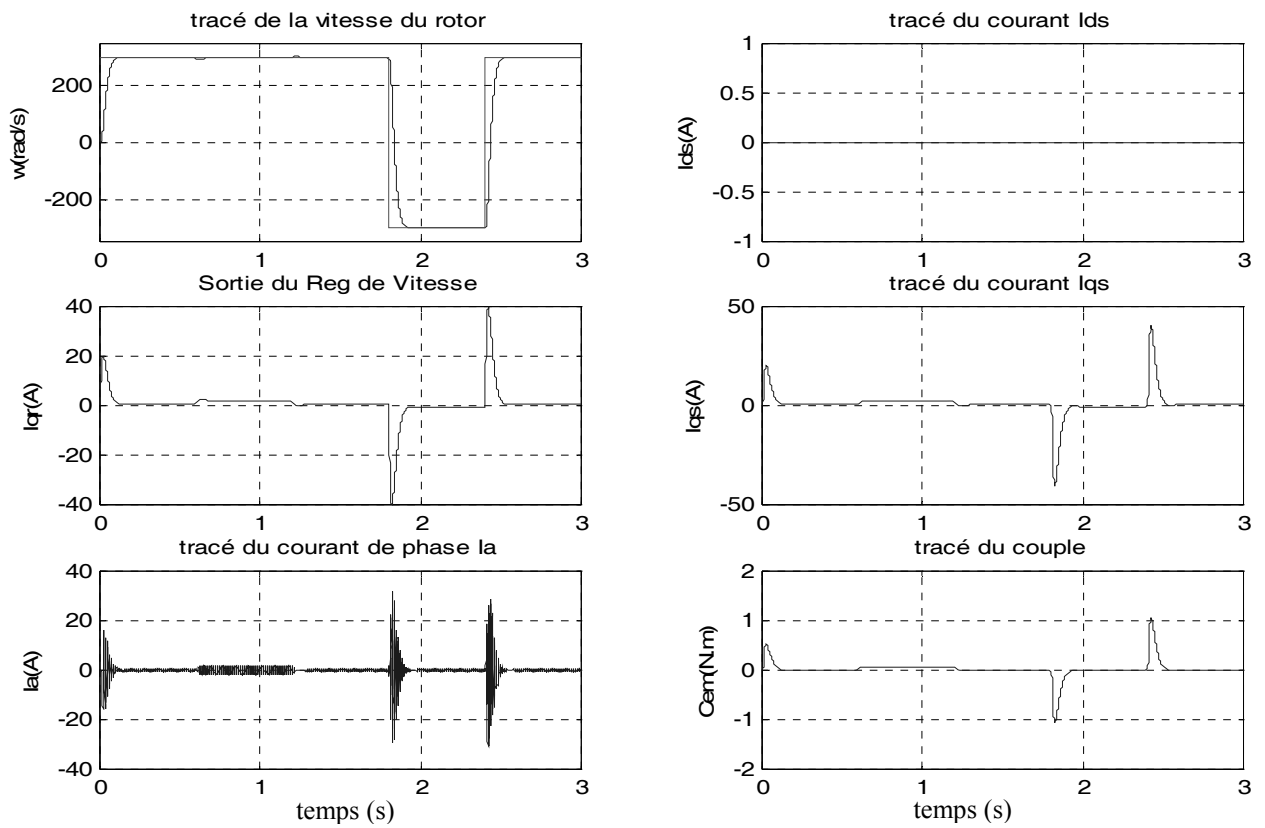


Figure II.7: Simulation de la commande en vitesse de la MSAP avec application du couple résistant puis variation consigne (300 à -300 rad/sec)

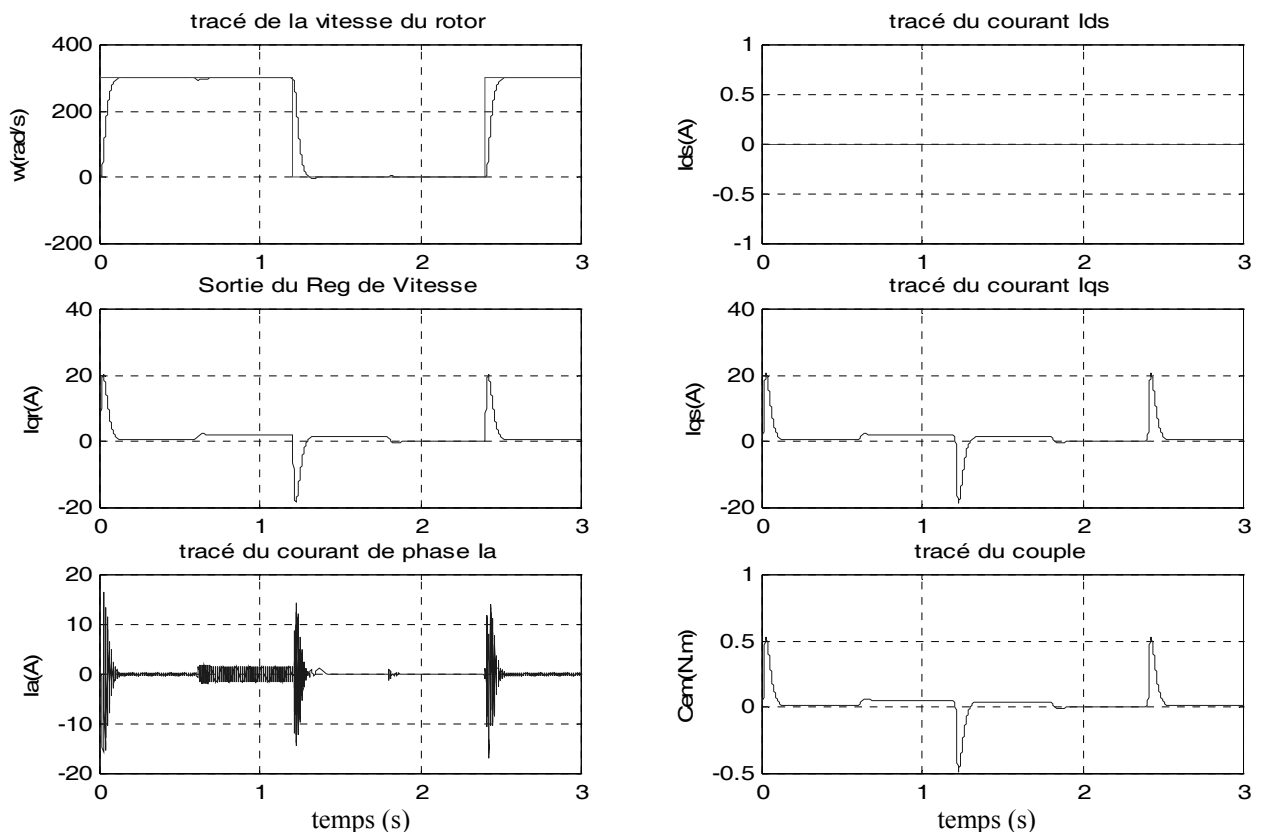


Figure II.8: Simulation de la commande en vitesse de la MSAP avec variation de consigne en présence d'une perturbation du couple résistant

II.5 ETUDE DE ROBUSTESSE AUX VARIATIONS PARAMETRIQUES

Dans la précédente série de tests, nous avons simulé les performances du régulateur PI pour la poursuite et le rejet de perturbation. Nous testons à présent sa robustesse, afin de vérifier son comportement face à des variations paramétriques de la machine. Nous appliquons donc une variation de la résistance, de l'inductance et du moment d'inertie. Ces tests de robustesse sont représentés par les figures II.9 à II.17.

La figure II.9 représente le test de la commande pour une augmentation de 100% de la résistance statorique. Ce test présente un dépassement de l'ordre de 4.6% lors de l'établissement de la vitesse de référence. Le couple résistant appliqué à l'instant 0.7 seconde. Le rejet et la stabilisation de la sortie sont relativement longs que lorsque la variation de la résistance n'était pas appliquée.

Dans la figure II.10, nous simulons la robustesse de la commande pour une poursuite de la consigne avec une variation de 100% de la résistance. Nous remarquons un dépassement à l'établissement de la vitesse après démarrage, et aussi lors de la variation de vitesse de consigne entre 0.7s et 1.4s de 4.5% avec un temps de réponse légèrement plus lent par rapport aux conditions normales.

Nous appliquons dans la figure II.11 une perturbation de la charge suivie d'une variation de la vitesse de consigne. Nous constatons aussi des dépassements de l'ordre de 4.8% avant d'atteindre la vitesse de référence. Plus le dépassement est important plus le retour à la référence est plus lent ce qui impose un temps de réponse plus lent. L'erreur en régime permanent dans ce cas est non nulle.

La figure II.12, et II.13 présentent la réponse du régulateur PI à une variation de 100% de la valeur de l'inductance. Une perturbation de charge est appliquée dans le premier test et une variation de la consigne de vitesse dans le second. Nous remarquons que cette variation n'influe pas du tout sur la qualité de l'asservissement.

La figure II.14 représente quand à elle, la réponse du régulateur par rapport à la variation de 100% de l'inductance L_q . Nous remarquons que ce soit pour l'application de la perturbation de charge, ou pour la poursuite de la consigne, l'inexistence de dépassements et la réaction rapide du PI face aux applications de la charge. Nous constatons que ce paramètre est sans influence sur les performances de réglage.

Nous simulons dans les figures II.15 et II.16, le test de robustesse de la commande par rapport à une variation de 50% du moment d'inertie. Nous remarquons un dépassement de l'ordre de 4.1%, après démarrage de la machine. L'application du couple résistant montre la réaction du régulateur PI par le rejet de celui-ci et une erreur en régime permanent nulle.

Enfin la figure II.17, montre encore une fois la perte de la robustesse due à l'application d'une augmentation de 100% du moment d'inertie, qui se traduit par les dépassements et l'instabilité du régime permanent.

Nous déduisons donc que la réponse du régulateur PI, est beaucoup plus affecté par des variations de la résistance et par l'augmentation du moment d'inertie, que par l'augmentation de l'inductance.

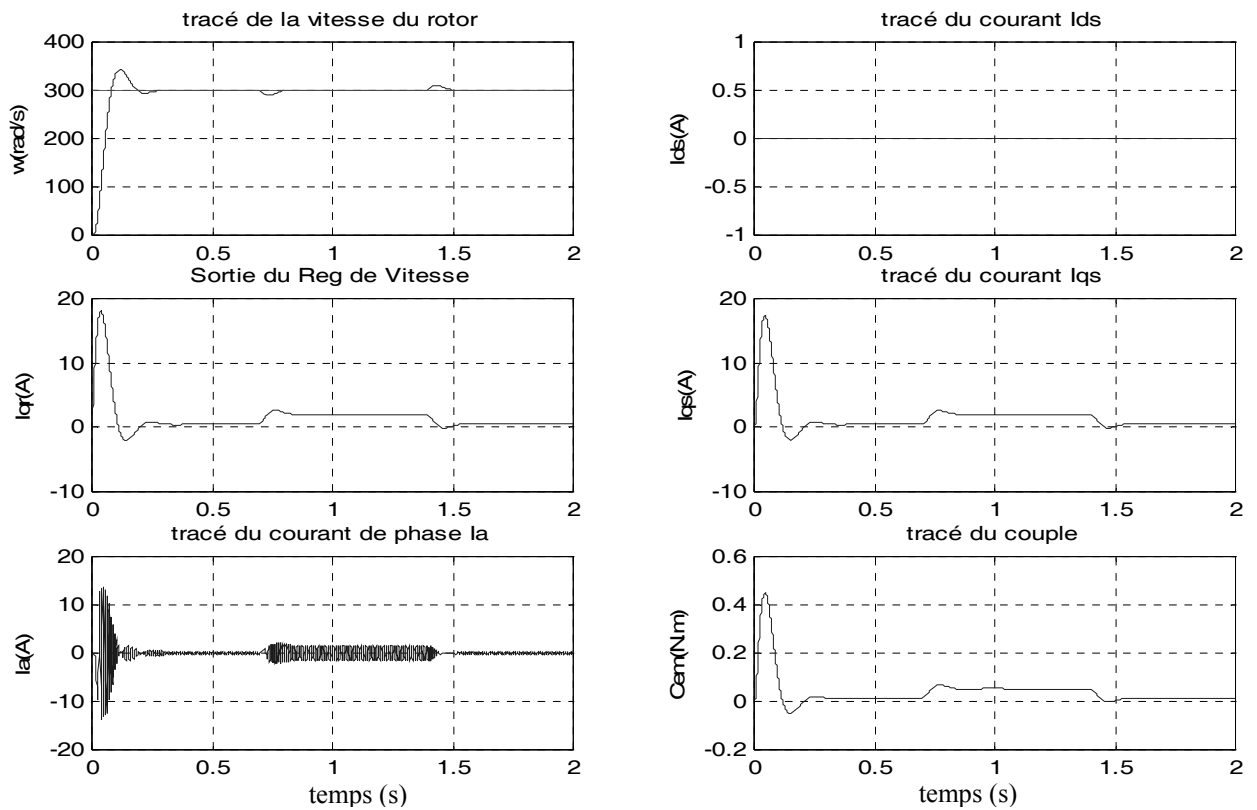


Figure II.9: Robustesse de la commande en vitesse pour une variation de 100% de la résistance R_s en présence d'une perturbation du couple résistant

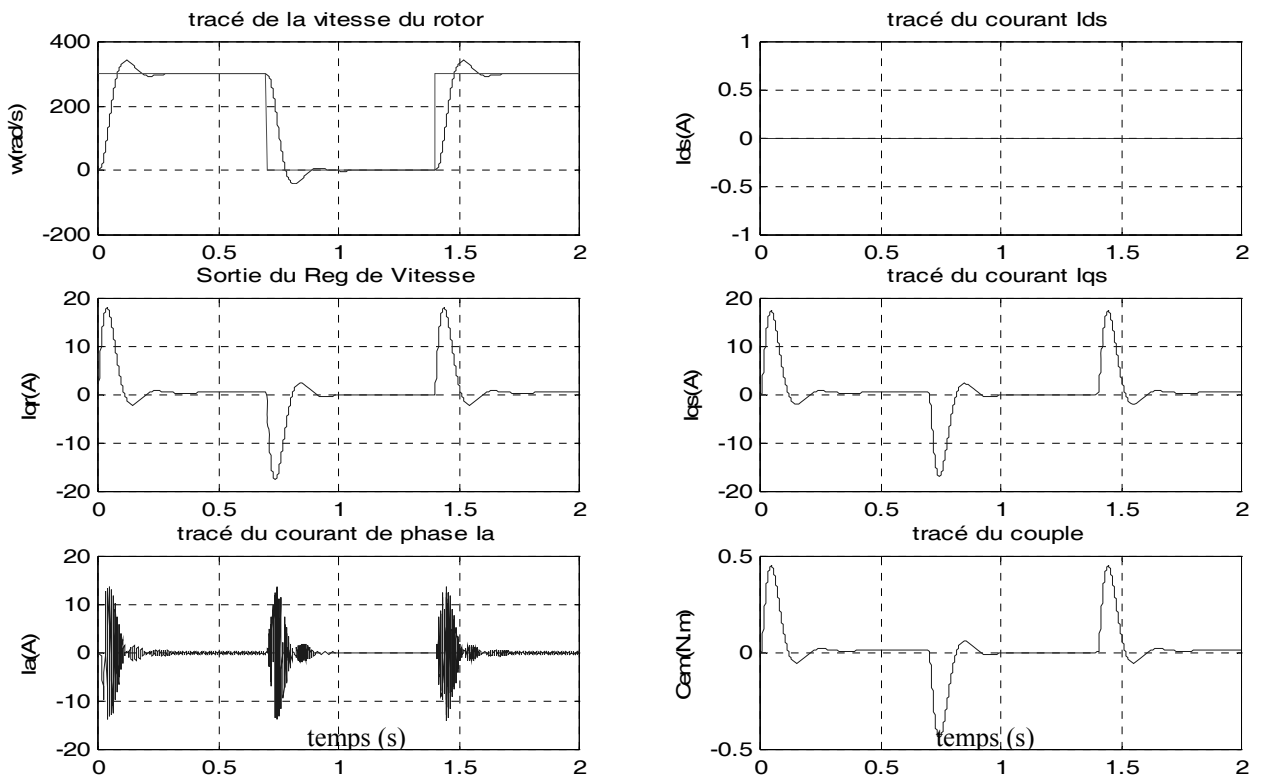


Figure II.10: Robustesse de la commande en vitesse pour une variation de 100% de la résistance R_s et pour une poursuite de la variation de consigne

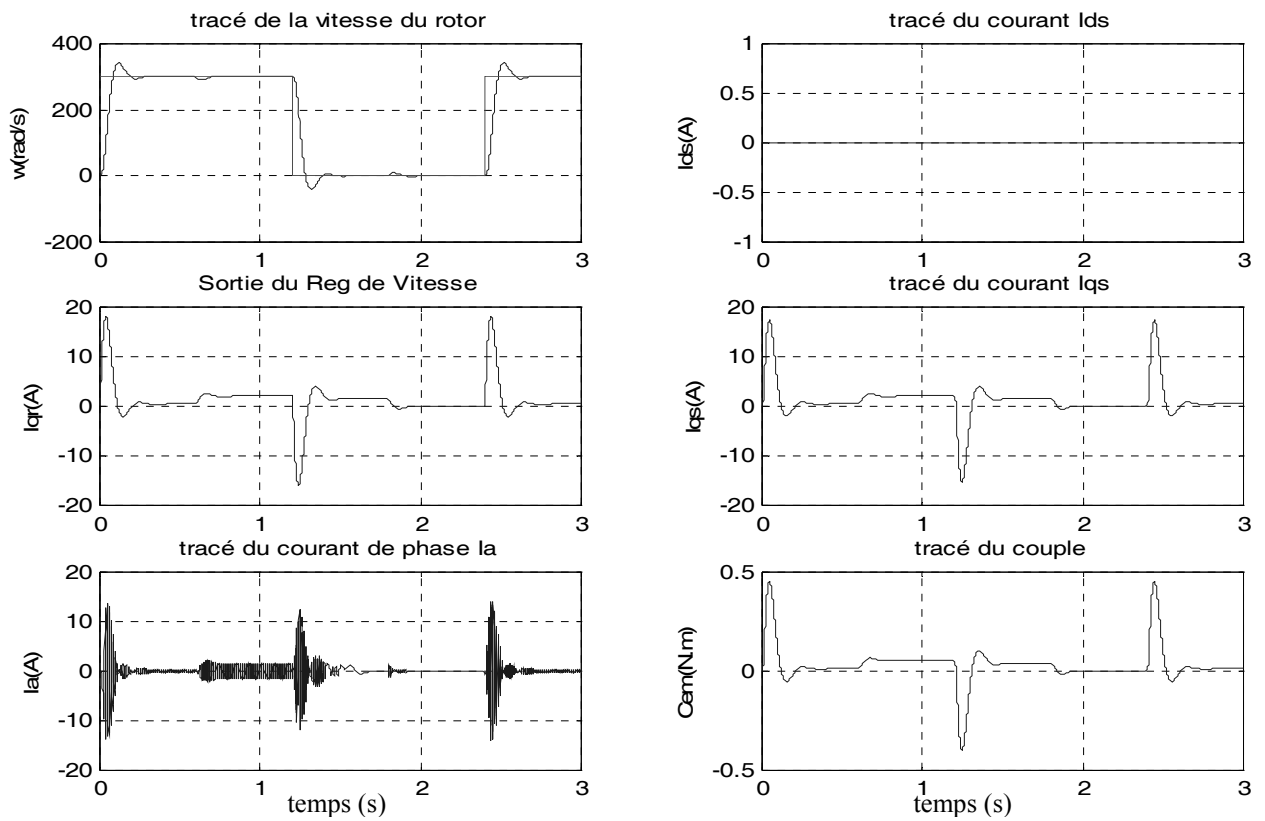


Figure II.11: Robustesse de la commande pour une variation de 100% de la résistance R_s et poursuite de la variation de consigne avec application du couple résistant

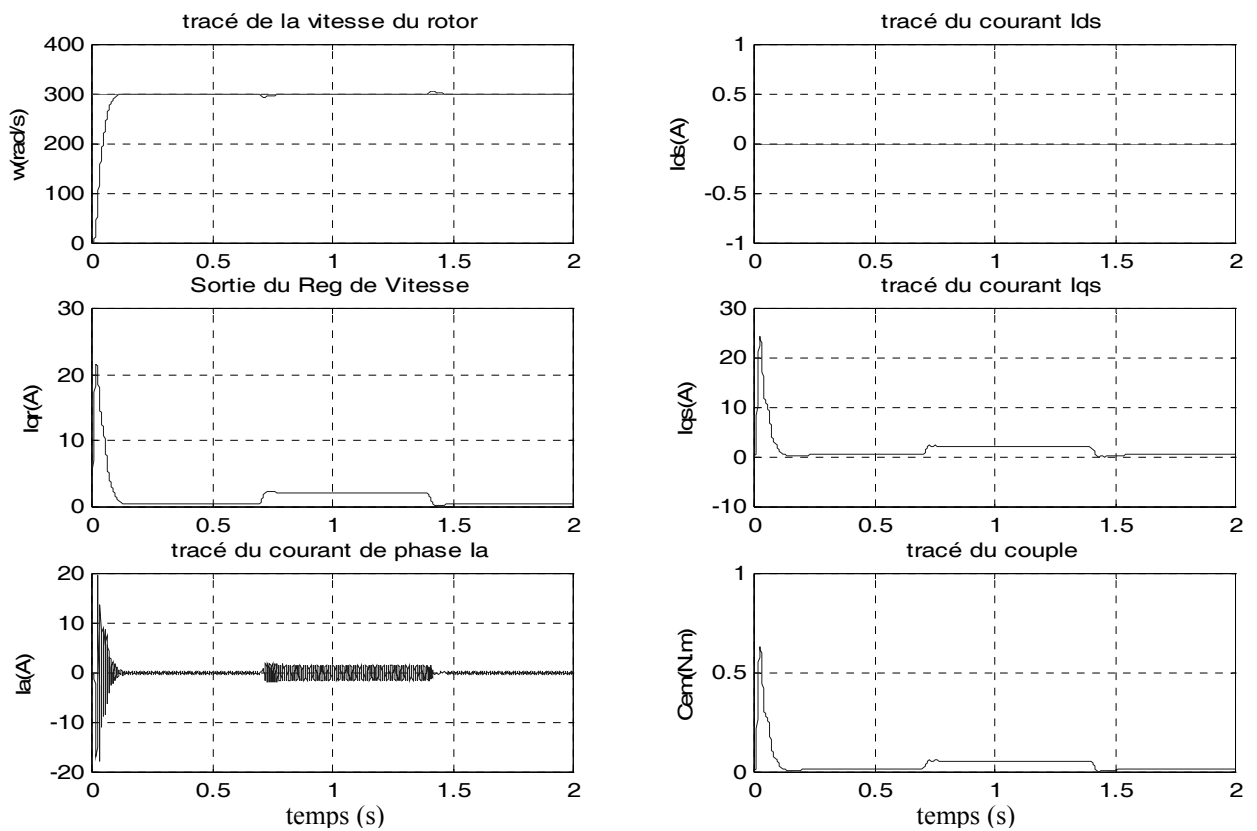


Figure II.12: Robustesse de la commande en vitesse pour une variation de 100% de l'inductance L_q en présence d'une perturbation du couple résistant

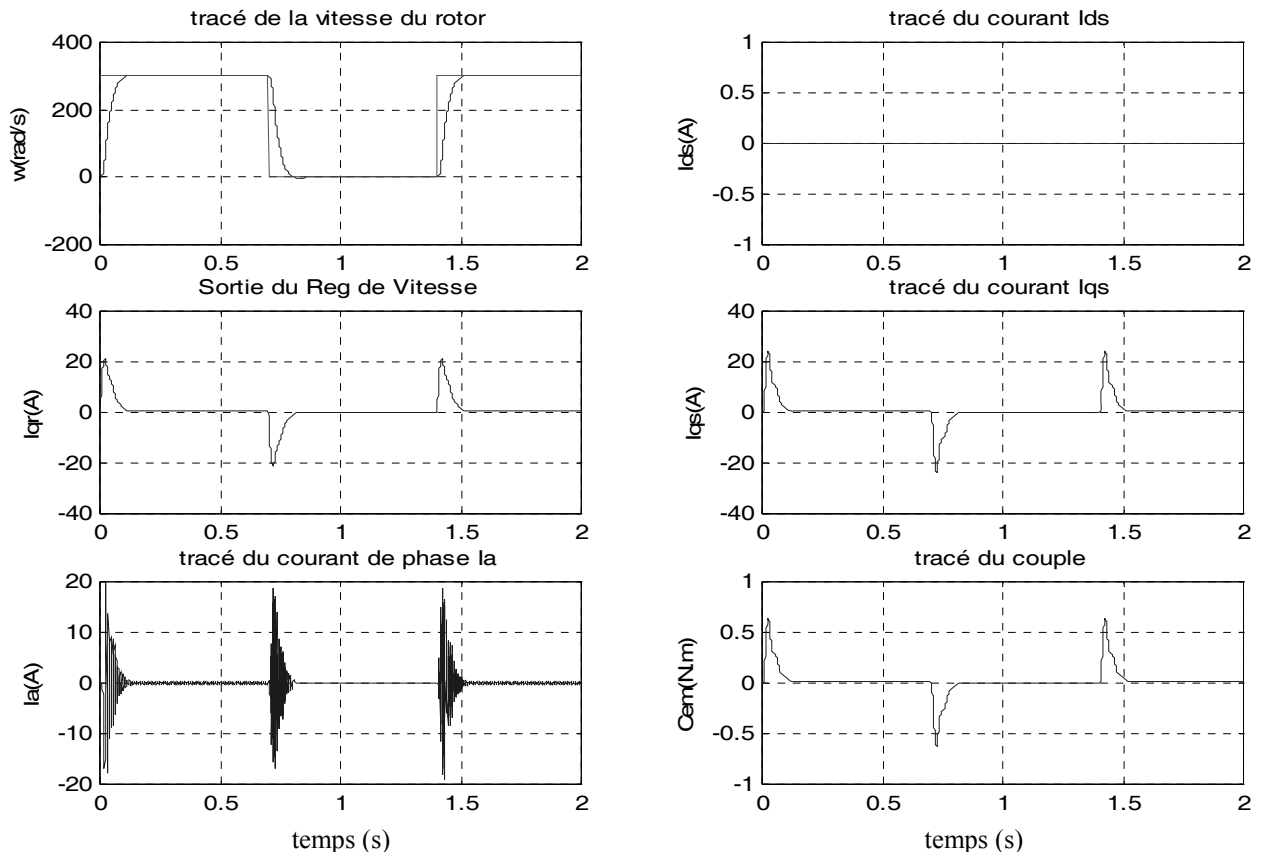


Figure II.13: Robustesse de la commande en vitesse pour une variation de 100% l'inductance L_q et pour une poursuite de la variation de consigne

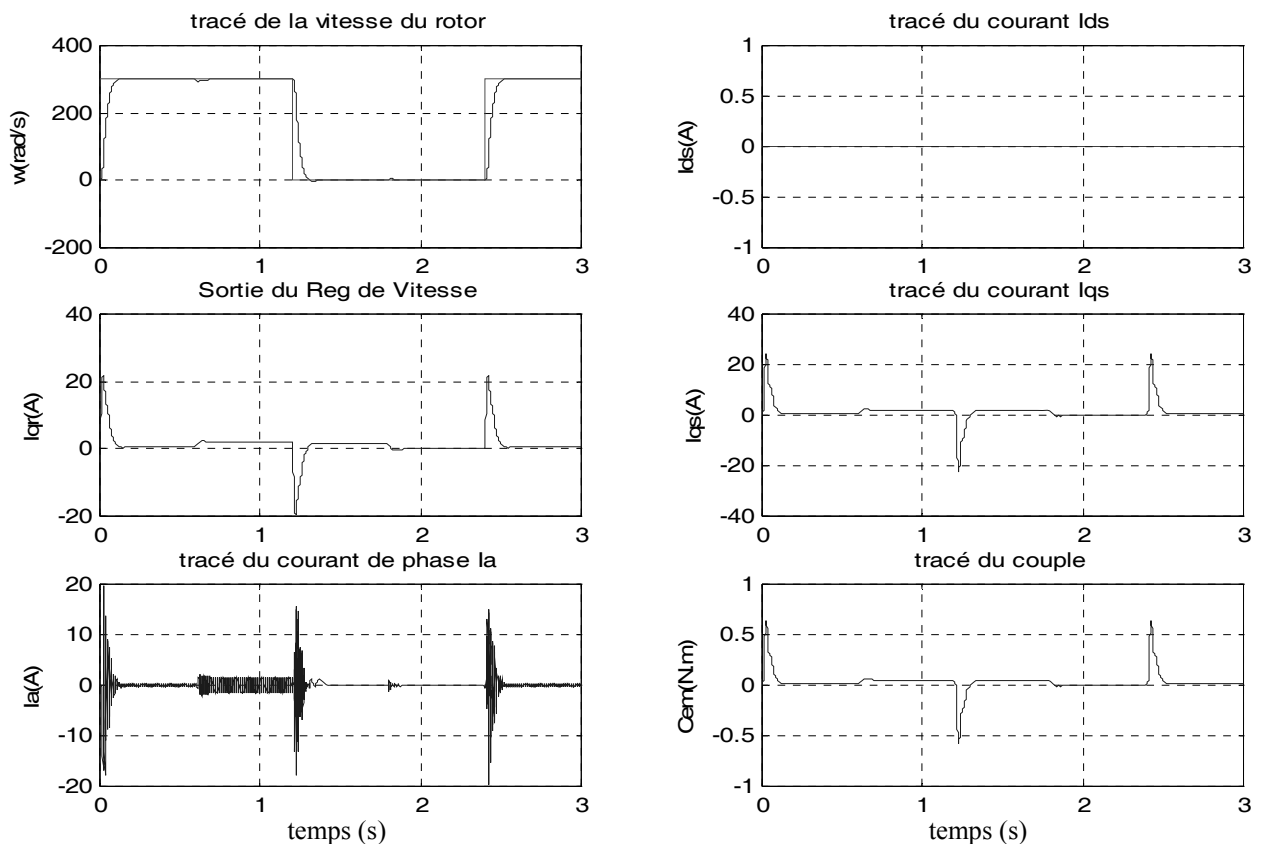


Figure II.14: Robustesse de la commande pour une variation de 100% l'inductance L_q et poursuite de la variation de consigne avec application du couple résistant

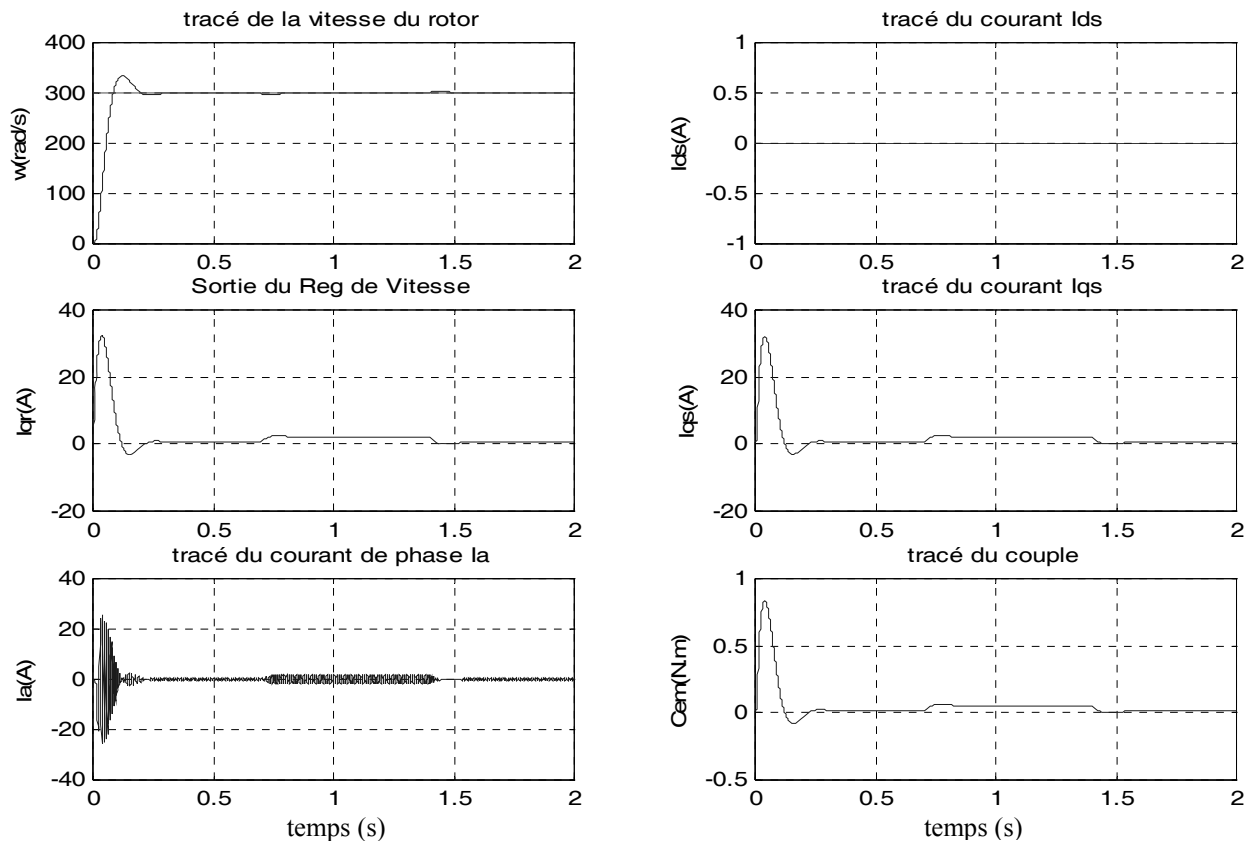


Figure II.15: Robustesse de la commande en vitesse pour une variation de 100% du moment d'inertie avec présence d'une perturbation du couple résistant

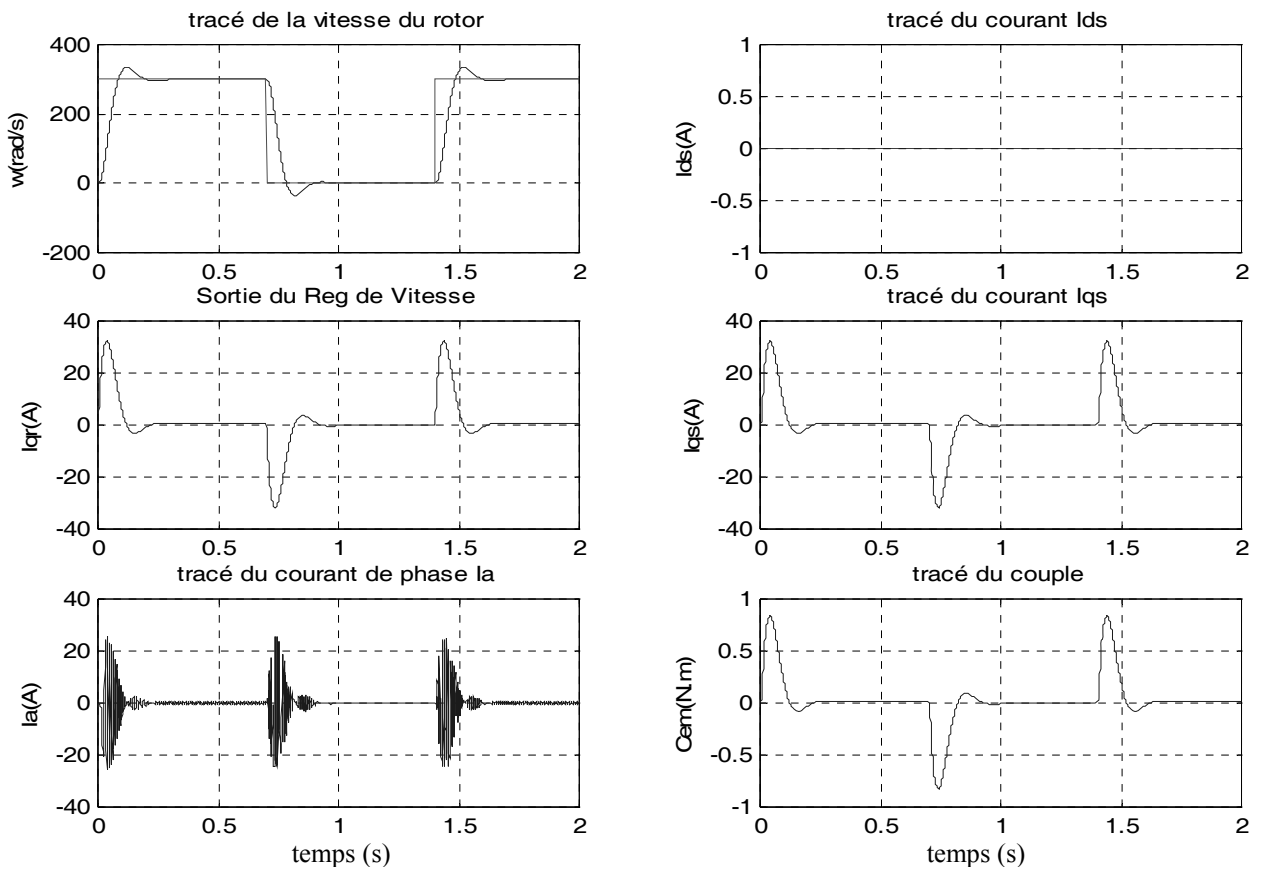


Figure II.16: Robustesse du régulateur PI pour une variation de 100% du moment d'inertie avec présence d'une variation de la consigne

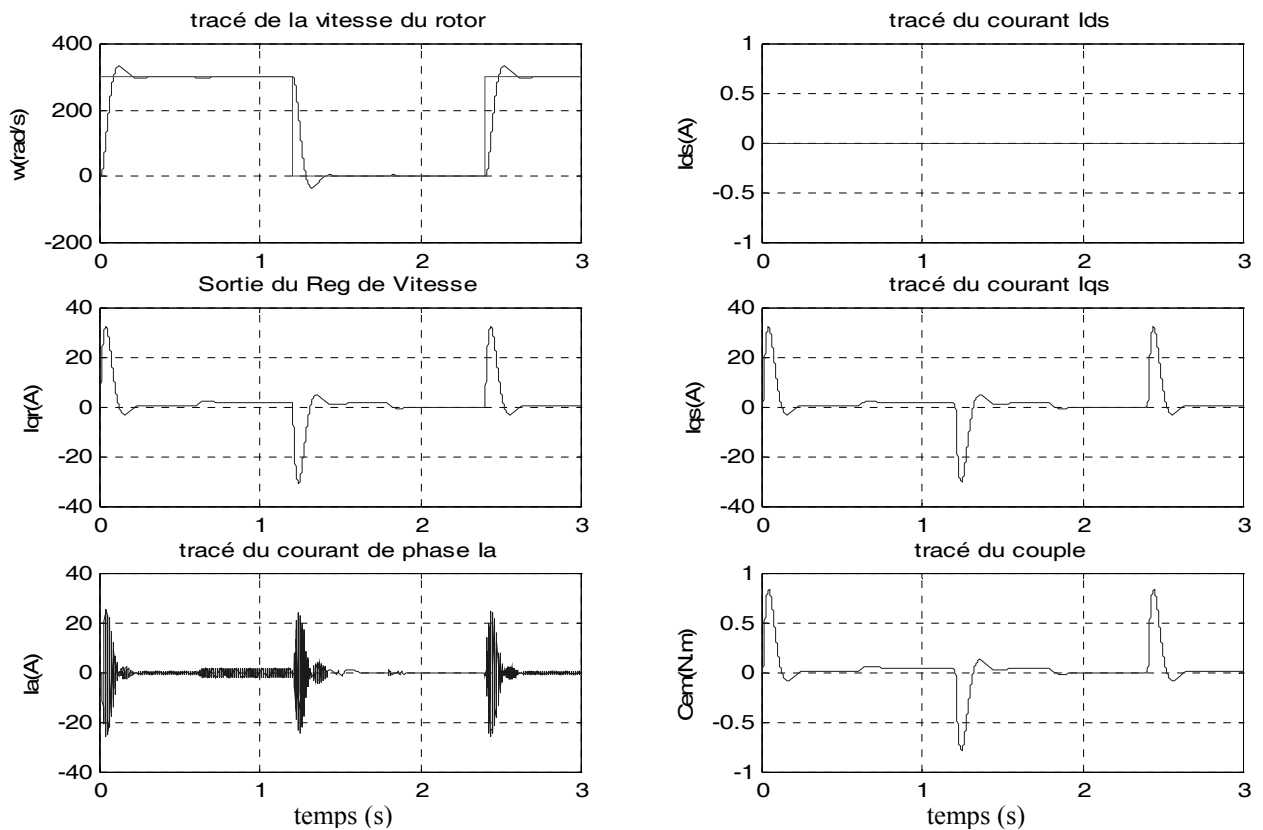


Figure II.17 : Robustesse de la commande en vitesse pour une variation de 100% du moment d'inertie pour la poursuite de la consigne et la perturbation du couple résistant

II.6 CONCLUSION

La commande vectorielle des moteurs à courant alternatif alimentés par onduleur MLI de tension permet de réaliser des systèmes d'actionnement électriques ayant les performances exigées par les domaines d'application.

Tant que le modèle de la machine reste invariant dans le temps, le régulateur classique insérer dans le circuit de commande, afin de réaliser la régulation en vitesse de la machine, donne des résultats satisfaisants. Lorsque le modèle subit des changements paramétriques, les performances de réglage se détériorent et le système perd sa robustesse. Ce qui est traduit par les dépassements observés de la vitesse, lors des tests de poursuite de trajectoire et de rejet de perturbation de la charge.

L'incapacité d'adaptation du PI à ce genre de perturbations, fait qu'il est nécessaire de mettre en place une stratégie de commande permettant de répondre aux critères imposés par l'industriel. Pour réaliser la commande en vitesse de la machine, nous nous intéressons dans la suite de ce travail à de nouvelles techniques de commande. Les techniques qui feront l'objet d'étude des prochains chapitres sont principalement la commande adaptative et la commande par les réseaux de neurones artificiels.

CHAPITRE

-III-

LA COMMANDE ADAPTATIVE

III.1 INTRODUCTION

La motivation réelle de la commande adaptative est d'apporter une solution au contrôle des systèmes dont le comportement dynamique est partiellement ou totalement inconnu et éventuellement variable dans le temps. En effet, dans ces situations, les méthodes classiques de commande n'aboutissent pas aux performances désirées en boucle fermée, et conduisent à une détérioration de la robustesse du régulateur.

En automatique, il est difficile de donner une définition précise à l'adaptation. Cependant, le sens retenu est de dire qu'un régulateur est adaptatif s'il peut modifier son comportement en réponse à des changements dans la dynamique du procédé ou des perturbations. Cette modification doit se faire d'une manière automatique, sans intervention externe. Le régulateur adaptatif, est défini par son aptitude à maintenir, dans ces conditions, un certain niveau de performance préalablement choisi [ÅST89].

Lorsque le système de commande est non-linéaire et que les paramètres de son modèle de connaissance varient dans de grandes proportions, la commande adaptative est une alternative intéressante qui permet de prendre en compte ces problèmes.

Plusieurs possibilités de commande adaptative peuvent être envisagées. Nous présentons, dans ce chapitre, la commande des systèmes linéaires, dont les paramètres sont inconnus ou variables dans le temps. Le régulateur comporte un algorithme d'identification récursif, permettant d'estimer en ligne les paramètres de la fonction de transfert échantillonnée équivalente, combiné avec une stratégie de commande. Ce qui donne des régulateurs auto-ajustables ou régulateurs adaptatifs [ÅST89].

Nous nous proposons de faire la synthèse de deux stratégies de commande. La première stratégie, consiste en la commande à erreur de prédiction minimale [GOO84]. La seconde est la commande à variance minimale généralisée [CLA75], [CLA79]. Ces stratégies sont liées aux hypothèses faites sur les zéros du système échantillonné et aux objectifs de la commande.

III.2 COMMANDE A CONTRE-REACTION ET COMMANDE ADAPTATIVE

La structure de base de la commande adaptative est constituée de deux boucles de régulation. On retrouve la boucle classique à contre réaction comportant le procédé et le régulateur à paramètres ajustables. La seconde boucle permet de réajuster en ligne les paramètres du régulateur en fonction d'un certain indice de performance. Cet indice regroupe les objectifs désirés [LAN86].

La boucle à contre réaction, prend en charge l'effet des perturbations agissant sur les variables à réguler. La boucle d'adaptation quand à elle, permet de diminuer l'effet des perturbations paramétriques agissant sur le procédé lui même, tout en gardant les objectifs fixés.

Le schéma de base de la commande adaptative est représenté par la figure (III.1).

III.3 DIFFERENTES TECHNIQUES DE COMMANDE ADAPTATIVE

Parmi les méthodes de commande adaptative, deux principales approches ont attiré le plus d'attention [ÅST89]:

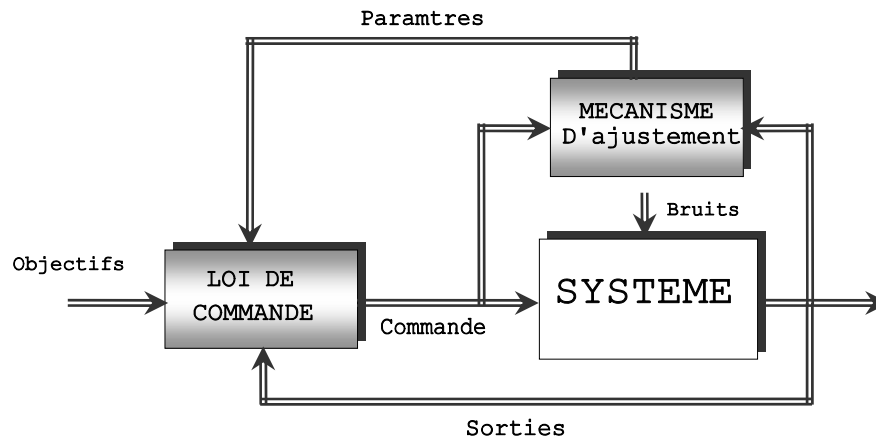


Figure III.1: Structure de base d'une commande adaptative

- ☐ La commande adaptative à modèle de référence (**Model Reference Adaptive Control "M.R.A.C"**).
- ☐ Les systèmes de commande auto-ajustable (**Self-Tuning Regulator "S.T.R"**).

La principale différence qui existe entre les deux approches réside dans le mode d'ajustement des paramètres du régulateur. De plus, les systèmes MRAC sont initialement développés pour les problèmes de poursuite en temps continu déterministe et la technique STR pour les problèmes de régulation discrète stochastique [LAN81].

Après la présentation des généralités sur ces deux approches, nous nous intéresserons, aux régulateurs auto-ajustables.

III.3.1 Commande adaptative à modèle de référence

Les systèmes (MRAC) sont importants puisqu'ils sont relativement simples à implémenter et qu'ils n'ont pas besoin d'une identification des performances de la dynamique du procédé. L'idée de base de ces systèmes est de synthétiser le signal de commande qui forcera le système de se conduire d'une certaine manière prescrite par un modèle de référence idéalement choisi (Figure III.2) [CHA87], [LAN79].

Dans cette approche, la comparaison entre l'indice de performance (IP) désiré et celui mesuré est obtenu directement en comparant la sortie du modèle de référence avec celle du système

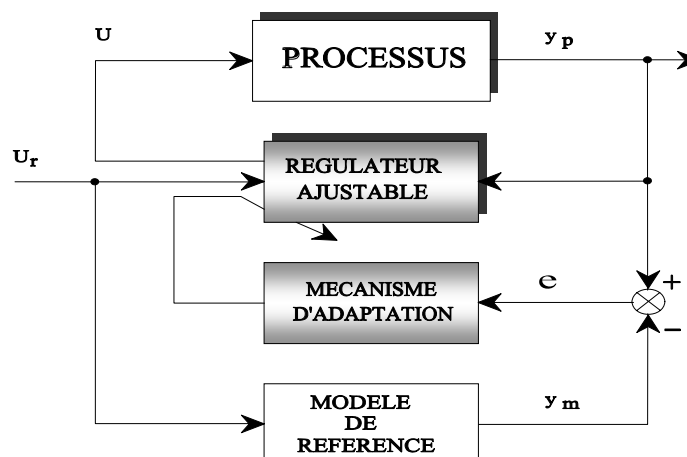


Figure III.2: Commande adaptative à modèle de référence.

à commander. L'erreur est utilisée par la suite par le mécanisme d'adaptation pour l'ajustement des paramètres du régulateur. Ce schéma représente les systèmes (MRAC) explicite (appelés aussi systèmes de commande adaptative directe).

Il existe des systèmes (MRAC) implicite (appelés aussi systèmes de commande adaptative indirecte), dans lesquels, un prédictif adaptatif dérivé des systèmes adaptatifs à modèle de référence (SAMR) est utilisé [LAN80], [LAN81], [SIL79]. Les paramètres de ce dernier servent pour le calcul du régulateur (Figure III.3). L'ensemble prédictif adaptatif-contrôleur se conduit à chaque instant comme le système à dynamique "désirée". Par ailleurs, il est démontré que pour une paramétrisation particulière du procédé, les deux aspects de commande conduisent aux mêmes équations [ÅST89], [GOO84].

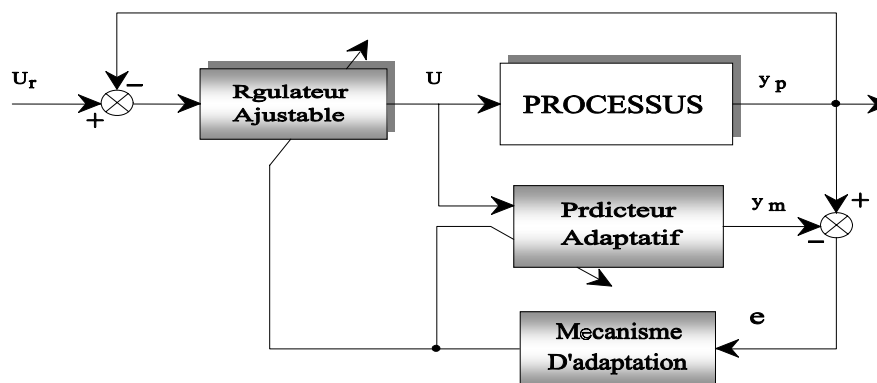


Figure III.3: Commande adaptative à modèle de référence implicite.

III.3.2 Commande auto-ajustable

Dans les systèmes adaptatifs, il est supposé que les paramètres du régulateur sont ajustés pendant tout le temps. Ceci implique que les paramètres du régulateur suivent les changements dans le processus. Le régulateur est appelé auto-ajustable du fait qu'il ajuste le contrôleur aux performances désirées [ÅST89].

Ce type de régulateur a été décrit par Kalman (1958). Un régulateur similaire, basé sur la commande à variance minimale, est publié par Wiestander et Wittenmark (1971), revu et développé en suite par Åström & Wittenmark (1973) [ÅST73].

Les régulateurs auto-ajustables se basent essentiellement sur l'estimation des paramètres du procédé. Les paramètres du régulateur sont calculés indirectement, via le prédictif (Figure III.4). C'est l'algorithme auto-ajustable indirect (explicit self-tuning control). Il est cependant possible de reparamétriser le modèle en termes de paramètres du régulateur de façon à ce qu'ils soient estimés directement (Figure III.5). Ce qui donne une simplification significative de l'algorithme. C'est le régulateur auto-ajustable direct (implicit self-tuning control) [ÅST73].

Nous allons utiliser deux de ces régulateurs auto-ajustables, à savoir:

- ☒ les régulateurs auto-ajustables à erreur de prédiction minimale.
- ☒ les régulateurs auto-ajustables à variance minimale généralisée.

L'approche que nous adopterons, pour les deux stratégies de commande, est l'approche directe. Elle a pour avantage d'éliminer l'étape de l'identification des paramètres du procédé. Ce qui nous facilite les calculs et réduit le temps d'exécution.

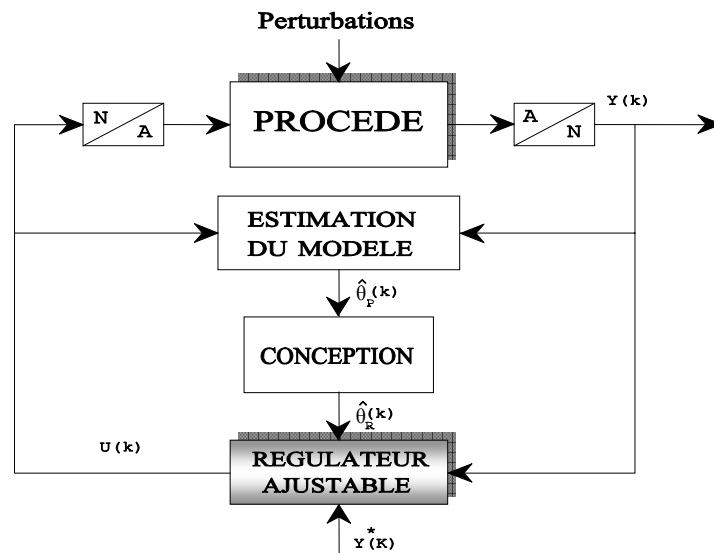


Figure III.4 : Commande auto-ajustable schéma de base

Le schéma de la commande adaptative à erreur de prédiction minimale est destiné à opérer dans un environnement déterministe et les régulateurs auto-ajustables à variance minimale généralisée sont destinés à opérer dans un environnement stochastique [ÅST89], [GOO84].

III.4 METHODES RECURSIVES D'IDENTIFICATION

Parmi l'ensemble des techniques d'identification développées depuis le début des années 60, les méthodes récursives ont rapidement vu le jour [LAN93a]. Leur principal objectif consiste à estimer, de manière consistante et en ligne, le modèle du processus étudié. Les deux raisons fondamentales de ce développement sont que :

- ☞ ces techniques permettent de suivre les évolutions (lentes) de la dynamique du système étudié en adaptant continuellement le modèle de procédé. Il est alors possible de prendre des décisions en temps réel en fonction des dernières informations acquises sur le processus;
- ☞ ces méthodes sont, la plupart du temps, moins gourmandes en ressource mémoire et en temps de calcul que leurs homologues hors ligne tout en étant capables de fournir un modèle fiable du procédé.

Comme la commande adaptative est articulée principalement sur une phase d'identification paramétrique, nous avons exposé dans le cadre de ce travail, deux méthodes d'identification paramétrique en Annexe A. Ces méthodes sont : l'Algorithme du gradient et les Moindres carrés récursifs.

III.5 COMMANDE AUTO-AJUSTABLE A VARIANCE MINIMALE

III.5.1 Principe de la commande

Le premier régulateur auto-ajustable développé par Åström & Wittenmark, est basé sur le principe de la variance minimale stochastique. C'est une approche utilisant l'objectif de la commande optimale. Les suppositions requises sont que le système soit à phase minimale, le temps de retard soit connu et qu'une limite de l'ordre du système soit donnée. L'objectif de cette approche est de minimiser la variance de la sortie mesurée à des instants d'échantillonnage [ÅST73].

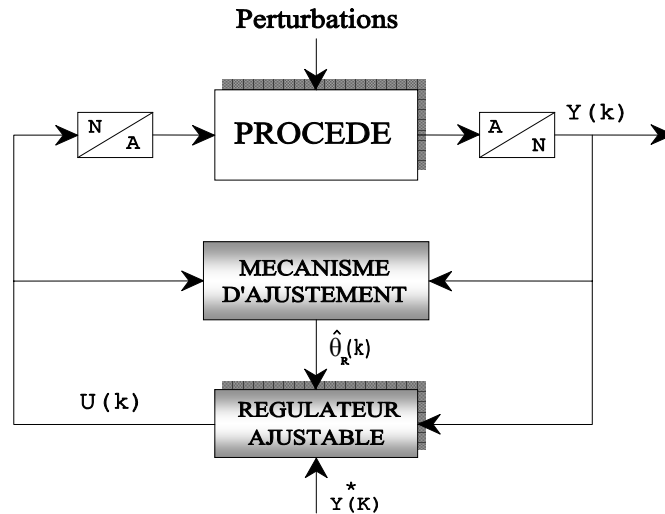


Figure III.5 : Cycle auto-ajustable implicite

Clarke et Gawthrop, ont étendu cette idée pour couvrir une large variété d'objectifs, en incluant les variations du point de fonctionnement de façon à assurer la faisabilité de la poursuite et de la régulation. Cela nécessite la minimisation d'une combinaison de la variance de la commande et celle de la sortie [CLA75], [CLA79].

III.5.2 Synthèse d'un régulateur auto-ajustable à variance minimale généralisée

Le système à commander est décrit par le modèle ARMAX (Auto-Regressif à moyenne Mobile et entrée eXogène), à paramètres constants. Ce modèle est à la base de la synthèse de la plupart des lois de commande linéaire et de techniques d'identification (Figure III.6). Il est donné par:

$$A(q^{-1})y(k) = q^{-d} B(q^{-1})u(k) + C(q^{-1})\xi(k) \tag{III.1}$$

d représente le retard pur du système. C'est un nombre entier de périodes d'échantillonnage, et $d \geq 1$. $y(k)$ et $u(k)$ sont respectivement, l'entrée et la sortie du système. $\xi(k)$ est une séquence non corrélée provenant du bruit de mesure, de variance σ .

Les polynômes A , B et C sont donnés par:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_m q^{-m} \quad , \quad b_0 \neq 0$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_l q^{-l}$$

La définition de d impose: $b_0 \neq 0$. Les racines du polynôme C sont supposées résider à l'intérieur du cercle unité (stabilité du système en boucle fermée).

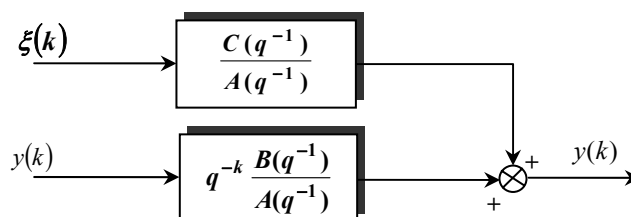


Figure III.6: Structure du modèle ARMAX.

Aucune restriction n'est imposée aux polynômes A et B . Cependant, le retard d et des bornes supérieurs de n , m et l doivent être connus a priori.

La structure du régulateur dépend du choix de la fonction coût à minimiser. Pour la technique de variance minimale généralisée, le critère choisit par Clarke et Gawthrop consiste à minimiser la variance d'une sortie auxiliaire [CLA79].

$$J = E[\Phi^2(k + d /_k)] \tag{III.2}$$

Φ est la sortie auxiliaire (Figure III.7) de la forme:

$$\Phi(k + d) = P(q^{-1})y(k + d) + Q(q^{-1})u(k) - R(q^{-1})W(k) \tag{III.3}$$

$W(k)$ désigne le signal de référence à l'instant k . Les Polynômes P , Q et R sont des pondérations qui peuvent être choisies selon les caractéristiques de la réponse désirée (action intégrale, minimisation des effets de la perturbation,...).

Ce critère introduit dans sa structure suffisamment de degrés de liberté pour pouvoir assurer la stabilité d'une large variété de systèmes.

Dans l'équation (III.3), le second et le troisième terme du second membre sont connus à l'instant k . Il reste alors à prédire le premier terme, qui est de la forme:

$$\Phi_y(k) = P(q^{-1})y(k) \tag{III.4}$$

La prédiction de la sortie Φ_y est donnée par [CLA79], [GAW82]:

$$C(q^{-1})\Phi_y^*(k + d /_k) = F(q^{-1})y(k) + G(q^{-1})u(k) \tag{III.5}$$

où F et G sont des polynômes vérifiant l'identité:

$$\frac{C(q^{-1})P(q^{-1})}{A(q^{-1})} = E(q^{-1}) + q^{-d} \frac{F(q^{-1})}{A(q^{-1})} \tag{III.6}$$

avec: $G = EB$. E et F sont respectivement, d'ordre $(d-1)$ et $(n-1)$. A partir des équations (III.3) et (III.5), la sortie auxiliaire prédite sera:

$$\Phi^*(k + d) = \Phi_y^*(k + d) + Q(q^{-1})u(k) - R(q^{-1})W(k) \tag{III.7}$$

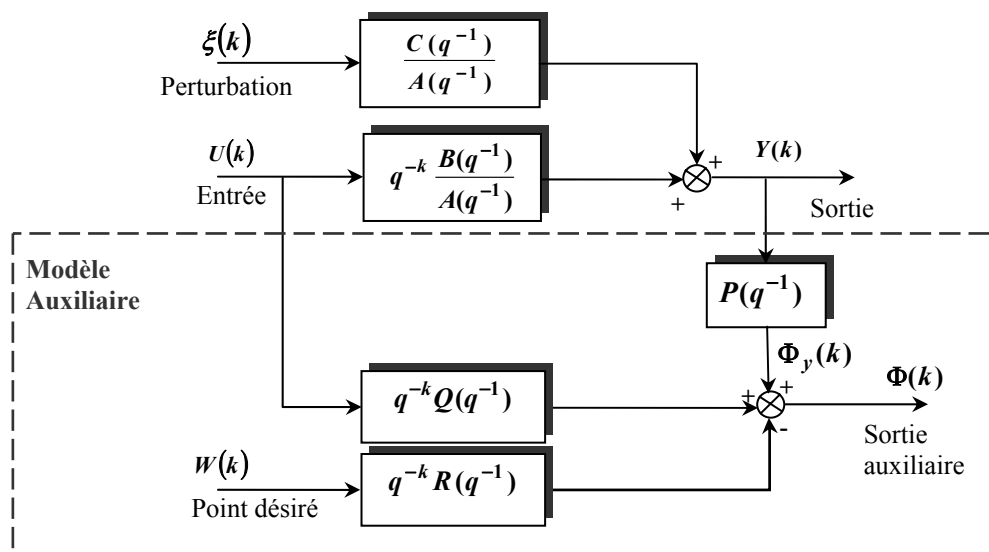


Figure III.7 : Structure du modèle du procédé avec le modèle auxiliaire.

La loi de commande optimale obtenue par minimisation du critère choisi, en annulant la prédiction à l'instant k de la sortie auxiliaire définie par (III.7), est donnée comme suit:

$$U_0(k) = \frac{R(q^{-1})W(k) - \Phi_y^*(k + d/k)}{Q(q^{-1})} \quad (\text{III.8})$$

En utilisant les équations (III.5) et (III.7), elle s'exprime par:

$$U_0(k) = \frac{C(q^{-1})R(q^{-1})W(k) - F(q^{-1})y(k)}{G(q^{-1}) + Q(q^{-1})C(q^{-1})} \quad (\text{III.9})$$

C'est une loi de commande incluant la trajectoire de référence. Cette commande optimale doit, en outre, être combinée avec un algorithme d'estimation paramétrique pour produire un régulateur auto-ajustable.

III.5.3 Algorithme auto-ajustable implicite

Le schéma implicite de Åstrom et Wittenmark, et Clarke & Gawthrop est basé sur l'idée d'éviter de séparer entre l'étape d'identification et celle de la commande, car les paramètres du contrôleur sont incorporés dans la procédure d'identification [ÅST73], [CLA75], [CLA79].

Pour calculer la commande optimale (III.8), il est nécessaire d'avoir la sortie $\Phi_y^*(k + d/k)$ définie par (III.5). Comme les polynômes F , G et C sont inconnus, leurs coefficients sont alors à estimer. Pour cela, un algorithme d'identification récursif basé sur les moindres carrés étendus (MCE), est utilisé. Le "*principe d'équivalence certaine*", permet d'utiliser les estimés à la place des vrais paramètres, dans la loi de commande [ÅST89].

Le prédicteur peut s'écrire comme suit:

$$\Phi_y^*(k + d/k) = F(q^{-1})y(k) + G(q^{-1})u(k) - \sum_{i=1}^n c_i \Phi_y^*(k + d - i/k_i) \quad (\text{III.10})$$

Définissons les vecteurs X et θ par:

$$\theta^T(k) = [g_0, \dots; f_0, \dots; -c_1, \dots]$$

$$X^T(k) = [u(k), \dots; y(k), \dots; \Phi_y^*(k + d - 1/k_{-1}), \dots]$$

Ces derniers désignent les observations et les paramètres correspondants. L'équation (III.5) devient alors:

$$\Phi_y^*(k + d/k) = X^T(k)\theta(k) \quad (\text{III.11})$$

Finalement, prédire la sortie $\Phi_y^*(k + d/k)$ revient à estimer le vecteur paramètres θ composé des coefficients des polynômes F , G et C .

L'algorithme auto-ajustable direct est donné par:

étape 1: Estimation du vecteur paramètres θ par les moindres carrés étendus.

étape 2: Calcul du prédicteur $\Phi_y^*(k + d/k)$ puis du signal de commande donné par l'équation (III.8) en utilisant les estimés obtenus dans l'étape 1.

Revenir à l'étape 1.

III.6 COMMANDE ADAPTATIVE A ERREUR DE PREDICTION MINIMALE

III.6.1 Principe de la commande

L'idée de cette commande est basée sur le fait que les performances désirées du système peuvent être obtenues en atteignant une erreur de poursuite de référence $(W(k)-y(k))$ nulle et réaliser la stabilité en boucle fermée. L'utilisation de l'approche directe pour cette commande, nécessite la mise en forme prédictive du modèle du procédé.

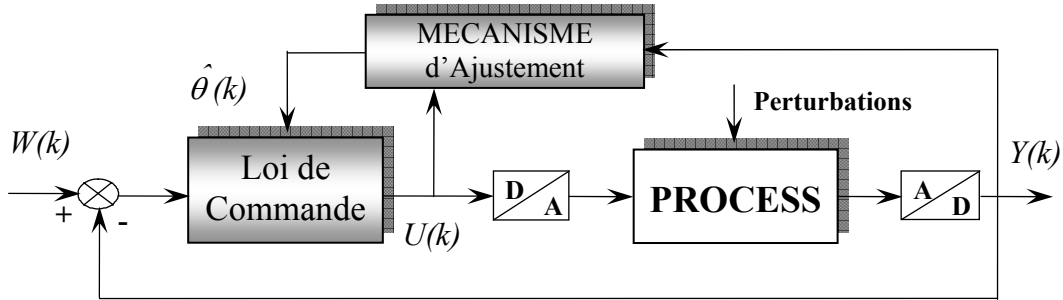


Figure III.8: Structure de commande adaptative à EPM.

III.6.2 Synthèse de la loi de commande

Le modèle retenu pour cette commande est le modèle DARMA (Deterministic ARMA) donné par l'équation polynomiale suivante:

$$A(q^{-1})y(k) = q^{-d} B(q^{-1})u(k) \quad (III.12)$$

Les polynômes A et B sont définis de la même manière que dans l'équation (III.1).

Les hypothèses suivantes doivent être vérifiées:

- ☑ le retard d du système et des bornes supérieures de n et m sont connus a priori,
- ☑ $B(q^{-1})$ doit avoir ses zéros à l'intérieur du cercle unité.

Pour avoir la forme prédictive du modèle, nous utilisons l'identité suivante [GOO84]:

$$I = F(q^{-1}) A(q^{-1}) + q^{-d} G(q^{-1}) \quad (III.13)$$

avec:

$$F(q^{-1}) = 1 + f_1 q^{-1} + \dots + f_{d-1} q^{-d+1}$$

$$G(q^{-1}) = g_0 + g_1 q^{-1} + \dots + g_{n-1} q^{-n+1}$$

le prédicteur à d pas est alors obtenu. Il est de la forme:

$$y(k+d) = \alpha(q^{-1}) y(k) + \beta(q^{-1}) u(k) \quad (III.14)$$

où les polynômes $\alpha(q^{-1})$ et $\beta(q^{-1})$ sont écrits comme suit:

$$\alpha(q^{-1}) = \alpha_0 + \alpha_1 q^{-1} + \dots + \alpha_{n-1} q^{-n+1}$$

$$\beta(q^{-1}) = \beta_0 + \beta_1 q^{-1} + \dots + \beta_{m+d-1} q^{-(m+d-1)} \quad ; \quad \beta_0 \neq 0$$

le modèle vectoriel devient donc:

$$y(k+d) = \phi^T(k) \theta(k) \quad (III.15)$$

θ et ϕ sont respectivement les vecteurs de paramètres et d'observations:

$$\theta(k) = [\alpha_0, \dots, \alpha_{n-1}, \beta_0, \dots, \beta_{m+d-1}]^T$$

$$\phi(k) = [y(k), \dots, y(k-n+1), u(k), \dots, u(k-m-d+1)]^T$$

L'erreur de poursuite est alors définie par

$$\begin{aligned} \varepsilon(k+d) &= y(k+d) - W(k+d) \\ &= \phi^T(k) \theta(k) - W(k+d) \end{aligned} \quad (\text{III.16})$$

où W représente la sortie désirée.

La loi de commande est choisie de façon à satisfaire:

$$\phi^T(k) \theta = W(k+d) \quad (\text{III.17})$$

et on aboutira à l'erreur $\varepsilon(k+d)$ nulle.

Comme les paramètres du vecteur θ sont inconnus et/ou variables dans le temps, un algorithme d'identification en ligne est alors utilisé pour les estimer à chaque pas. Nous utiliserons, par la suite, les estimés dans la loi de commande, à la place des vraies paramètres.

L'équation précédente devient alors:

$$\phi^T(k) \hat{\theta}(k) = W(k+d) \quad (\text{III.18})$$

où $\hat{\theta}(k)$ désigne les estimés des paramètres θ à l'instant k .

De l'équation (III.18), l'expression de la commande est obtenue comme suit:

$$\begin{aligned} u(k) = \frac{1}{\hat{\theta}_{n+1}(k)} [& -\hat{\theta}_1(k) y(k) \dots - \hat{\theta}_n(k) y(k-n+1) - \hat{\theta}_{n+2}(k) u(k-1) \\ & \dots - \hat{\theta}_{n+m+d}(k) u(k-m-d+1) + W(k+d)] \end{aligned} \quad (\text{III.19})$$

Cette loi ne s'applique qu'aux procédés ayant des zéros asymptotiquement stables. D'où l'idée d'étendre la méthode à d'autres systèmes, notamment, les systèmes à phase non minimale. Si nous gardons l'expression du prédicteur, donnée en Annexe "A" par (A.9), l'expression de la commande deviendra:

$$u(k) = \frac{W(k+d) - G(q^{-1})}{F(q^{-1})B(q^{-1})} \quad (\text{III.20})$$

Nous remarquons tout de suite que l'existence de $u(k)$, fini qui assure les objectifs définis, est liée à la position des racines de $B(q^{-1})$ par rapport au cercle unité. C'est pour cette raison que $B(q^{-1})$ doit avoir ces racines à l'intérieur du cercle unité. Autrement dit le système doit, obligatoirement, être à phase minimale.

Pour calculer $u(k)$, $W(t+d)$ doit être disponible à l'instant k . La sortie de référence W est engendrée par un modèle de référence. Sa fonction de transfert est déterminée à partir des performances souhaitées (temps de réponse, dépassement, temps d'établissement).

La fonction de transfert du modèle de référence est: $B_m(q^{-1})/A_m(q^{-1})$. La commande $u(k)$ doit réaliser cette fonction de transfert entre la consigne $r(k)$ et la sortie réelle du procédé y . Néanmoins, cela n'est pas possible car le retard d ne peut être éliminé ou compensé. La trajectoire à suivre est modifié par :

$$W(k) = q^{-d} \frac{B_m(q^{-1})}{A_m(q^{-1})} r(k) \quad (\text{III.21})$$

d'où la nécessité de connaître d .

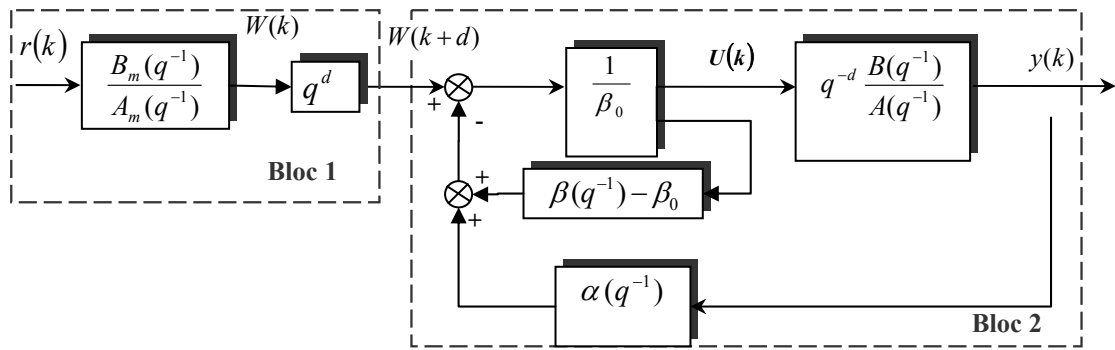


Figure III.9 : Structure de commande adaptative à EPM.

Dans la pratique, l'algorithme fonctionne si $\theta_{n+1}(k)$, paramètre qui correspond à l'estimation de β_0 , est différent de zéro. Pour éviter la division par zéro, il faut assurer que ce paramètre ne tend pas vers zéro.

On montre facilement que la fonction de transfert du bloc 1 est égale à q^{-d} , c'est à dire que

$$\begin{aligned} y(k) &= q^{-d} W(k+d) \\ &= W(k) \end{aligned} \quad (\text{III.22})$$

Avec cette commande, on voit bien que $y(k)$ suit parfaitement $W(k)$. La fonction de transfert du système en boucle fermée est:

$$y(k) = \frac{B_m(q^{-1})}{A_m(q^{-1})} r(k) \quad (\text{III.23})$$

Ce qui est l'objectif de la commande.

III.7 COMMANDE A ERREUR DE PREDICTION MINIMALE PONDEREE

III.7.1 Principe de la commande

La commande à erreur de prédiction minimale pondérée (EPMP) est une généralisation de la stratégie à erreur de prédiction minimale ordinaire (EPMO), qui nécessite certaines hypothèses pour garantir la stabilité de la boucle fermée [GOO84]. La commande à (EPMP) a pour objectif de faire un compromis entre l'énergie dépensée de la commande et l'obtention d'une erreur de poursuite de la référence ($W(k+d)-y(k+d)$) nulle, en réalisant la stabilité de la boucle fermée. En conséquence, cette technique permet:

- de réduire l'énergie de la commande
- de traiter une large gamme de systèmes.

Les hypothèses sont réduites à la connaissance a priori du retard d et des bornes supérieures de n et m .

III.7.2 Synthèse de la loi de commande

Comme l'approche de cette commande est directe, le modèle du système (III.12) est mis sous forme prédictive (III.14).

Les polynômes $\alpha(q^{-1})$ et $\beta(q^{-1})$ sont formulés par:

$$\begin{aligned}\alpha(q^{-1}) &= G(q^{-1}) \\ \beta(q^{-1}) &= F(q^{-1}) B(q^{-1})\end{aligned}\quad (\text{III.24})$$

tels que les polynômes $F(q^{-1})$ et $G(q^{-1})$ sont solutions de l'identité (III.13).

Pour garder la simplicité de l'algorithme, nous allons faire une extension du critère de la commande à (EPMO) en introduisant suffisamment de degrés de liberté dans la structure du critère afin d'assurer la stabilité d'une large variété de classes de systèmes.

Le critère à utiliser s'écrit:

$$J = \left\{ \frac{1}{2} \varepsilon^2(k+d) + \frac{\lambda}{2} \bar{u}^2(k) \right\} \quad (\text{III.25})$$

où l'erreur de poursuite est définie par (III.16) et $\bar{u}(k)$ est exprimé par la relation:

$$P(q^{-1}) \bar{u}(k) = R(q^{-1}) u(k) \quad (\text{III.26})$$

$P(q^{-1})$ et $R(q^{-1})$ sont des polynômes de pondération de la forme:

$$P(q^{-1}) = 1 + p_1 q^{-1} + \dots + p_l q^{-l}$$

$$R(q^{-1}) = 1 + r_1 q^{-1} + \dots + r_l q^{-l}$$

En substituant l'équation (III.16) dans (III.25), et en annulant la dérivée par rapport à $u(k)$ de l'expression résultante, nous aboutissons à:

$$\beta_0 \{ y(k+d) - W(k+d) \} + \lambda \bar{u}(k) = 0 \quad (\text{III.27})$$

En multipliant cette équation par $A(q^{-1})$ et en utilisant (III.12), nous obtenons l'expression de la loi de commande minimisant le critère J :

$$u(k) = \frac{P(q^{-1}) A(q^{-1}) W(k+d)}{P(q^{-1}) B(q^{-1}) + \frac{\lambda}{\beta_0} R(q^{-1}) A(q^{-1})} \quad (\text{III.28})$$

Si nous utilisons l'équation (III.14) au lieu de (III.12) la commande aura pour expression:

$$u(k) = \frac{P(q^{-1}) [W(k+d) - \alpha(q^{-1}) y(k)]}{P(q^{-1}) \beta(q^{-1}) + \frac{\lambda}{\beta_0} R(q^{-1})} \quad (\text{III.29})$$

La boucle fermée est stable si le dénominateur de l'équation (III.28) est stable. Ce polynôme peut être stabilisé par un choix approprié de λ , $P(q^{-1})$ et $R(q^{-1})$ pour tout système. En pratique, un filtrage et une action intégrale pour garantir une erreur de poursuite nulle, sont introduits [CLA79], [GOO84]. Ils sont choisis généralement, de la forme $R(q^{-1})/P(q^{-1}) = (1 - q^{-1})/(1 - \alpha q^{-1})$ où $0 < \alpha < 2$. Cette loi peut être appliquée aux procédés ayant des zéros instables.

Nous remarquons bien que pour calculer la séquence $u(k)$ dans (III.28), il faut avoir les polynômes $\alpha(q^{-1})$ et $\beta(q^{-1})$. Comme leurs paramètres sont supposés inconnus et/ou variables dans le temps, un algorithme d'identification récursif est alors utilisé pour les estimer. En se basant sur "*le principe d'équivalence certaine*", les estimés seront utilisés, par la loi de commande, comme s'ils étaient les vrais paramètres [ÅST89].

L'algorithme des moindres carrés récursifs développé en Annexe A est donné par les équations suivantes:

$$\begin{aligned}
\hat{\theta}(k) &= \hat{\theta}(k-1) + K(k)[y(k) - \phi^T(k-1)\hat{\theta}(k-1)] \\
K(k) &= \frac{P(k-1)\phi(k-1)}{\mu + \phi^T(k-1)P(k-1)\phi(k-1)} \\
P(k) &= \frac{1}{\mu} \left[P(k-1) - \frac{P(k-1)\phi(k-1)\phi^T(k-1)P(k-1)}{\mu + \phi^T(k-1)P(k-1)\phi(k-1)} \right]
\end{aligned} \tag{III.30}$$

$P(k)$ et $K(k)$ sont respectivement la matrice covariance et le gain d'adaptation.

Pour garantir une bonne estimation, cet algorithme est initialisé en choisissant la matrice covariance initiale de la forme: $P(0)=I/\varepsilon$, où ε est très faible si on ne dispose pas d'information, à priori, sur les paramètres, et proche de 1 si on dispose d'une estimation initiale.

Le facteur μ du gain d'adaptation est introduit pour maintenir une capacité d'adaptation de l'algorithme de façon à prendre en charge les variations paramétriques tout au long de l'estimation. Ce facteur est choisi souvent variable, et il est calculé pour assurer une trace constante de la matrice covariance [LAN93a] [LAN93b].

III.8 CONCLUSION

Dans ce chapitre, une approche de la commande adaptative, basée sur les régulateurs auto-ajustables, a été analysée. Deux stratégies de commande sont explorées : la commande à erreur de prédiction minimale pondérée (EPMP) et la commande à variance minimale généralisée. L'approche directe du régulateur auto-ajustable a été considérée. La première commande est une extension de la technique à erreur de prédiction minimale simple (EPM) qui nécessite des hypothèses pour garantir la stabilité de la boucle fermée. L'application de cette nouvelle stratégie couvre tous les types de systèmes grâce à l'introduction, dans la structure du critère à minimiser, un certain nombre de degrés de liberté pour assurer la stabilité de la boucle fermée. Il suffit pour cela de faire le bon choix des coefficients des polynômes de pondération.

La deuxième technique est introduite pour prendre en charge les bruits qui peuvent entacher les grandeurs mesurées. Ces lois de commande sont associées à un algorithme d'estimation paramétrique, lequel est basé sur les moindres carrés récurrents, pour estimer directement des paramètres de la loi de commande et lieu à un régulateur auto-ajustable. Les algorithmes présentés permettent d'illustrer, tant au niveau des performances que de la robustesse, l'apport de la commande adaptative pour des entraînements électriques avec la machine synchrone à aimants permanents qui fera l'objet du prochain chapitre.

CHAPITRE

-IV-

APPLICATION DE LA COMMANDE ADAPTATIVE A LA MSAP

IV APPLICATION DE LA COMMANDE ADAPTATIVE A LA MACHINE SYNCHRONNE A AIMANTS PERMANENTS

IV.1 INTRODUCTION

Les stratégies de contrôle de l'ensemble convertisseur-machine électrique deviennent de plus en plus complexes. Cette complexité suit l'évolution des besoins industriels dont les cahiers de charge sont actuellement plus contraignants, et nécessitent l'utilisation d'outils modernes de l'automatique. Les régulateurs de type PI ou PID à coefficients constants, sont insuffisants pour répondre à ces exigences, il convient alors d'utiliser un contrôle adaptatif [KAD00].

La commande adaptative est une alternative intéressante permettant de résoudre le problème de la variation des paramètres internes de la machine. Plusieurs possibilités de commande adaptative ont été envisagées.

Dans ce présent chapitre, le régulateur adaptatif comporte un algorithme d'identification de type moindres carrés récurrents (MCR) permettant d'estimer les paramètres de la fonction de transfert échantillonnée de la machine synchrone à aimants permanents, combiné avec les stratégies de commande développées au chapitre précédant à savoir : la commande à erreur de prédiction minimale ordinaire et pondérée et la commande à variance minimale généralisée. Les correcteurs obtenus permettent la régulation et la poursuite de la trajectoire de la vitesse de référence en présence des perturbations paramétriques.

IV.2 APPLICATION DE LA COMMANDE ADAPTATIVE A LA MSAP

Nous appliquons, dans cette section, les stratégies du contrôle adaptatif analysées à la machine synchrone subissant les différentes perturbations. Le régulateur auto-adaptatif est surtout destiné à prendre en charge les variations paramétrique dues aux changement dans les conditions de fonctionnement de la machine ou à des perturbations indésirables. Parmi les différentes dispositions probables la variation du couple de charge que subit l'axe de rotation, de la poursuite des variations de la consigne et éventuellement les variations des paramètres physiques de la machine.

La structure générale de la commande adaptative appliquée à la MSAP est montrée par le schéma global de la figure IV.1. Nous distinguons la boucle de régulation et la boucle d'adaptation.

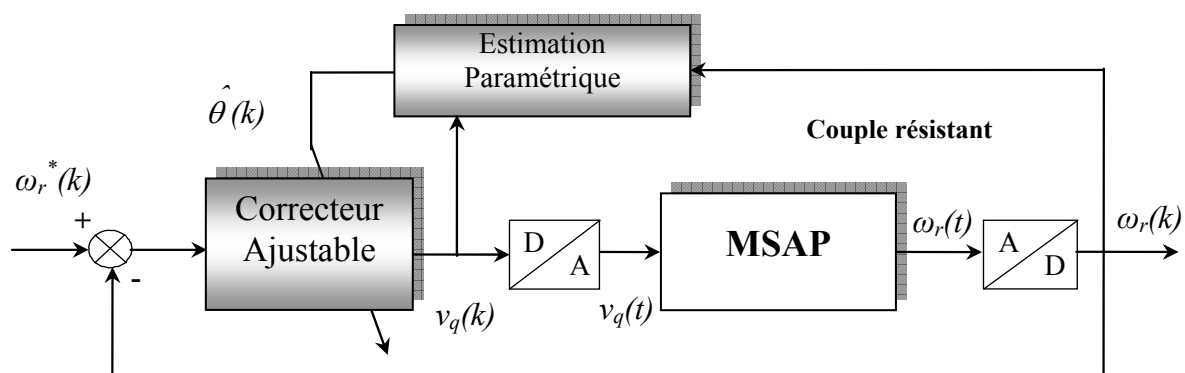


Figure IV.1: Schéma bloc de la commande adaptative à EPM de la MSAP

IV.3 RESULTATS DE SIMULATION ET DISCUSSION

Les paramètres utilisés pour simuler le comportement de la machine M1 sont donnés en Annexe D. Le comportement de la commande vis-à-vis de la variation de la consigne de vitesse, de la charge et de la variation des paramètres de la machine sera examiné pour les différentes stratégies de commande présentées.

IV.3.1 Commande à EPM et à EPMP de la MSAP

Nous appliquons, dans cette section, les stratégies du contrôle adaptatif à savoir la commande à erreur de prédiction minimale avec et sans pondération à la machine synchrone à aimants permanents. Le régulateur adaptatif est destiné à tenir compte des variations du couple de charge, de la poursuite de la consigne et éventuellement des paramètres physiques.

Notons que les valeurs du retard d et des degrés n, m des polynômes $A(q^{-1}), B(q^{-1})$ sont connus. Donc : $d=1, n=2, m=1$

Le comportement du système a été simulé en utilisant les paramètres de la machine M1 donnés par le Tableau D.1 de l'Annexe D. Nous présentons ici les résultats obtenus en utilisant la commande adaptative à EPMP directement. La figure IV.2 montre une comparaison entre la commande à EPM et à EPMP lors d'un démarrage de la machine. Nous remarquons une amélioration sur la qualité de réglage par rapport à la commande à EPM traduite par un temps d'établissement réduit sans dépassement et un rejet des oscillations du couple.

L'évolution des variables de la machine, lors du démarrage avec une vitesse de référence de 300 rad/s, est présentée dans la figure IV.3. Le test comporte aussi l'application d'un échelon de couple résistant entre l'instant $k=1s$ et $k=2s$. Nous constatons que l'effet de la charge est complètement pris en charge avec la commande. Nous avons varié le paramètre λ de la pondération de l'entrée et l'agitation du couple a été considérablement réduite (Figure IV.4).

L'efficacité de la commande en présence d'une perturbation paramétrique est testée pour une vitesse de consigne de 300 rad/s ainsi qu'une perturbation de charge dans la figure IV.5. Cette même variation est associée à un test de poursuite de variation de trajectoire de référence en figure IV.6. Dans le premier cas, nous avons effectué une augmentation de l'inertie de la machine et dans le second nous avons introduit une variation de la résistance statorique. Pour ces deux tests nous avons remarqué que ces variations influent sur le temps de réponse ce qui est normale mais pas sur l'action de la loi de commande qui assure un établissement sans dépassement, une régulation et une poursuite de la variation de consigne.

L'ensemble de ces résultats, montre les performances et la robustesse de cette loi de commande. En effet, il apparaît bien que la commande maintient la qualité de l'asservissement et du réglage aussi bien lors des variations paramétriques que celles de la charge.

Afin de mettre en évidence l'intérêt de la commande à EPMP, nous allons effectuer des tests de robustesse en explorant d'une façon très élaborée les différents paramètres de la loi de commande et de l'algorithme d'estimation.

La structure du modèle de la machine choisie est ($n=2, m=1$). Les paramètres des polynômes de pondération sont choisis comme suit: $\lambda = 0.03$ et $\alpha = -0.85$ pour le premier test et $\lambda = 1.03$ pour le reste.

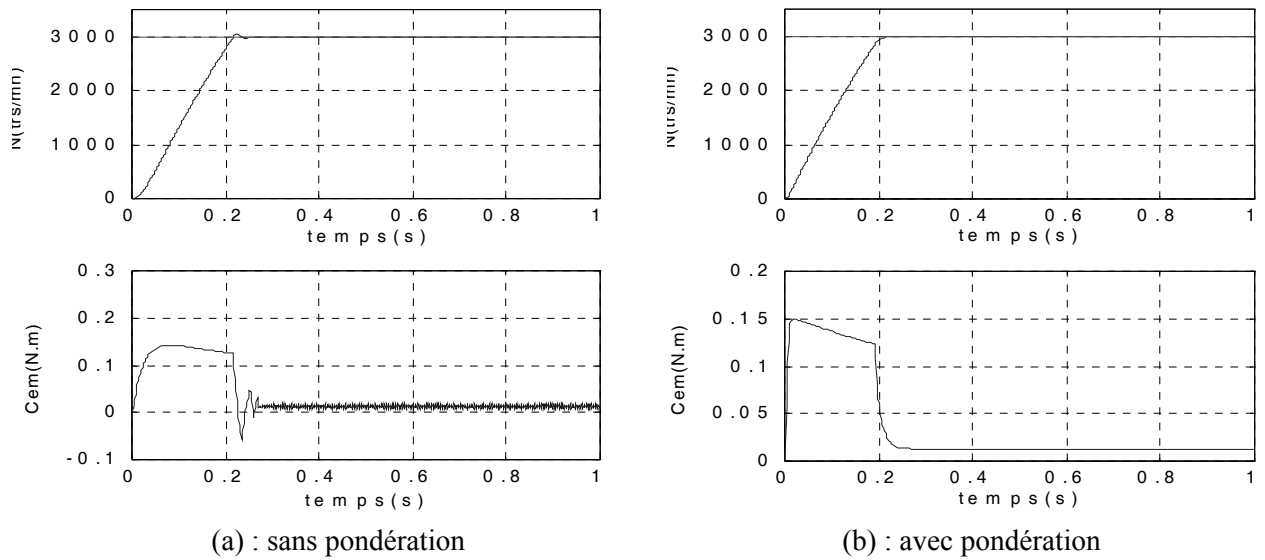


Figure IV.2: Démarrage de la machine avec et sans pondération

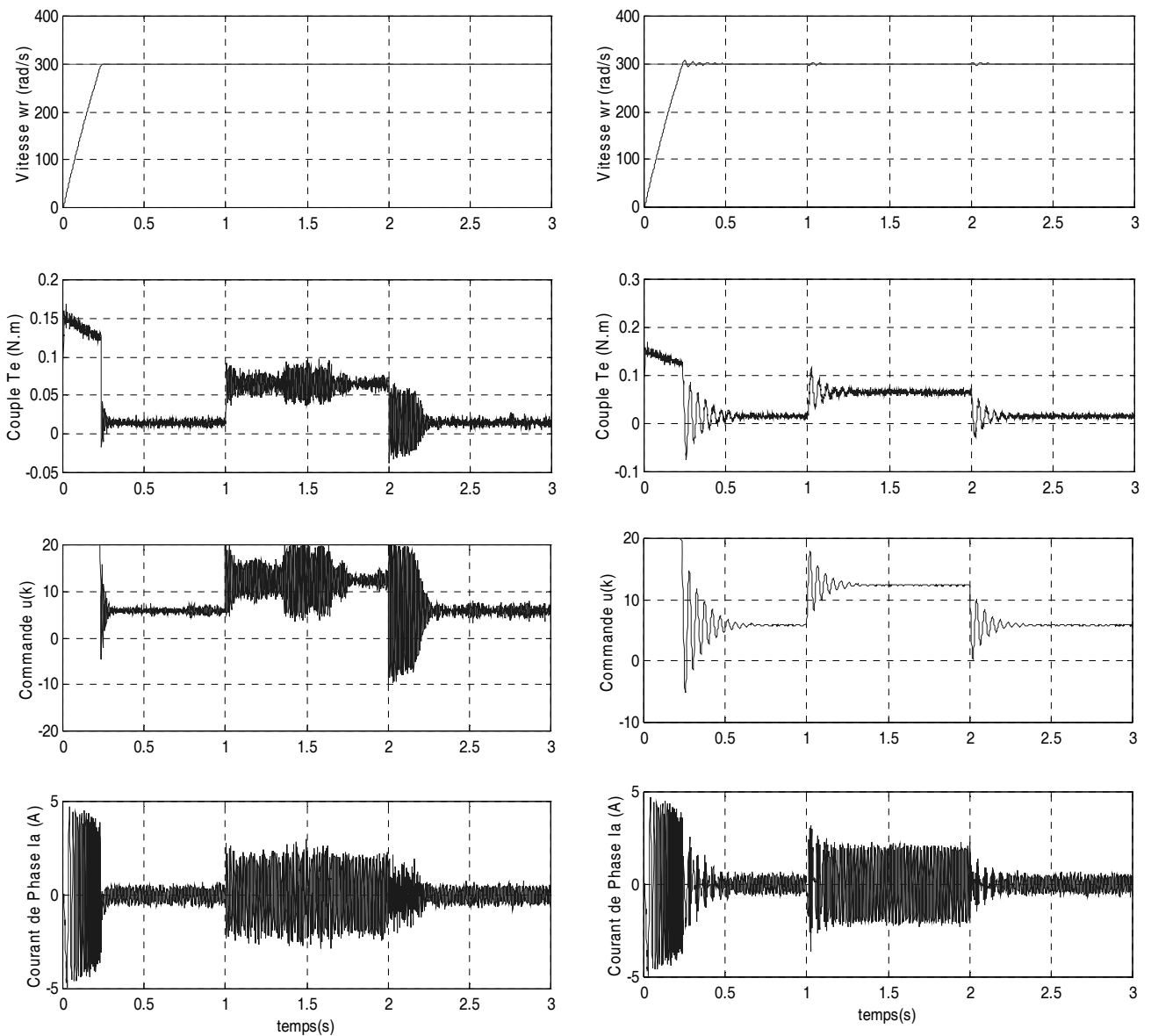


Figure VI.3: Perturbation de charge ($\lambda=0.03$).

Figure VI.4: Perturbation de charge ($\lambda=0.8$).

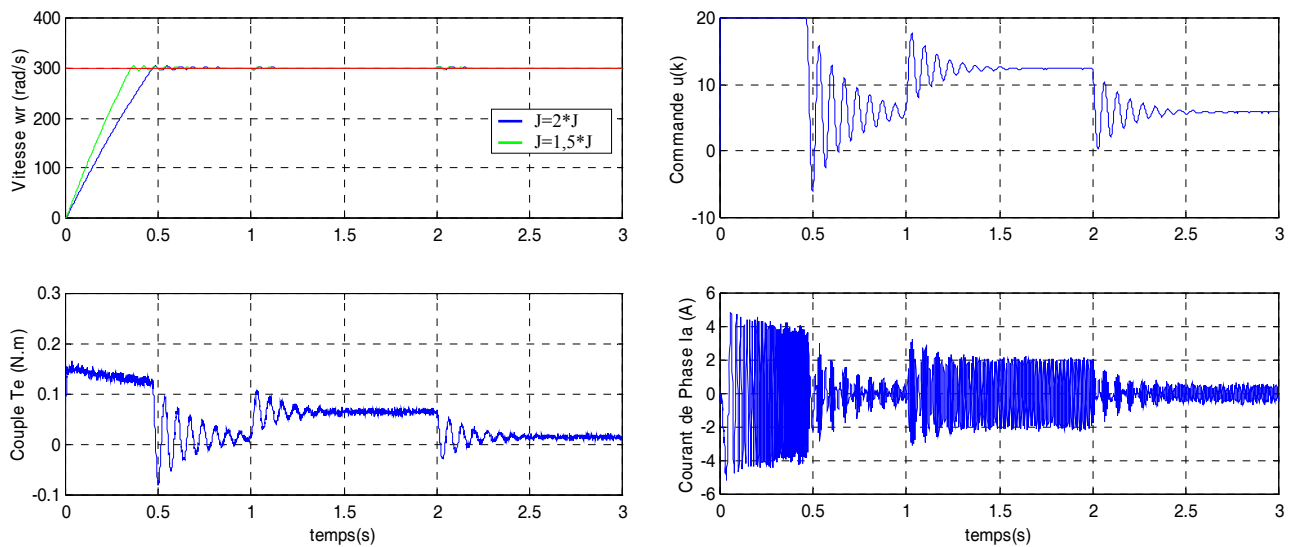


Figure IV.5: Test de Variation de J .

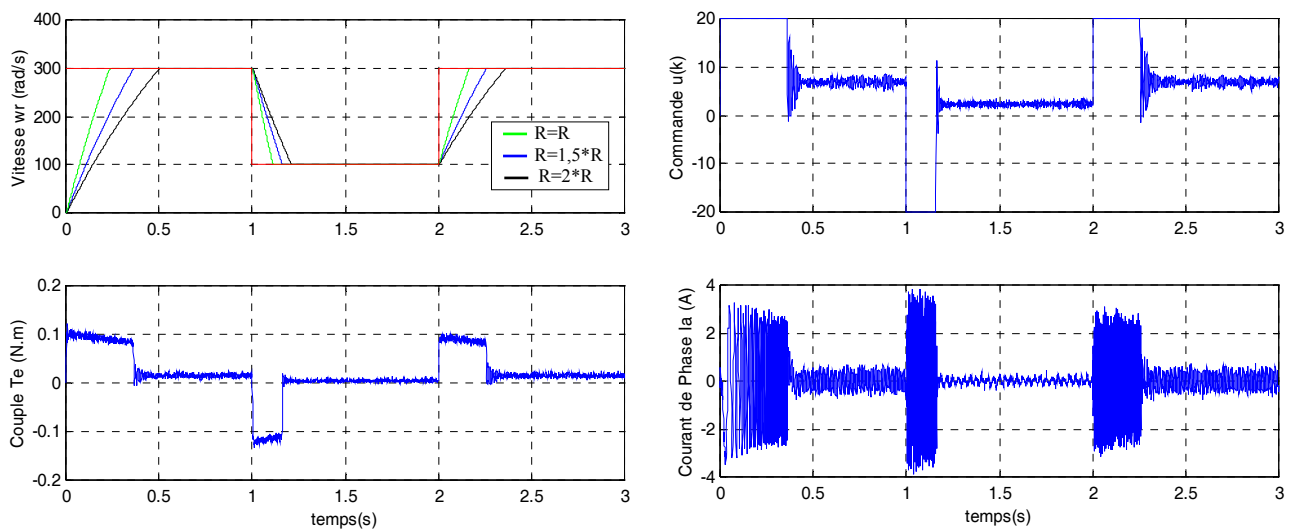


Figure IV.6: Test de poursuite de trajectoire avec variation de R .

La réponse à un changement de consigne et à une perturbation de charge des Figure IV.7, et IV.8 respectivement, montre que le contrôleur adaptatif est capable de poursuite de trajectoire de référence et de rejeter la perturbation de charge. Il est à noter, que dans le premier cas ($\lambda = 0.03$), le régulateur est très sollicité, alors que pour $\lambda = 1.03$ l'action du régulateur est plus douce et moins agitée.

La robustesse par rapport aux variations paramétriques de la MSAP, est explorée dans les figures IV.9, pour une variation de 50 % de R_s appliquée à $t = 0.5s$, la figure IV.10 pour une variation de 300 % de J_m . ces résultats montrent une prise en charge du rejet de la perturbation du couple résistant avec une petite oscillation et un établissement de la vitesse plus long qui est dû à l'augmentation de la constante de temps de la MSAP.

Les Figures IV.11 et IV.12 montrent la réponse de la vitesse dans le cas de la présence des perturbations paramétriques de R_s et J_m associées à la perturbation extérieure de charge. Ces tests sont effectués pour les faibles et les grandes vitesses. Nous pouvons constater la supériorité du contrôleur adaptatif élaboré sur le PI.

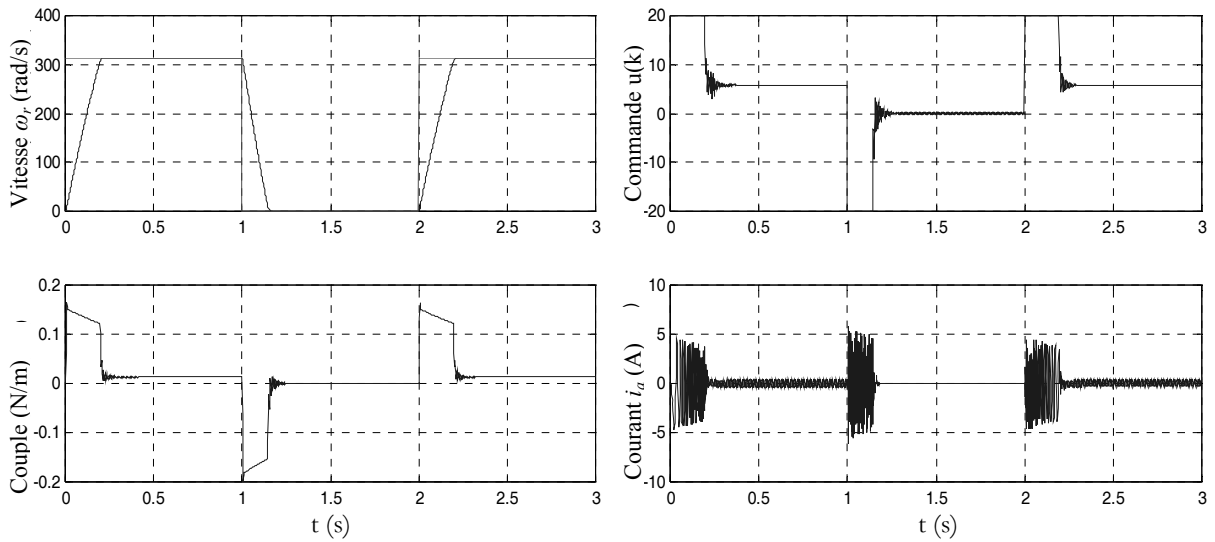


Figure IV.7: Poursuite de trajectoire de vitesse de référence

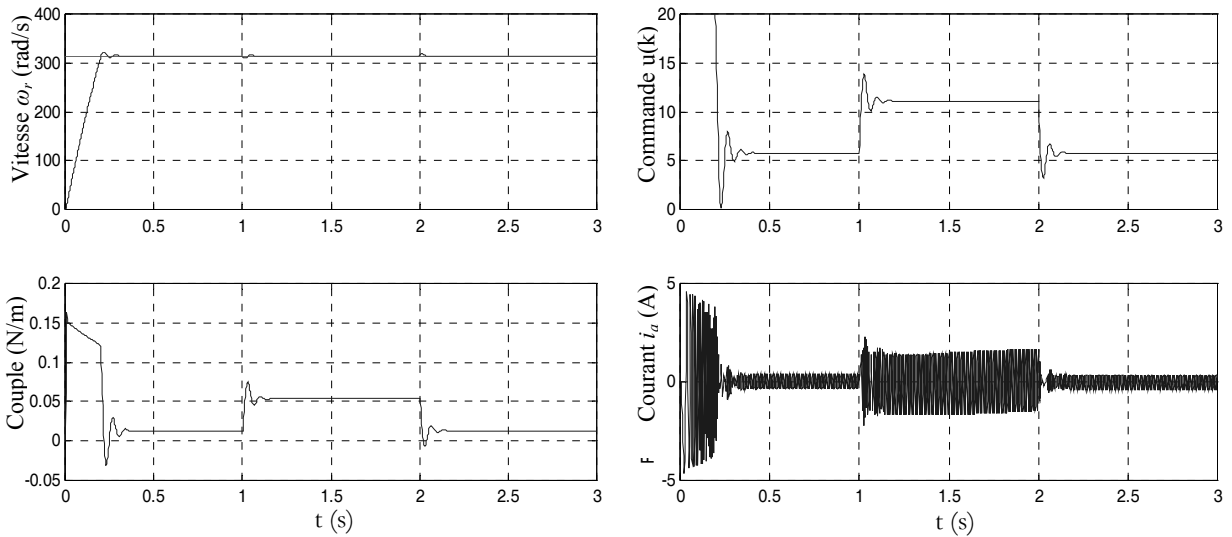


Figure IV.8: Application d'un échelon de couple résistant entre 1s et 2s ($T_r=0.05$ Nm)

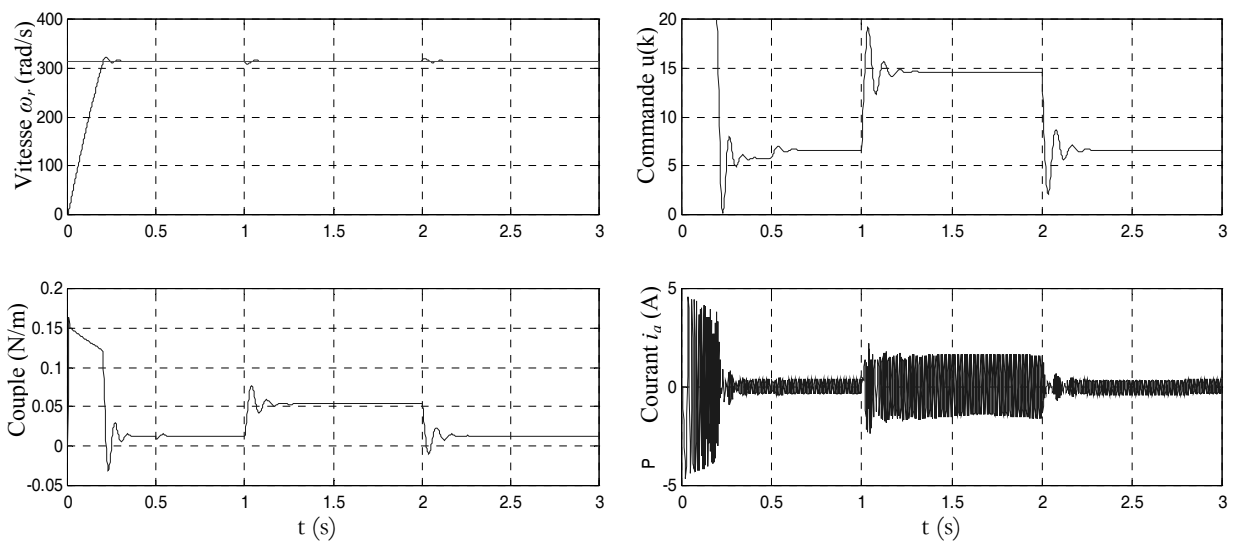


Figure IV.9: rejet de perturbation de couple avec une erreur +50% of résistance statorique R_s

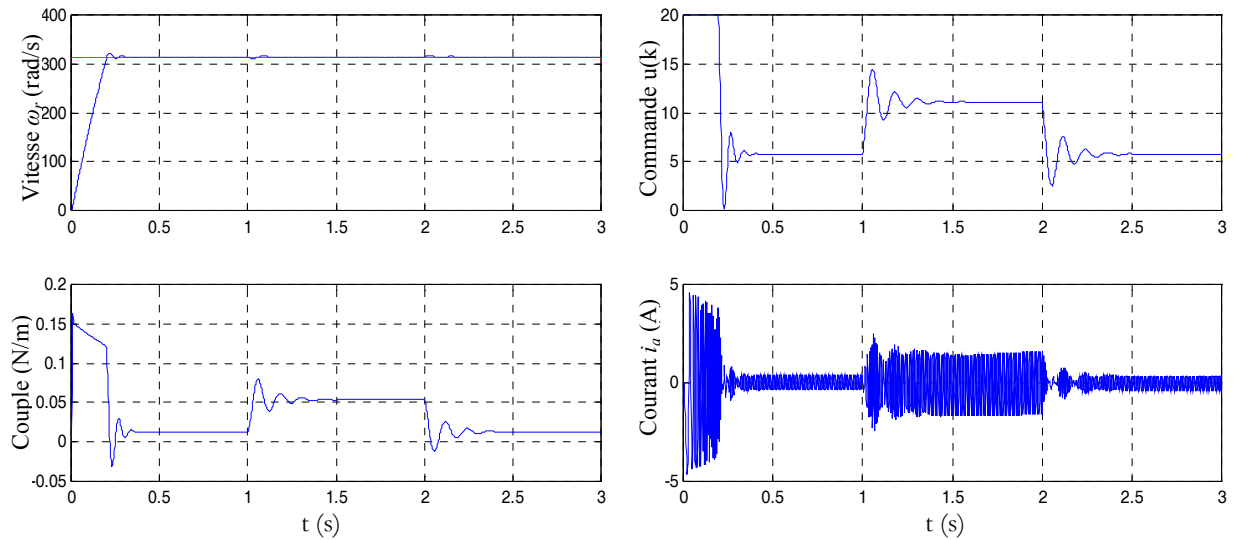


Figure IV.10: Application du couple résistant avec une variation de +300% de J_m

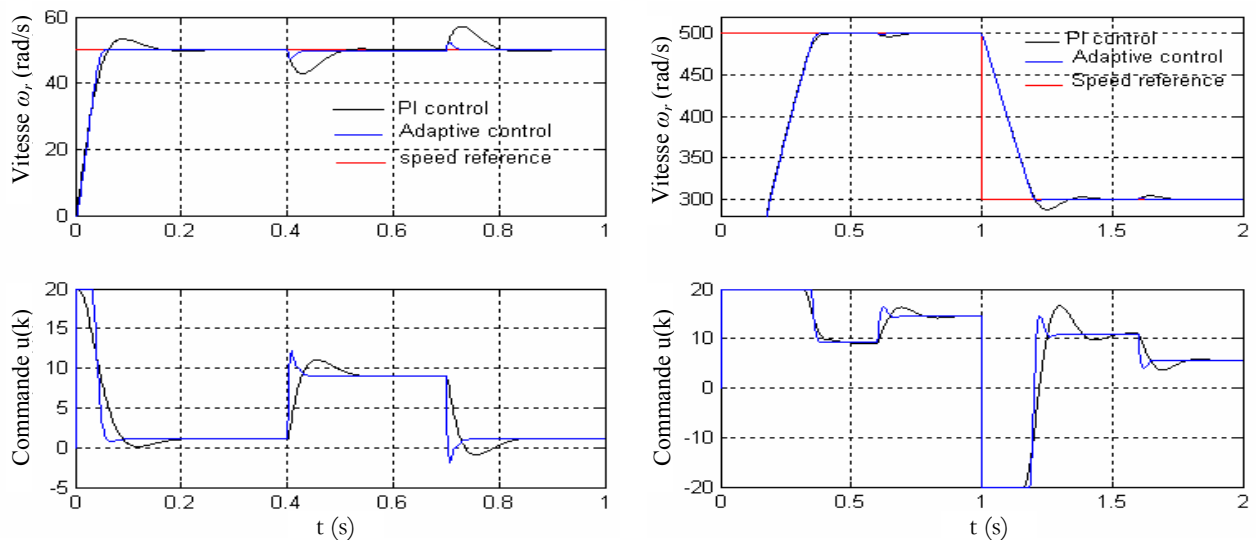


Figure IV.11: Robustesse de la commande pour les faibles vitesses (50 rad/s) (+50% de R_s)

Figure IV.12: Robustesse de la commande pour les faibles vitesses (500 rad/s) (+300% de J_m) at $t=0.4s$

IV.3.2 Application de la commande à VMG

Les tests effectués ont été simulés en choisissant une structure du modèle avec ($n=2$, $m=1$ et $l=1$). Les polynômes de pondération choisis sont: $R=1$, $P=1$ et $Q=l(1-q^{-l})$, avec $\lambda=0.03$ pour le premier test et 0.2 pour la suite. Un bruit de variance $\sigma=0.2$ et d'une moyenne nulle, est associé à la mesure de vitesse de rotation.

Les tests des figures IV.13 et IV.14 concernent la variation de la consigne de vitesse et montrent l'effet de la pondération sur les grandeurs de la machine. Le régulateur assure bien la poursuite de la consigne et minimise l'effet du bruit sur les différentes grandeurs. Il est évident que le paramètre λ a réduit d'une façon considérable les oscillations des grandeurs de la machine. La robustesse de la commande vis-à-vis d'une perturbation par échelon de couple résistant, est testée en Figure IV.15. La réponse montre bien l'action de la commande pour compenser l'effet de la perturbation et du bruit.

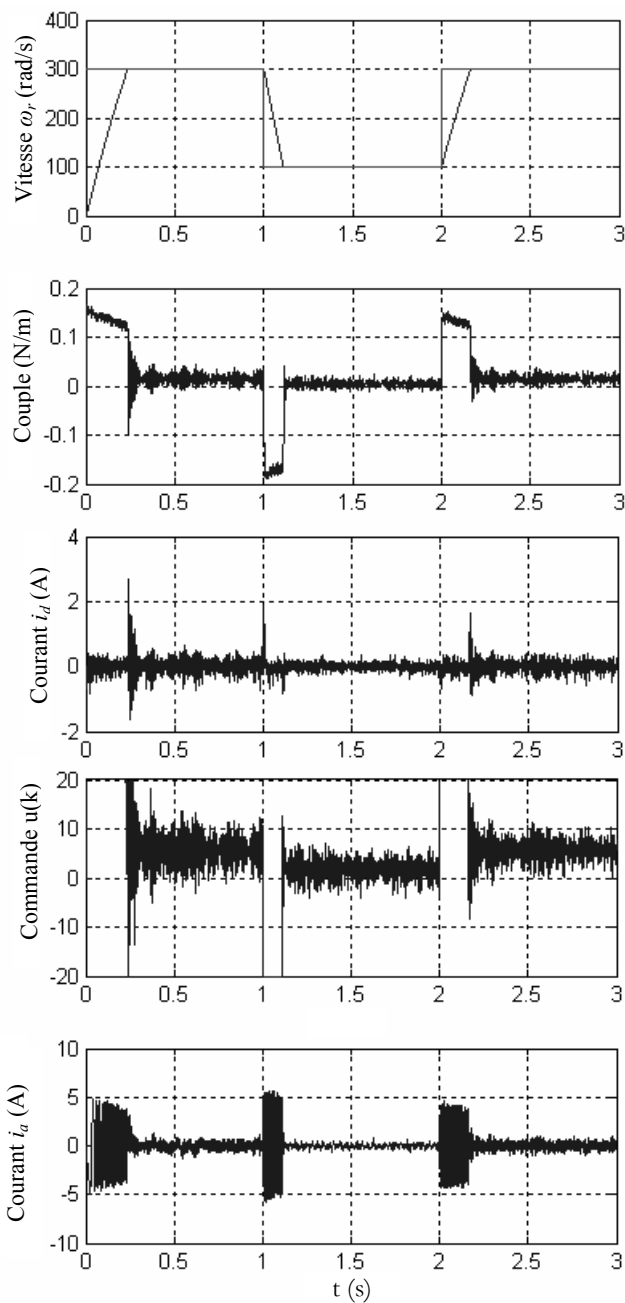


Figure IV.13: Test de poursuite ($\lambda=0,03$)

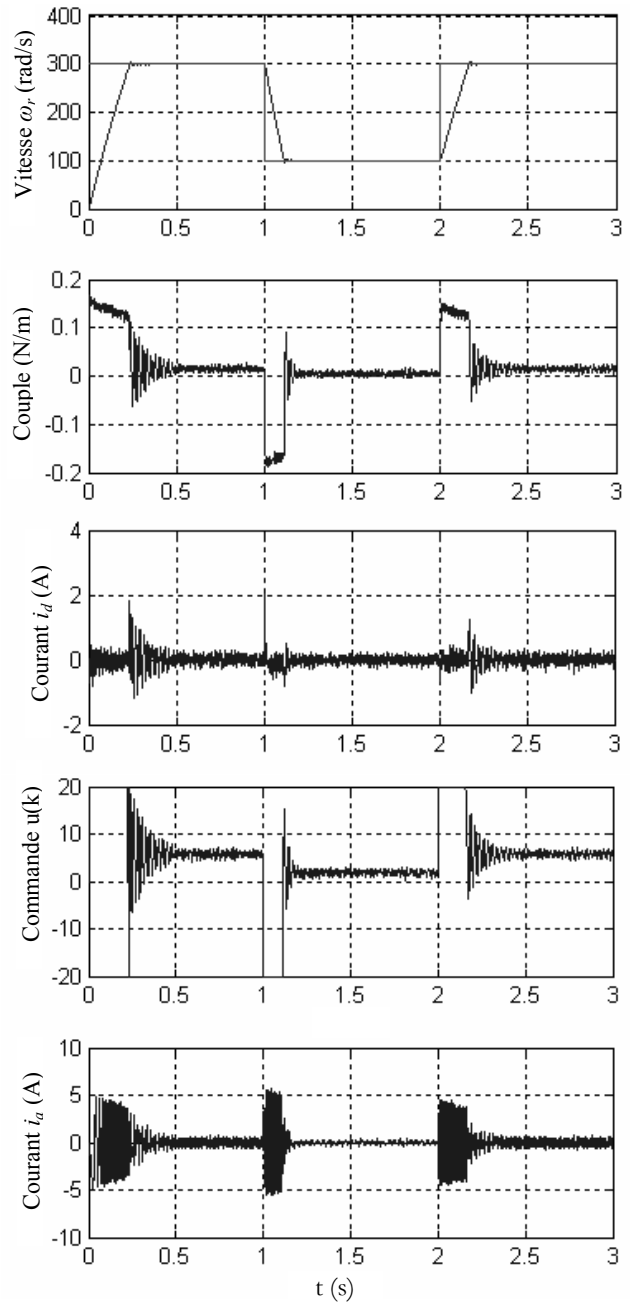


Figure IV.14: Test de poursuite ($\lambda=0,2$)

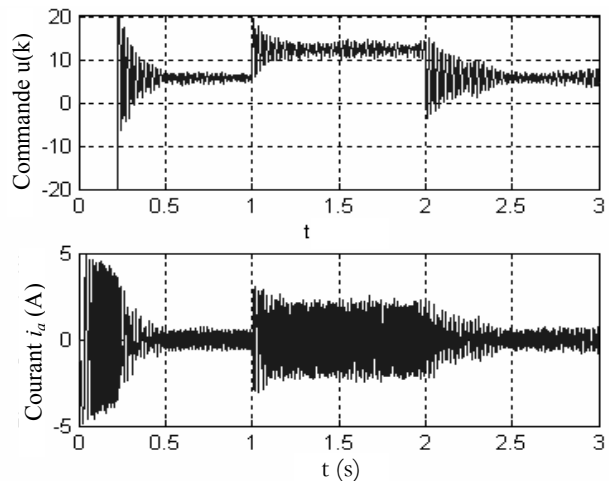
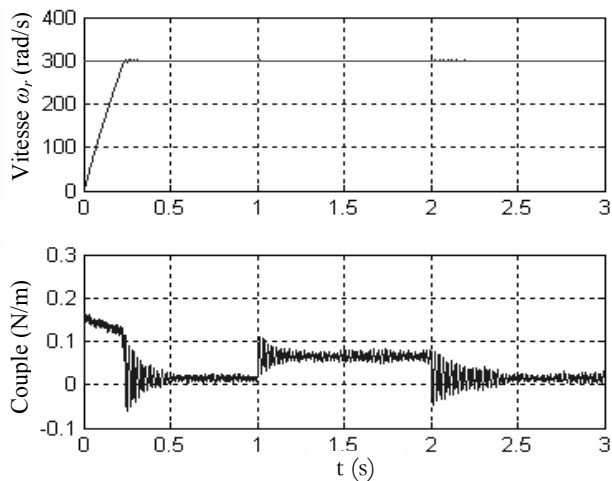


Figure IV.15: Application d'un couple de charge

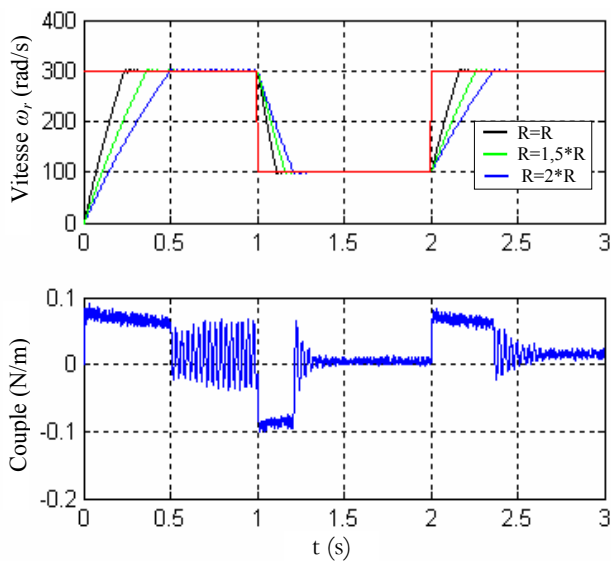


Figure IV.16: Variation de la résistance R en poursuite

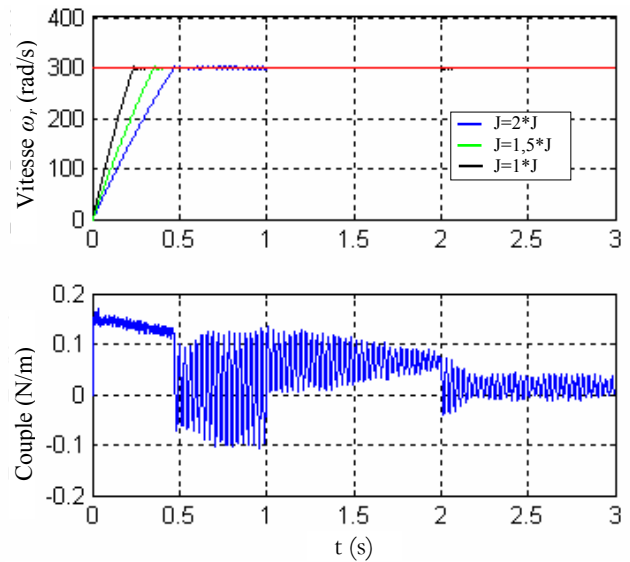


Figure IV.17: Variation de l'inertie J en régulation

L'influence d'une variation paramétrique en présence d'une perturbation de charge, appliquée à est montrée dans les figures IV.16 et IV.17. La variation de la résistance a été associée à un test de poursuite et celle de l'inertie à celui de la régulation. Ces résultats montrent que le régulateur est peu sensible aux variations du couple de charge et de la consigne même en présence du bruit de mesure. Il s'adapte bien lorsque les paramètres de la machine sont mal connus. La commande montre, donc, une bonne efficacité à maintenir les performances requises.

IV.4 CONCLUSION

Ce chapitre décrit l'application des deux stratégies de commande adaptative directe, à la boucle de vitesse d'une machine synchrone à aimants permanents. L'algorithme d'estimation utilisé est celui des Moindres Carrés Récursifs (MCR). L'utilisation de cet algorithme nous donne une convergence rapide. Différents régimes transitoires ont été simulés pour apprécier l'apport de ces commandes: démarrages à vide et en charge, applications d'échelons de couple résistant, variations paramétriques.

Les résultats obtenus montrent l'efficacité de la commande à erreur de prédiction minimale ordinaire à répondre aux différents types de variation paramétriques (la résistance statorique, l'inductance et le moment d'inertie) et de la charge, malgré le risque d'instabilité. C'est ce qui nous a conduit à introduire des polynômes de pondération dans la structure du critère. Les résultats obtenus sont nettement meilleurs. La commande à EPMP conduit mieux la sortie du système car la vitesse de rotation suit parfaitement la référence. Elle montre éventuellement une bonne robustesse à rejeter les perturbations ainsi qu'une amélioration du temps de réponse.

Ces résultats ont permis d'illustrer, tant au niveau des performances que de la robustesse, l'efficacité de la commande adaptative dans les entraînements électriques avec machine synchrone à aimants permanents.

CHAPITRE

-V-

LES RESEAUX DE NEURONES
ARTIFICIELS

V.1 INTRODUCTION

Les réseaux de neurones artificiels (RNA) sont des systèmes de traitement de l'information dont la structure s'inspire de celle du système nerveux. Ils sont destinés à effectuer des tâches auxquelles les approximateurs traditionnels semblent moins bien adaptés. Ainsi, les applications des réseaux de neurones artificiels à la reconnaissance de formes, à la modélisation, à la commande et à la classification ont pris une place importante au sein des réalisations industrielles. Dans le domaine du traitement du signal et de l'automatique, les années 90 ont été marquées par des avancées significatives tant du point de vue fondamental que du point de vue des applications industrielles [PER03].

L'utilisation de réseaux de neurones artificiels pour la réalisation de systèmes de commande a connu un essor important au cours des dernières années [CIR02]. Dans ce chapitre, nous allons introduire les réseaux de neurones en passant par l'historique de l'évolution de ces systèmes. Nous allons présenter aussi les possibilités offertes par ces techniques, ainsi que les modalités de leur utilisation, dans le domaine de la commande.

V.2 HISTORIQUE SUR LES RNA

L'origine de l'inspiration des réseaux de neurones artificiels remonte à 1890 où W. James, célèbre psychologue américain, introduit le concept de mémoire associative. Il propose ce qui deviendra une loi de fonctionnement pour l'apprentissage des réseaux de neurones, connue plus tard sous le nom de loi de Hebb. Quelques années plus tard, en 1943, J. Mc Culloch et W. Pitts donnent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes [MCC43]. En 1949, D. Hebb présente dans son ouvrage « The Organization of Behavior » une règle d'apprentissage. Cette règle ouvre la voie à l'apprentissage fondé sur les régularités statistiques des activations (apprentissage non-supervisé). De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb [DAU00].

C'est dans les années soixante, que les premiers modèles du perceptron ont fait leur apparition par F. Rosenblatt. C'est dans ce cadre que se développent les algorithmes d'apprentissage supervisé, fondés sur le terme d'erreur entre sortie produite et sortie souhaitée, qui est à l'origine de la règle de rétropropagation (apprentissage supervisé). C'est alors qu'en 1960, le modèle Adaline (Adaptative Linear Element) est développé par B. Widrow et Hoff. Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons Multi Couches (PMC). M. Minsky et S. Papert publient ensuite en 1969 un ouvrage qui met en évidence les limitations théoriques du Perceptron [HAR98]. La recherche dans ce domaine va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

Il faut attendre les années 80 pour voir se développer de nouveaux modèles. En 1982, John Joseph Hopfield introduit un modèle de réseau de neurones complètement récurrent. Les cartes auto-organisatrices de Kohonen mettent en place une règle d'apprentissage non-supervisée fondée sur la compétition. Enfin, en 1986, D. Rumelhart, G. Hinton et R. Williams publient,

l'algorithme de rétropropagation du gradient de l'erreur pour le perceptron multi-couches, ouvrant la voie à de nombreuses applications. A partir de ce moment, la recherche sur les réseaux de neurones connaît un essor fulgurant et les applications commerciales de ce succès académique suivent au cours des années 90 [BAG99].

Aujourd'hui, il existe plusieurs applications des réseaux de neurones dans pratiquement tous les domaines : financier, pharmaceutique, aéronautique, économique ; etc.

V.3 LE NEURONE BIOLOGIQUE

L'élément de base du système nerveux est le neurone. Le cerveau humain contient environ 10^{11} neurones, avec 10^3 à 10^4 dendrites par neurone. Le neurone est une cellule composée du soma (corps cellulaire) qui contient le noyau, les dendrites autour du soma et l'axone qui est un prolongement de grande taille au bout duquel se trouve une arborisation qui entre en contact avec les dendrites des autres neurones via les synapses (Figure V.1). Les dendrites sont si nombreuses que l'on parle de chevelure ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma (corps du neurone). Elle est traitée alors par le corps cellulaire. Si le potentiel d'action dépasse un certain seuil, le corps cellulaire répond par un stimulus. Le signal transmis par le neurone chemine ensuite le long de l'axone (unique) pour être transmis aux autres neurones [REN95]. La communication entre l'axone du neurone afférent et les dendrites du neurone efférent n'est pas directe et s'effectue par des synapses. Une synapse reliant deux neurones peut être de nature excitatrice ou inhibitrice. Dans le premier cas, le neurone émetteur aura tendance à activer le neurone récepteur, et dans l'autre cas le neurone émetteur aura tendance à inhiber l'activité du neurone récepteur. Chaque synapse est en outre caractérisée par l'efficacité avec laquelle elle assure la connexion [PER03].

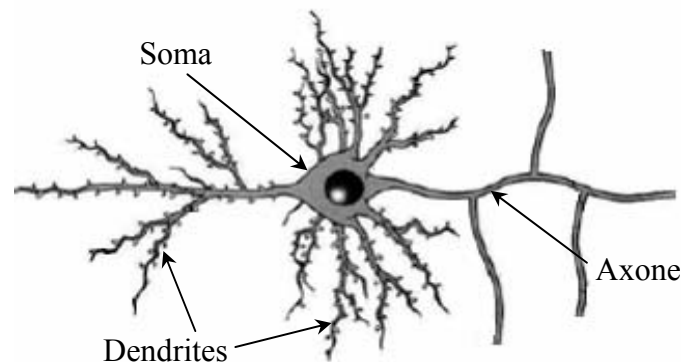


Figure V.1: Neurone biologique

V.4 LE NEURONE FORMEL

C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau, que les réseaux de neurones artificiels se sont développés. La structure d'un neurone artificiel est montrée dans la figure V.2. Chaque neurone artificiel est une unité fonctionnelle. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associé un poids ω , représentatif de la force de la connexion. Chaque unité fonctionnelle est dotée d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones aval [DAV90].

Le neurone formel est une fonction algébrique non linéaire et bornée, dont la valeur dépend des paramètres poids ω . Les variables de cette fonction sont les entrées du neurone, et la valeur de la fonction est la sortie. Un neurone est donc un opérateur mathématique, dont la valeur numérique est calculée par des opérations mathématiques.

V.4.1 Modèle mathématique non-linéaire

Par analogie avec le neurone biologique, le comportement du neurone artificiel est modélisé par deux phases :

- Un opérateur de sommation, qui élabore le potentiel a . Cet opérateur effectue la somme pondérée des entrées. On soustrait parfois à cette somme la valeur du seuil d'activation θ_i .
- un opérateur non linéaire qui calcule la valeur de l'activation. Cet opérateur s'appelle la fonction de transfert.

La nouvelle version du modèle non-linéaire de McCulloch & Pitts est représentée par la figure V.2.

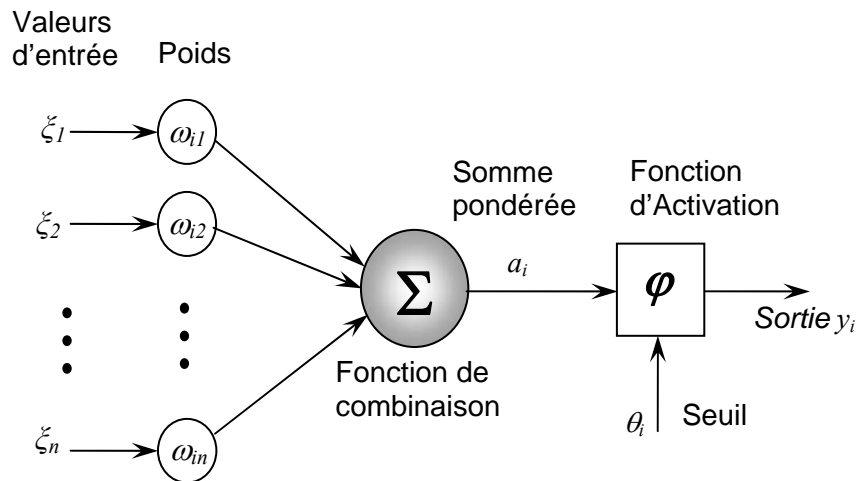


Figure V.2: Modèle Non-Linéaire du Neurone formel

ξ_j : entrée du neurone i (sortie du neurone amont j).

ω_{ij} : la valeur du poids synaptique de la connexion dirigée du neurone j vers le neurone i ,

☐ un poids positif indique un effet excitateur du neurone émetteur (j) vers le neurone récepteur (i).

☐ un poids négatif représente un effet inhibiteur.

θ_i : est le seuil du neurone i qui représente la limite à partir de laquelle le neurone s'activera.

La fonction combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques. Autrement dit, elle calcule la somme pondérée des entrées selon l'expression suivante:

$$a_i = \sum_{j=1,n} \omega_{ij} \cdot \xi_j \tag{V.1}$$

A partir de cette valeur, une fonction de transfert (ou fonction d'activation, ou encore fonction de seuillage) φ calcule la valeur de l'état du neurone. Cette valeur sera transmise aux neurones aval.

$$y_i = \varphi(a_i - \theta_i) \tag{V.2}$$

Les neurones les plus fréquemment utilisés sont ceux pour lesquels la fonction φ est une fonction non linéaire (généralement une tangente hyperbolique) d'une combinaison linéaire des entrées.

V.4.2 Modèle mathématique étendu

Le modèle peut être modifié en introduisant une valeur de biais θ_i à la place de la valeur de seuil: $\xi_0 = +1$ et $\omega_{i0} = \theta_i$.

L'équation mathématique qui décrit ce modèle est donnée par:

$$y_i = \varphi(a_i) \tag{V.3}$$

avec:

$$a_i = \sum_{j=0,n} \omega_{ij} \cdot \xi_j \tag{V.4}$$

Ce qui donne alors le modèle non-linéaire étendu de la figure V.3.

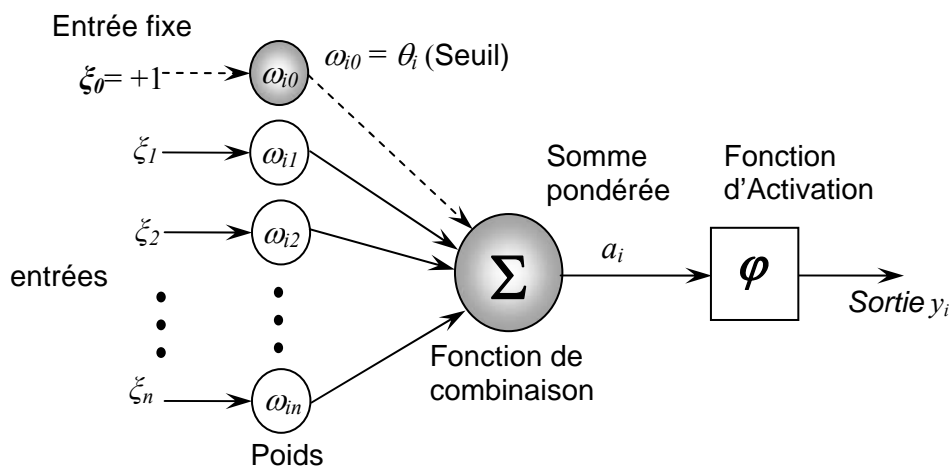


Figure V.3 : Modèle Etendu du Neurone formel

V.4.3 Les fonctions de transfert

La fonction de transfert ou d'activation définit la valeur de sortie d'un neurone en termes des niveaux d'activité de ses entrées. Ces fonctions présentent généralement trois intervalles :

1. en dessous du seuil, le neurone est non-actif (sa sortie vaut 0 ou -1) ;
2. aux alentours du seuil, une phase de transition ;
3. au-dessus du seuil, le neurone est actif (sa sortie vaut 1).

La fonction de transfert peut prendre différentes formes possibles tel que: fonction linéaire à seuil, fonction seuil, fonction gaussienne, etc (figure V.4).

☞ *Fonction à seuil* : Comme son nom l'indique, cette fonction applique un seuil sur son entrée. La sortie effective du neurone est calculée en comparant la valeur du potentiel à un seuil [HAR98]. Cette fonction délivre des valeurs binaires [+1 0] ou [+1 -1] (figure V.4).

$$y_i = \varphi(a_i) = \begin{cases} 1 & a_i > \theta_i \\ 0 & a_i \leq \theta_i \end{cases} \tag{V.5}$$

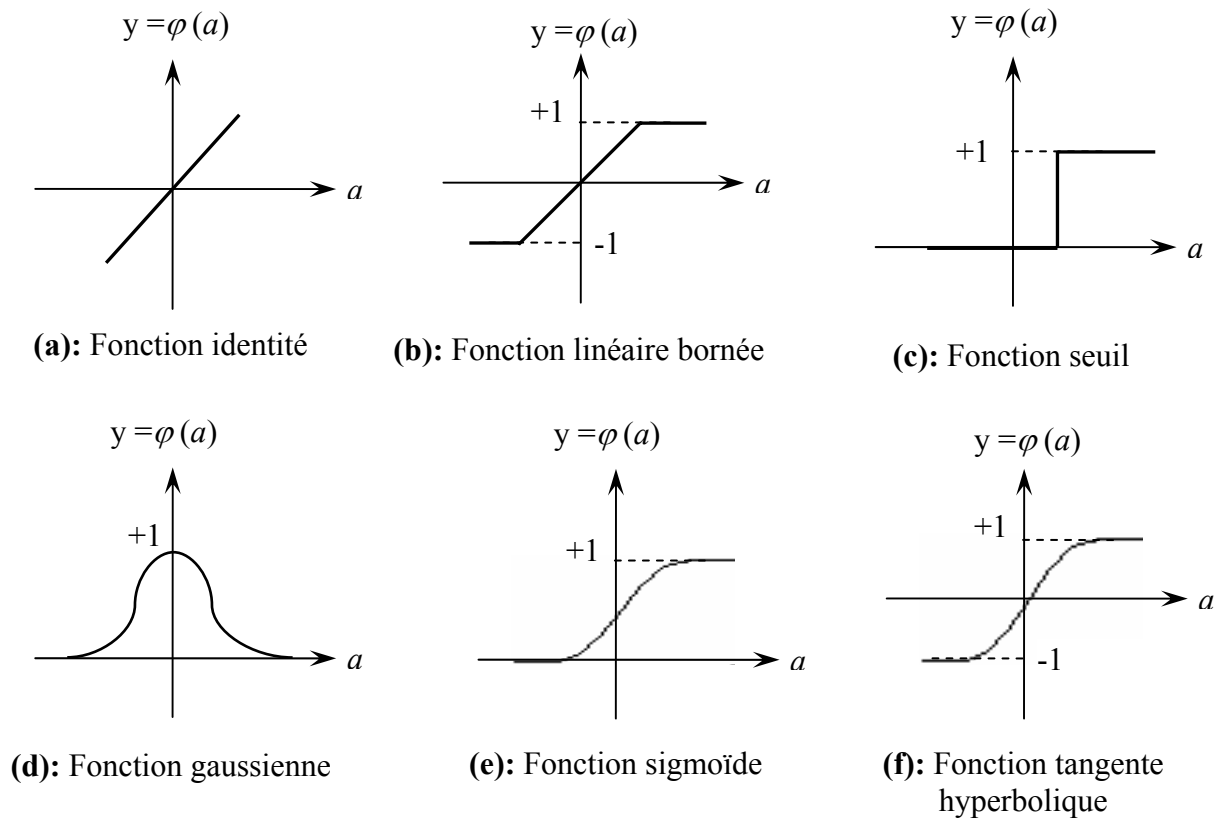


Figure V.4: Les formes usuelles de la fonction d'activation

☞ *Fonction linéaire par partie* : La fonction linéaire est une fonction simple. La sortie effective du neurone est directement liée à son entrée selon la relation $y = \varphi(a)$. Cette fonction peut aussi être bornée.

$$y_i = \varphi(a_i) = \begin{cases} +1 : \text{si } a_i > \theta_i \\ -1 : \text{si } a_i < \theta_i \\ \varphi(a_i) : \text{sinon} \end{cases} \quad (\text{V.6})$$

☞ *Fonction sigmoïde* : Cette fonction n'est pas polynomiale et est indéfiniment continûment dérivable. C'est une propriété simple qui permet d'accélérer le calcul de la dérivée, ce qui réduit le temps calcul nécessaire à l'apprentissage d'un réseau de neurones. Elle est définie par la relation mathématique suivante :

$$y = \frac{1}{1 + e^{-\alpha a}} \quad (\text{V.7})$$

☞ *Fonction gaussienne* : Cette fonction d'activation n'est pas monotone croissante, elle est d'un caractère local qui la rend très puissante en approximation de fonction et en classification.

V.5 LE PERCEPTRON

Le perceptron est généralement constitué d'une seule couche, de plusieurs neurones dont les fonctions d'activations sont toutes des fonctions de seuil. Ce type de réseau est utilisé seulement dans les problèmes linéairement séparables. On distingue aussi d'autres types de réseaux de neurones qui lui sont similaires, mais dont l'utilité est différente [DRE98]:

- L'adaline (Adaptative Linear Neuron Networks).
- Le madaline (Many Adaptative Linear Neuron Networks).

V.6 NEURONE A BASE RADIALE

Ces dernières années, plusieurs types de neurones et fonctions d'activation ont été introduits pour résoudre différentes sortes de problèmes pratiques. En particulier, les neurones à base radiale qui ont montré leur utilité dans les applications de l'identification et de la commande des systèmes [CIR02]. Un neurone à fonction de base radiale (Radial Basis Functions) est construit à partir d'une fonction du même nom. Un réseau de neurones à fonction de base radiale est un réseau à une seule couche cachée avec des fonctions d'activation à base radiale, le plus souvent des gaussiennes. Les entrées sont directement connectées aux neurones de la couche cachée. La couche de sortie est à un neurone avec une fonction d'activation identité.

V.7 LES RESEAUX DE NEURONES FORMELS OU RESEAU MULTICOUCHE

Un réseau de neurone (RN) est formé d'un ensemble de neurones interconnectés. Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Celle-ci peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité. Généralement, la structure des RN est constituée d'une couche d'entrée, des couches cachées et une couche de sortie.

Un réseau de neurone est donc composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de n_i neurones, prenant leurs entrées sur les n_{i-1} neurones de la couche précédente. À chaque synapse est associée un poids synaptique, de sorte que les n_{i-1} sorties précédentes sont multipliés par ces poids, puis additionnés par les neurones de niveau i . la fonction de sortie introduit une non linéarité. Il est très important de faire un choix judicieux d'une bonne fonction de sortie : un réseau de neurones dont les sorties seraient linéaires n'aurait aucun intérêt.

Deux grands types d'architectures de réseaux de neurones sont distingués: les réseaux de neurones non bouclés et les réseaux de neurones bouclés.

V.7.1 Les réseaux de neurones non bouclés

Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones élémentaires. Ce réseau est représenté par un ensemble de neurones connectés entre eux, de telle sorte que l'information se propage des entrées vers les sorties sans retour en arrière. Ce réseau est de type « feedforward ».

La Figure V.5 représente un réseau de neurones non bouclé qui a une structure très fréquemment utilisée. Il comprend des entrées, une couche de neurones cachés et des neurones de sortie. Les neurones de la couche cachée ne sont pas connectés entre eux. Cette structure est

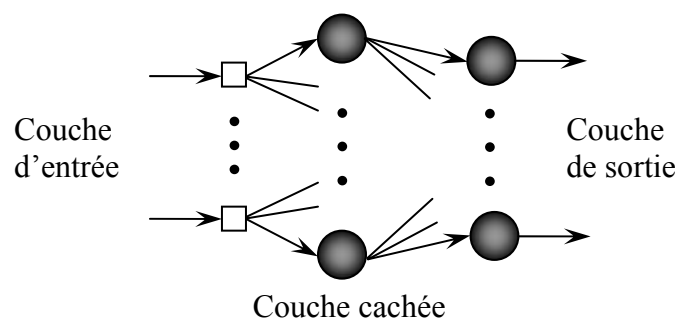


Figure V.5: Structure d'un réseau multicouche (feedforward)

appelée Perceptron multicouche (PMC). Ces réseaux sont des systèmes statiques. Si les entrées sont indépendantes du temps, les sorties le sont aussi. Ils sont utilisés principalement pour les approximations de fonction non linéaire, la classification ou la modélisation de processus statiques non linéaires.

Le Réseau multicouche à connexions locales est un cas de la structure multicouche, mais qui conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (figure V.6). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique [REN95].

V.7.2 Les réseaux de neurones bouclés

Contrairement aux réseaux de neurones non bouclés dont le graphe de connexions est acyclique, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties. Ces réseaux sont de type « feedback ». Pour qu'un tel système soit causal, il faut associer à toute boucle un retard. Un réseau de neurones bouclé est donc un système dynamique.

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation de systèmes dynamiques, de commande de processus. Parmi les réseaux bouclés, nous citons quelques configurations :

- **Réseau à connexions récurrentes** : Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales (figure V.7) [PER03].

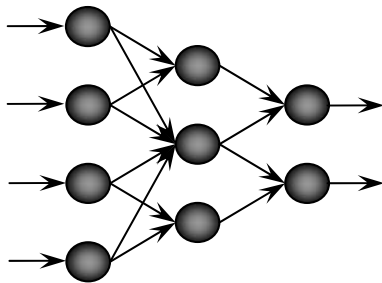


Figure V.6 : Réseau multicouche à connexions locales

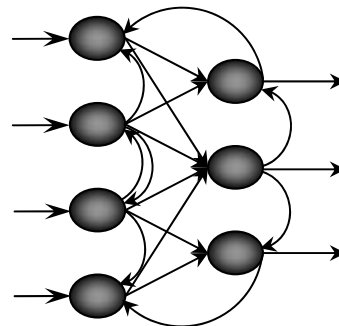


Figure V.7 : Réseau à connexions récurrentes (ou feedback)

- **Réseau à connexions complexe** : C'est la structure d'interconnexion la plus générale ou chaque neurone est connecté à tous les neurones du réseau (figure V.8).

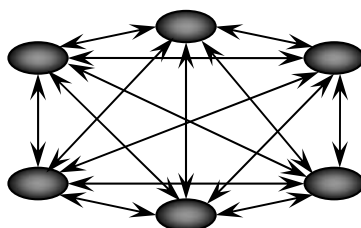


Figure V.8 : Réseau à connexion complexe

V.7.3 Le choix du nombre d'unités cachées

Le nombre d'unités cachées joue un rôle crucial dans le contrôle de la capacité du réseau de neurones. Si ce nombre est trop petit, alors le réseau possède trop peu de paramètres et ne peut capter toutes les dépendances qui servent à modéliser et prédire les valeurs du processus observé. Il faut comprendre que les réseaux de neurones peuvent modéliser n'importe quelle fonction si le nombre d'unités cachées est suffisant [MIL92].

La structure des réseaux généralement utilisés pour la commande est celle qui est mise sous la forme canonique. Elle donnée par la figure V.9.

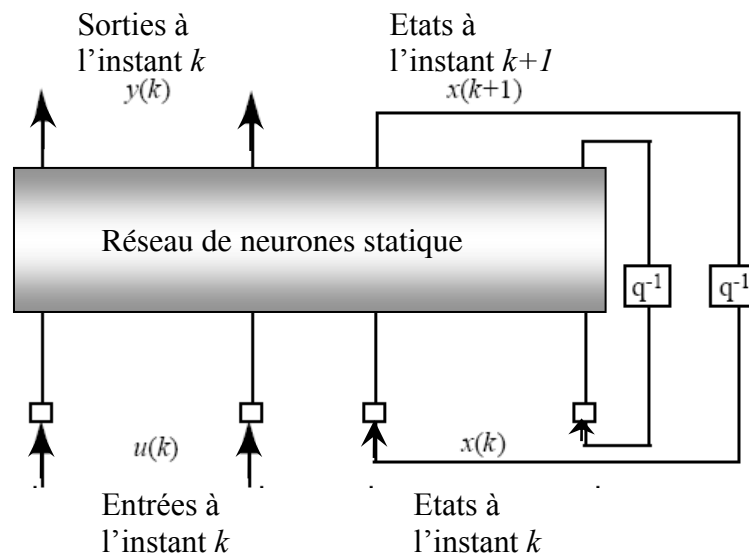


Figure V.9: Forme canonique d'un réseau de neurones bouclé

V.8 APPRENTISSAGE DES RESEAUX DE NEURONES

L'une des propriétés des plus importantes des réseaux de neurones artificiels est leur faculté d'apprentissage. C'est une phase du développement du réseau de neurones durant laquelle son comportement est modifié jusqu'à l'obtention du comportement désiré. Celui-ci peut s'effectuer par présentation répétée d'une série de « patrons » ou « modèles » et peut être supervisé ou non. Dans le cas des réseaux multicouches les approches les plus utilisées sont celles inspirées de la méthode de rétro-propagation du gradient mais d'autres méthodes peuvent également être mises en oeuvre.

L'algorithme d'apprentissage est souvent associé à la description du modèle des réseaux de neurones artificiels. Dans la majorité des algorithmes, les variables modifiées pendant l'apprentissage sont les poids des connexions. Il est souvent impossible de décider a priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de la phase d'apprentissage, les poids sont fixés : C'est alors la phase de validation.

Le nombre de cycles d'apprentissage peut être défini par l'utilisateur. Plus les cycles seront nombreux, plus l'erreur sur l'ensemble d'entraînement deviendra petite. L'erreur diminuera aussi sur l'ensemble de tests jusqu'à un certain nombre de cycles. Passé ce nombre, l'erreur sur l'ensemble de tests commencera à augmenter à nouveau. Ceci est dû au fait que plus l'apprentissage est long, plus le réseau apprend les spécificités propres à l'ensemble d'entraînement, et ainsi perd son pouvoir de généralisation.

Les algorithmes d'apprentissage sont divisés en deux grandes classes, selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage.

V.8.1 L'apprentissage non supervisé

Le calcul des coefficients dans ce cas se fait sur la base de la suite des vecteurs d'entrées seulement. Le réseau a donc la charge d'extraire des régularités de la suite des vecteurs d'entrée. L'apprentissage non supervisé a pour modèle de référence les cartes dites « autoorganisatrices » (*Self Organizing Maps*) de Kohonen (1982).

V.8.2 L'apprentissage supervisé

Ces algorithmes utilisent une base de données composée d'un ensemble de paires entrée-sortie. L'apprentissage dans ce cas, consiste à calculer les coefficients de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties "désirées", qui peuvent être la valeur de la fonction approchée ou de la sortie du processus à modéliser, ou encore la sortie souhaitée du processus à commander.

L'apprentissage des réseaux de neurones bouclés (pour réaliser des modèles dynamiques) est semblable à celui des réseaux non bouclés.

V.9 REGLES ET METHODES D'APPRENTISSAGE

Le principe de l'apprentissage consiste à soumettre le réseau à un vecteur entré dont la réponse souhaitée est connue, autant de fois qu'il lui est nécessaire à la modification des poids des connexions en minimisant une fonction coût (erreur quadratique), jusqu'à obtention de la bonne réponse. Il existe plusieurs règles de modification des poids dont les principales sont :

- ☞ La règle de Hebb
- ☞ La règle de Widrow-Hoff
- ☞ L'algorithme de rétropropagation du gradient de l'erreur

V.9.1 La règle de Hebb

La règle de Hebb est inspirée de la biologie. Le principe est de renforcer les connections entre deux neurones lorsque ceux-ci sont actifs simultanément. Elle peut être classée comme apprentissage non supervisé, ou supervisé car on sait calculer directement les poids correspondant à l'apprentissage d'un certains nombres d'exemples.

Cette règle d'apprentissage a été le point de départ des travaux de Rosenblatt qui a développé la première version d'un modèle neuronal très connu de nos jours, à savoir le perceptron. La modification des poids est inspirée de la règle de Hebb. Son principe est de rajouter, dans le cas où la sortie obtenue y_i du réseau est différente de la sortie désirée y_i^* , une quantité Δw_{ij} aux poids de chaque connexion. Dans le cas contraire, les connexions demeurent inchangées [DAU00]. Ce principe peut être exprimé par:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad (\text{V.8})$$

Δw_{ij} est la quantité ajoutée au poids w_{ij} , d'où :

$$\Delta w_{ij} = \eta (y_i^* - y_i) \zeta_j \quad (\text{V.9})$$

Le paramètre η est appelé taux d'apprentissage, il détermine la dynamique suivant la quelle les modifications vont avoir lieu et ζ_j représente l'entrée.

V.9.2 La règle de Widrow-Hoff

En 1960 Widrow et Hoff ont proposé, contrairement au modèle de Rosenblatt où les neurones ont des fonctions d'activation à seuils, un modèle de perceptron avec neurones linéaires (fonction d'activation linéaire). L'avantage d'utiliser des unités linéaires est que ces dernières permettent de calculer une fonction qui évalue l'erreur que commet le réseau. Cette erreur peut être calculée en fonction des réponses désirées et des réponses obtenues par le réseau. Elle est donc fonction des poids du réseau [PER03]:

$$E = \frac{1}{2} \sum_i (y_i^* - y_i)^2 = \frac{1}{2} \sum_i \left(y_i^* - \sum_j w_{ij} \zeta_j \right) \quad (\text{V.10})$$

L'objectif de l'apprentissage est de modifier les valeurs des poids du réseau de façon à minimiser cette erreur et à effectuer une descente du gradient. L'algorithme de descente du gradient consiste à changer chaque poids w_{ij} d'une quantité Δw_{ij} proportionnelle au gradient de l'erreur:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \sum_i (y_i^* - y_i) \zeta_j \quad (\text{V.11})$$

L'expression (V.11) est la même que celle trouvée lors de l'apprentissage par la règle de Hebb à base de la fonction à seuil (V.9).

V.9.3 Méthode de la rétropropagation du gradient (backpropagation)

Un des désavantages du perceptron est qu'il minimise une erreur en tout ou rien à cause de sa fonction d'activation à seuil. De ce fait, l'algorithme est très peu robuste. La règle d'apprentissage de Widrow-Hoff ne fonctionne plus en tout ou rien mais minimise une fonction d'erreur quadratique, donc plus robuste. Malheureusement, cette règle ne peut s'appliquer que sur des réseaux à une seule couche de poids adaptatifs. C'est donc en étendant la règle de Widrow-Hoff que l'algorithme d'apprentissage de rétropropagation du gradient de l'erreur a été développé [DAV90].

L'algorithme d'apprentissage de la rétropropagation du gradient d'erreur est généralisé ensuite par l'équipe de Rummelhart en 1986. Cet algorithme fournit une façon de modifier les poids des connexions de toutes les couches d'un Perceptron Multi Couches.

La figure V.10, représente un réseau à deux couches, dans lequel:

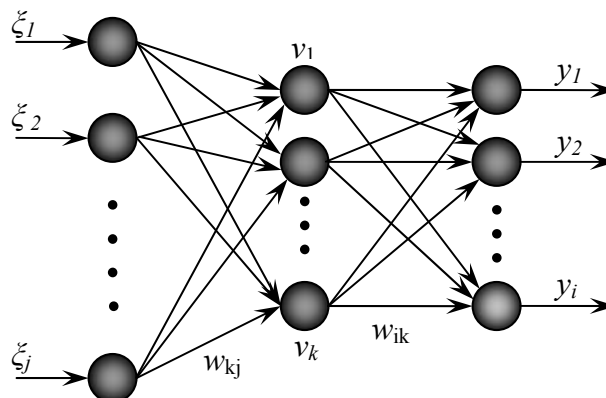


Figure V.10 : Perceptron Multi Couches

ζ_j : sont les unités d'entrées

y_i : sont les unités de sorties

v_k : sont les unités cachées

w_{kj} : sont les poids des connexions entre les unités d'entrées et les unités cachées

w_{ik} : sont les poids des connexions entre les unités cachées et les unités de sorties

Les étapes du développement de cet algorithme sont:

- Calcul de la valeur de sortie de l'unité cachée k

$$v_k = f(a_k) = f\left(\sum_j w_{kj} \zeta_j\right) \quad (\text{V.12})$$

- calcul de la valeur de sortie de l'unité de sortie i

$$y_i = f(a_i) = f\left(\sum_k w_{ik} v_k\right) \quad (\text{V.13})$$

- calcul des erreurs partielles et globales

$$E_p = \frac{1}{2} \sum_i (y_i - y_i^*)^2 \quad (\text{V.14})$$

et

$$E = \sum_p E_p \quad (\text{V.15})$$

- Minimisation de la fonction d'erreur globale par la descente du gradient. La valeur de chaque connexion est modifiée selon la règle :

$$\Delta w = -\eta \frac{\partial E}{\partial w} = -\eta \sum_p \frac{\partial E_p}{\partial w} \quad (\text{V.16})$$

Selon la position des poids de connexions, deux cas se présentent :

1. *Cas des connexions de la couche de sortie (w_{ik})*: Pour le cas des neurones de sortie, la relation (V.16) est fonction du paramètre w_{ik} qui influe uniquement sur la sortie du neurone d'indice i . La dérivée peut être décomposée de cette relation par :

$$\frac{\partial E_p}{\partial w_{ik}} = \frac{\partial E_p}{\partial y_i} \frac{\partial y_i}{\partial a_i} \frac{\partial a_i}{\partial w_{ik}} = (y_i - y_i^*) f(a_i) v_k \quad (\text{V.20})$$

La relation (V.16) devient :

$$\Delta w_{ik} = \eta \sum_p f(a_i) (y_i^* - y_i) v_k \quad (\text{V.21})$$

2. *Cas des connexions de la couche cachée (w_{kj})* : Si le neurone appartient à la couche cachée, la relation (V.16) est en fonction du paramètre w_{kj} qui influe non seulement sur la sortie du neurone k de la couche cachée, mais aussi sur tous les neurones i de la couche de sortie (en aval) qui lui sont connectés. Nous obtenons alors l'équation suivante :

$$\frac{\partial E_p}{\partial w_{kj}} = \frac{\partial E_p}{\partial v_k} \frac{\partial v_k}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} = \frac{\partial E_p}{\partial v_k} f(a_k) \zeta_j \quad (\text{V.17})$$

Le premier terme de cette expression devient alors :

$$\frac{\partial E_p}{\partial v_k} = \sum_i \frac{\partial E_p}{\partial y_i} \frac{\partial y_i}{\partial v_k} = \sum_i \frac{\partial E_p}{\partial y_i} \frac{\partial y_i}{\partial a_i} \frac{\partial a_i}{\partial v_k} = \sum_i (y_i - y_i^*) f(a_i) w_{ik} \quad (\text{V.18})$$

La modification des poids devient :

$$\Delta w_{kj} = \eta \sum_p \left(f(a_k) \zeta_j \left(\sum_i (y_i^* - y_i) f(a_i) w_{ik} \right) \right) \quad (\text{V.19})$$

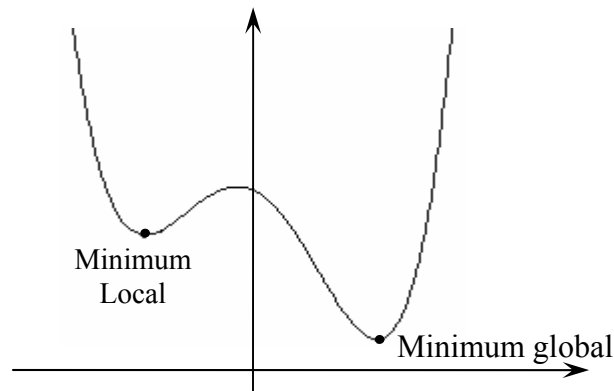


Figure V.11: Méthode de la rétropropagation du gradient

L'algorithme de rétropropagation du gradient de l'erreur a permis de dépasser les limites du perceptron simple. Il est capable de résoudre un grand nombre de problèmes de classification et de reconnaissance des formes. Cet algorithme souffre néanmoins de nombreux défauts, parmi lesquels [DAV90] :

- ☞ une limitation du temps de calcul (l'apprentissage est très long),
- ☞ une grande sensibilité aux conditions initiales, c'est-à-dire à la manière dont sont initialisés les poids des connexions,
- ☞ de nombreux problèmes dus à la géométrie de la fonction d'erreur (minimums locaux).
- ☞ le problème de dimensionnement du réseau dont la structure est fixé a priori. La structure est définie par le nombre de couches cachées, le nombre de neurones par couches et la topologie des connexions. Un mauvais choix de structure peut dégrader considérablement les performances du réseau.

V.9.4 Méthode de backpropagation avec momentum

Depuis l'apparition de l'algorithme de backpropagation, quelques variantes, visant à améliorer les performances de cet algorithme, ont été mises en œuvre. Parmi ces variantes, celle qui introduit un terme momentum dans la règle d'apprentissage pour améliorer la vitesse de convergence. Il s'agit en fait d'utiliser le gradient précédent pour donner une inertie à l'algorithme. Cette technique est utilisée aussi pour éviter que la descente de gradient ne se bloque dans un minimum local (Figure V.11) [FRE92].

La règle d'apprentissage devient:

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w}(t) + \alpha \Delta w(t-1) \quad (\text{V.22})$$

avec: $0.5 < \alpha \leq 1$ et $0 < \eta \leq 0.5$.

Le second terme de l'équation est appelé momentum.

V.9.5 Méthode du gradient conjugué

Cette méthode permet de changer rapidement la direction de recherche, pour obtenir l'optimum désiré. A chaque itération le coefficient d'apprentissage et le momentum sont modifiés. Cette méthode possède une vitesse de convergence supérieure à celle des algorithmes de backpropagation classiques, néanmoins elle nécessite plus de calculs.

L'équation de réajustement des poids est :

$$w(t+1) = w(t) + \eta d(t) \quad (\text{V.23})$$

Avec :

$$d(t) = -\frac{\partial E(t)}{\partial w(t)} + \beta(t-1) d(t-1) \quad (\text{V.24})$$

d : est la direction de la descente.

V.9.6 Algorithmes d'optimisation

Les algorithmes d'optimisation sont choisis de manière à permettre d'estimer le vecteur paramètres w pour lequel la fonction coût (J) choisie est minimale (plus la valeur de J est petite, plus le modèle produit fidèlement les observations utilisées pour l'apprentissage). Cette fonction coût est une fonction scalaire qui dépend du vecteur de paramètre w du modèle. Elle mesure l'écart entre les sorties du réseau de neurones et les sorties mesurées présentes dans l'ensemble d'apprentissage.

V.9.6.1 Méthode de Newton

L'idée de la méthode de Newton est d'utiliser la courbure (dérivée seconde) de la fonction coût pour atteindre le minimum plus rapidement.

$$w(t) = w(t-1) - H^{-1}(t) \nabla J(t) \quad (\text{V.25})$$

La direction de la descente est :

$$d(t) = H^{-1}(t) \nabla J(t) \quad (\text{V.26})$$

H : matrice dérivée seconde de J par rapport à w (Hessien)

∇J : dérivée première de J par rapport à w (gradient de $J(w)$)

Cette méthode est très efficace au voisinage d'un minimum. Cependant pour que la méthode converge vers un minimum, le Hessien doit être défini positif. Dans le cas général d'un modèle non linéaire, cette hypothèse de convergence n'est pas toujours respectée et la méthode peut ne pas converger. En pratique elle est peu employée car elle nécessite, de plus, le calcul de la matrice H à chaque itération.

V.9.6.2 Méthode de Levenberg-Marquardt

Cette méthode est particulièrement intéressante car elle s'adapte elle-même à la forme de la fonction de coût. Elle choisit automatiquement un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, si $\lambda(t)$ est grand, on retrouve la méthode du gradient et s'il est petit, la modification des paramètres correspond à celle de la méthode de Newton.

L'équation de la modification des poids est :

$$w(t) = w(t-1) - [H(t) + \lambda(t) I]^{-1} \nabla J(t) \quad (\text{V.27})$$

avec : I : matrice identité

λ : coefficient positif

V.10 LES ETAPES DE LA CONCEPTION D'UN RESEAU

Pour construire un réseau de neurones, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir ses données d'apprentissage et ceux du test de validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. La construction d'un réseau de neurones passe par quatre étapes importantes [HAR98] :

V.10.1 Choix des échantillons

Le choix des échantillons représente la première phase d'élaboration d'un réseau de neurones. Cette étape est déterminante pour déterminer le type de réseau approprié au problème à résoudre. La présentation de l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la manière dont il faudra mener l'apprentissage et la validation.

V.10.2 Normalisation des données

Afin d'améliorer la performance des réseaux neuronaux multicouches, il est important de normaliser les données d'entrée et de sortie de telle sorte qu'elles se trouvent dans l'intervalle $[-1, 1]$.

V.10.3 Elaboration de la structure du réseau

La structure du réseau dépend de près du type des échantillons utilisés. Il faut d'abord choisir le type de réseau. Dans le cas du réseau multicouche, il faudra choisir le nombre de neurones dans la couche cachée. Une des méthodes utilisée est de prendre une moyenne du nombre de neurones d'entrée et de sortie, mais il est plus judicieux de tester les différentes possibilités et choisir celle qui offre les meilleurs résultats.

V.10.4 Apprentissage

Cette phase consiste à calculer les pondérations optimales des différentes liaisons réseau de neurones, en utilisant un échantillon. Un algorithme d'entraînement est utilisé. Il consiste à présenter des valeurs à l'entrée et en fonction de l'erreur obtenue à la sortie, les poids accordés aux pondérations sont corrigés. C'est un cycle qui est répété jusqu'à ce que l'erreur du réseau soit minimale. Il faut prendre garde de ne pas surentraîner le réseau de neurones pour ne pas le spécialiser et le rendre moins performant.

V.10.5 Validation

Une fois le réseau calculé, il faut procéder à des tests pour vérifier que le réseau réagit bien comme souhaité, c'est la phase de validation. La méthode la plus simple étant de garder une partie de l'échantillon réservé à l'apprentissage, pour la validation.

V.11 APPLICATION DES RNA A LA COMMANDE DES PROCESSUS

Les réseaux de neurones artificiels constituent des outils particulièrement efficaces pour la modélisation et la commande des systèmes complexes ou mal définis. Leur utilisation pour la commande de processus non linéaires découle naturellement de leurs aptitudes à la modélisation [PER03].

Commander un processus, c'est déterminer l'action à lui appliquer, pour lui assurer un comportement donné, en dépit de perturbations. Cette action est délivrée par un organe ou une loi de commande, qui peut être réalisé par un réseau de neurones.

Il existe plusieurs architectures différentes pour la commande de processus par réseau de neuronal [REN95].

V.11.1 Système de commande par simple bouclage

La majeure partie des systèmes de commande neuronaux pour l'asservissement de poursuite appartient à la catégorie des systèmes dits par simple bouclage, dont le schéma est représenté sur la figure V.12. Dans ces systèmes, le signal de rétroaction est la sortie du processus. Les coefficients du correcteur sont mis à jour périodiquement si la commande est adaptative; dans le cas contraire, ses coefficients sont fixes une fois l'apprentissage terminé [PER03].

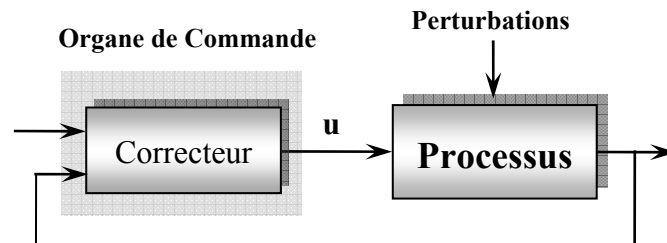


Figure V.12 : Système de commande par simple bouclage

V.11.2 Système de commande avec modèle interne

Dans les systèmes de commande avec modèle interne, le signal de rétroaction principal est l'erreur entre la sortie du modèle interne et celle du processus. Cette erreur est représentative des défauts de modélisation et de l'effet de perturbations. Grâce à cela, la conception d'un système de commande neuronal robuste vis-à-vis de ces perturbations est plus facile. De plus, l'utilisation de réseaux de neurones comme modèle interne peuvent aider à l'amélioration de ces systèmes de commande dans le cas de processus non linéaires.

La structure de commande avec modèle interne est représentée sur la Figure V.13.

Elle comprend :

- un modèle neuronal, qui est le modèle direct du processus.
- un correcteur neuronal qui est le modèle inverse du processus.

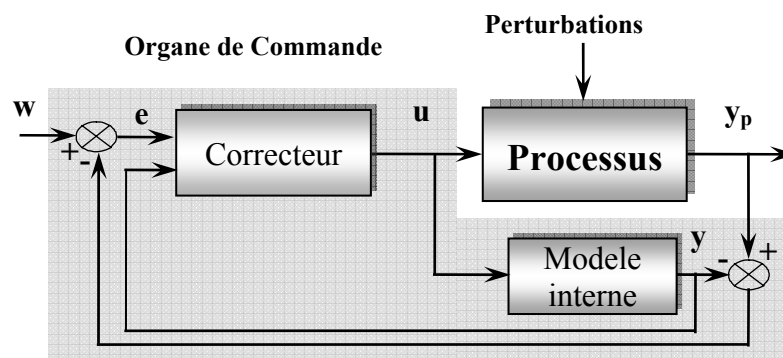


Figure V.13 : Système de commande avec modèle interne

V.12 CONCLUSION

Dans ce chapitre, nous avons effectué une présentation générale des réseaux de neurones. Nous avons introduit des définitions et des notions utiles dans leur application.

Les réseaux de neurones constituent une véritable solution pour la résolution de plusieurs problèmes, là où les méthodes classiques ont montré leurs limites. Leur utilisation s'est avérée efficace dans les processus qui nécessitent une interaction avec l'environnement, et ce par leur pouvoir d'adaptation souvent appelé « Plasticité synaptique ».

De plus, l'information à l'intérieur des réseaux de neurones, étant traitée en parallèle et d'une manière distribuée, fait que le temps de calcul des opérations est réduit. Ces caractéristiques les rendent plus utiles là où une solution classique existe déjà.

L'utilisation des réseaux de neurones artificiels (RNA) pour la commande a été explorée. L'emploi des réseaux de neurones plutôt que des techniques classiques pour commander des systèmes complexes peut se justifier par la simplicité de mise en œuvre (peu d'analyse mathématique préliminaire), par leur capacité d'approximation universelle, par la possibilité de considérer le processus comme une boîte noire et par la possibilité de débiter la conduite avec un minimum d'informations sur le processus. L'utilisation alors des réseaux de neurones pour l'asservissement du moteur synchrone à aimants permanents présente alors un intérêt considérable.

Le développement d'un régulateur neuronal et son application à la commande de vitesse de la MSAP fera l'objet de la suite de ce travail.

CHAPITRE

-VI-

APPLICATION DES RESEAUX DE NEURONES ARTIFICIELS POUR LA COMMANDE DE LA MSAP

VI APPLICATION DES RESEAUX DE NEURONES ARTIFICIELS POUR LA COMMANDE DE LA MSAP

VI.1 INTRODUCTION

Les réseaux de neurones formels sont devenus en quelques années des outils précieux dans des domaines très divers de l'industrie. La mise en oeuvre des réseaux de neurones est très simple. L'application de cette technique de manière inadaptée, ne peut conduire qu'à des déceptions. Pour apprendre, comprendre ou définir un processus d'apprentissage, il est nécessaire de modéliser l'environnement dans lequel opère le réseau neuronal. C'est-à-dire qu'il faut savoir quel type d'information sera fournie au réseau [HAR98]. Cette modélisation est le mode d'apprentissage. Ensuite, il faut comprendre comment le réseau met à jour les poids de ses connexions, c'est-à-dire quelles sont les règles d'apprentissage qui gouvernent la mise à jour. L'algorithme d'apprentissage est la procédure dans laquelle les règles d'apprentissage sont utilisées.

L'application des réseaux neuronaux dans la commande des machines, a permis la résolution de plusieurs problèmes liés au contrôle de ces systèmes, tout en répondant aux exigences de performance et de développement de ces dernières [REN95]. Le réseau neuronal peut être aisément construit par un choix simultané d'une architecture et de champs de neurones adéquats aux exigences du système de commande.

Dans ce chapitre, nous appliquerons les réseaux neuronaux à la commande en vitesse de la machine synchrone à aimants permanents. Nous déterminerons la meilleure structure à adopter, qui sera validée par les différentes simulations effectuées dans ce sens. Enfin, la robustesse de la commande avec régulateur neuronal pour des variations paramétriques sera étudiée.

VI.2 CONCEPTION DU REGULATEUR NEURONAL

Chaque réseau remplit une fonction bien définie, selon la complexité de la commande et les objectifs attribués à ce réseau. Cette fonction déterminera sa structure moyennant un algorithme d'apprentissage pour réaliser au mieux sa tâche.

Nous procédons en premier lieu, en choisissant des exemples d'apprentissage. Ces exemples sont issus des essais effectués sur le système de commande et exprimant le comportement recherché. Ces essais seront utilisés pour l'entraînement du réseau. En deuxième lieu, l'architecture du réseau sera déterminée après plusieurs simulations. Enfin, le réseau obtenu sera utilisé comme régulateur, afin de réaliser la commande en vitesse de la machine.

Il est à noter que pour notre étude, le régulateur neuronal utilisé est de nature échantillonné. La fréquence de calcul de la sortie correspond à la période d'échantillonnage de la machine.

VI.2.1 Création de la base d'apprentissage

Les exemples sont choisis à partir des essais effectués sur la machine commandée par le régulateur classique. L'essai choisi doit caractériser le fonctionnement désiré de la machine en régulation et en poursuite de la trajectoire de référence. Les exemples obtenus après échantillonnage serviront à la création de la base d'apprentissage. La figure IV.1 représente la

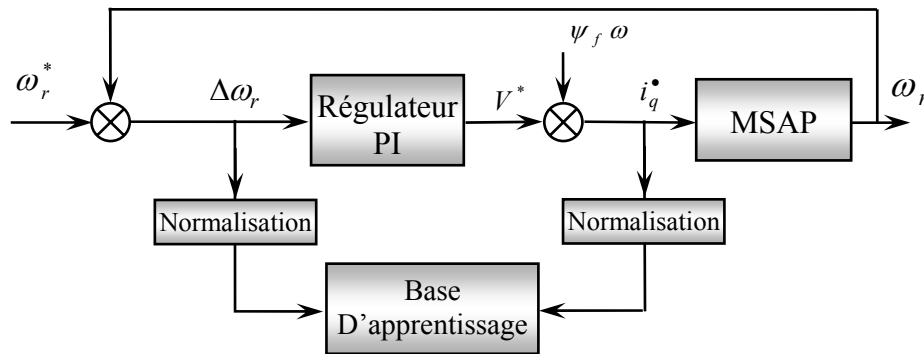


Figure VI.1 : Schéma de sauvegarde des données d'entraînement

procédure de collecte des données d'entraînement à partir d'exemples d'essais de la commande avec régulateur PI. Nous représentons, du schéma de commande proposé au chapitre II, juste la partie où se fait la récupération des exemples c'est à dire où se trouve le régulateur de vitesse. Les couples entrées sorties, sont sauvegardées dans un fichier de données, grâce à des convertisseurs analogiques numériques, pour nous permettre de prendre des valeurs échantillonnées (discrètes).

Pour la période d'échantillonnage le choix à été fait en respectant le théorème de Shannon, c'est à dire que cette valeur doit être inférieure à un demi de la plus petite constante de temps électrique τ_e .

VI.2.2 Procédures d'apprentissage

La phase d'apprentissage d'un réseau de neurone consiste à l'entraîner moyennant les données rassemblées auparavant. Cet entraînement permet de modifier progressivement les valeurs des poids synaptiques des couches cachées, à travers un algorithme d'apprentissage

L'apprentissage d'un réseau de neurone prend en considération plusieurs critères :

- ☒ l'architecture du réseau
- ☒ l'algorithme d'apprentissage
- ☒ la fonction d'activation
- ☒ L'initialisation des poids

Puisque nous disposons d'exemples d'entrées et de sorties, ce qui correspond à l'apprentissage supervisé, nous cherchons alors à construire un modèle capable de prédire la valeur de la sortie connaissant celle de l'entrée. L'algorithme utilisé, est celui de la rétro propagation de l'erreur. Cet algorithme charge les poids synaptiques du réseau en utilisant la méthode de Levenberg Marquardt qui permet une convergence rapide de l'algorithme.

La fonction d'activation définit l'état interne du neurone, nous avons opté pour la fonction tangente hyperbolique ou "tansig", car c'est la plus utilisée est la mieux adaptée.

Au départ, l'initialisation des poids est effectuée d'une façon aléatoire. Au cours de la phase d'apprentissage, si l'évolution de l'erreur décroît lentement ou se trouve être dans un minimum local, l'apprentissage est relancé avec les poids obtenus précédemment, jusqu'à aboutissement du résultat. Si l'erreur observée est trop importante la structure est revue en augmentant ou en diminuant le nombre de neurones ou le nombre de couches cachées.

VI.3 REGULTEUR NEURONAL DE VITESSE

Pour remplacer le régulateur de vitesse PI dans le schéma de la commande vectorielle de la MSAP, le nouveau contrôleur neuronal doit décrire la relation entre le courant i_q^* et l'erreur de vitesse [SOL00] [SOL99].

L'entraînement du réseau est effectué en utilisant l'algorithme d'apprentissage backpropagation de Levenberg–Marquardt pour sa convergence rapide. L'initialisation des poids du réseau dans notre cas, est choisi aléatoire [PAJ05].

Nous procédons à tester deux structure de réseau : la structure monocouche cachée et la structure à plusieurs couches (multi couche).

VI.3.1 Réseau à structure monocouche

Le choix de l'architecture du réseau de neurone artificiel se fait selon l'erreur quadratique moyenne (EQM) ou (MSE) obtenue au cours de l'apprentissage. La structure du réseau est obtenue après étude, et plusieurs essais, en faisant varier le nombre d'entrées, le nombre de couches cachées et le nombre de neurones dans chaque couche comme montré par les tableaux VI.1 et VI.2.

Tableau VI.1 : Entrées du RNA

		ANN Inputs
Input Number	1	$E(k)$
	2	$E(k), i_{qr}(k-1)$
	3	$E(k), E(k-1), i_{qr}(k-1)$
	4	$E(k), E(k-1), E(k-2), i_{qr}(k-1)$

Ou $E(k) = \Delta \omega_r(k)$; $E(k-1) = q^{-1} E(k)$;
 $i_{qr}(k-1) = q^{-1} i_{qr}(k)$; ($i_{qr} = i_q^*$)

Tableau VI.2 : Résultats d'apprentissage pour une structure monocouche

		Nbre de Neurone dans la CC		
EQM		4	6	10
Nbre d'Entrée	1	$7,376 \cdot 10^{-3}$	$8,981 \cdot 10^{-3}$	$6,936 \cdot 10^{-3}$
	2	$1,551 \cdot 10^{-5}$	$2,154 \cdot 10^{-5}$	$6,562 \cdot 10^{-6}$
	3	$5,287 \cdot 10^{-9}$	$5,878 \cdot 10^{-9}$	$8,112 \cdot 10^{-9}$
	4	$5,572 \cdot 10^{-9}$	$1,0 \cdot 10^{-9}$	$9,975 \cdot 10^{-8}$

Nous remarquons que, la structure à 4 entrées et 6 neurones en couche cachée, donne le meilleur résultat avec une erreur quadratique moyenne EQM = $1 \cdot 10^{-9}$.

L'architecture qui donne la plus petite valeur de l'erreur quadratique moyenne avec un temps de calcul acceptable est celle d'un réseau monocouche à quatre entrées et à six neurones dans la couche cachée. Ce réseau est montré par la Figure VI.2. Les neurones de la couche cachée ont une fonction de transfert sigmoïde non-linéaire et le neurone de sortie a une fonction de transfert linéaire. Les entrées du réseau sont l'erreur de vitesse à l'instant k , $k-1$ et $k-2$, et le courant d'axe q à l'instant $k-1$. La sortie est le courant d'axe q .

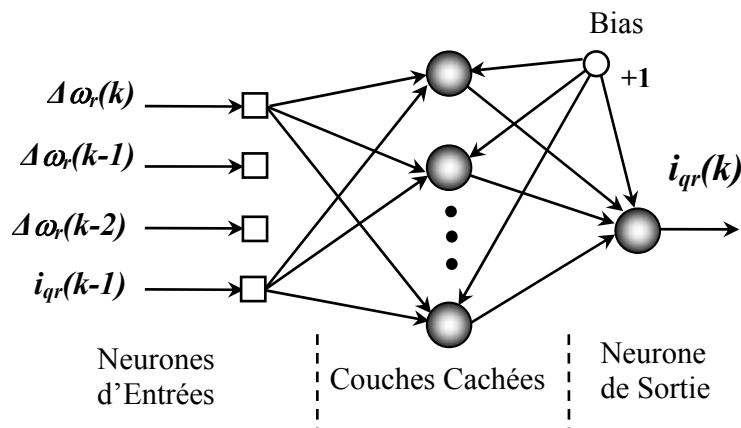


Figure VI.2 : l'Architecture du Réseau.

Pour ce réseau, la convergence atteint une erreur de 10^{-9} à la 168^{ème} époque qui est l'objectif demandé contrairement aux autres structures. L'augmentation du nombre d'entrées ne donne pas forcément une faible EQM [SAL98] [RAH98].

VI.3.2 Réseau à structure multicouche

Dans le Tableau VI.3, nous avons effectué l'apprentissage des réseaux de neurones multicouches pour un entraînement sur 250 epochs et un objectif de convergence de l'erreur quadratique moyenne de 10^{-9} . Les résultats d'apprentissage obtenus, pour des structures multicouches demandent un temps de calcul plus important, vu la complexité du réseau. Les valeurs des EQM sont présentées dans le tableau VI.3

Tableau VI.3: Résultats d'apprentissage pour une structure Multi couches cachées

		Nombre de neurones en couches cachées		
		3-4	5-6	3-5-4
Nbr d'entrées	1	$7.303 \cdot 10^{-3}$	$6.908 \cdot 10^{-3}$	$7.108 \cdot 10^{-3}$
	2	$1.750 \cdot 10^{-5}$	$5.480 \cdot 10^{-6}$	$1.295 \cdot 10^{-5}$
	3	$3.931 \cdot 10^{-8}$	$1.137 \cdot 10^{-8}$	$2.902 \cdot 10^{-5}$
	4	$1.076 \cdot 10^{-7}$	$1.433 \cdot 10^{-7}$	$1.432 \cdot 10^{-9}$

D'après les valeurs obtenues, nous remarquons que l'augmentation du nombre de couches cachées dans un réseau de neurone, ne donne pas forcément un meilleur résultat ou une erreur quadratique moyenne plus petite. Nous remarquons aussi que le temps d'apprentissage augmente avec le nombre de couches cachées vu le volume de calcul.

Nous optons donc, pour une structure mono couche, constituée de quatre entrées, et six neurones en couche cachée, représentée par la figure VI.2.

Pour l'apprentissage du réseau obtenu, il est nécessaire d'avoir assez de couple de données entrée-sortie. Puisque notre but est d'obtenir un contrôleur neuronale de vitesse qui couvre toutes les conditions de fonctionnement du système, les vecteurs appropriés d'entrée et de sortie sont obtenus par la simulation de la commande de la MSAP en utilisant un régulateur PI synthétisé pour différents points de fonctionnement.

Le processus d'apprentissage est off-line. Il est représenté par la figure VI.3.

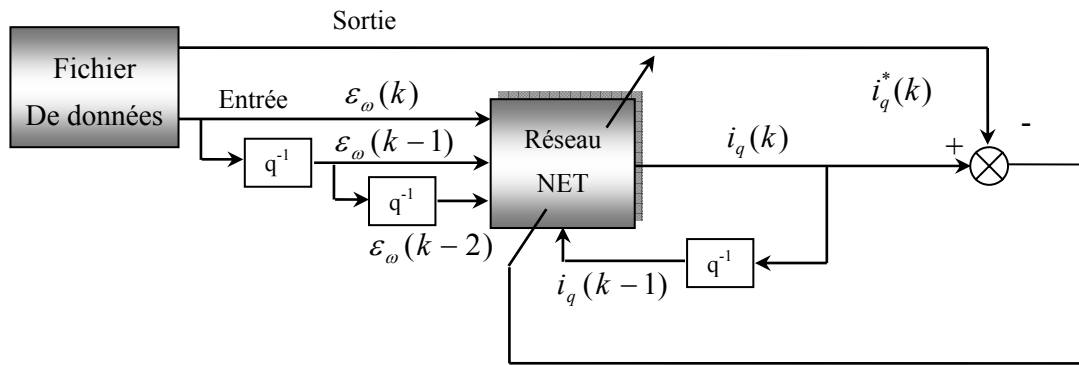


Figure VI.3 : processus d'apprentissage.

VI.3.3 Test et validation

Afin de tester l'efficacité de ce dernier et sa validation, nous simulons la machine à partir de données nouvelles qui n'ont pas été utilisées lors de l'apprentissage du réseau de neurone. La figure VI.4 représente l'évolution de l'erreur quadratique durant l'apprentissage ainsi que le test de validation du réseau obtenu.

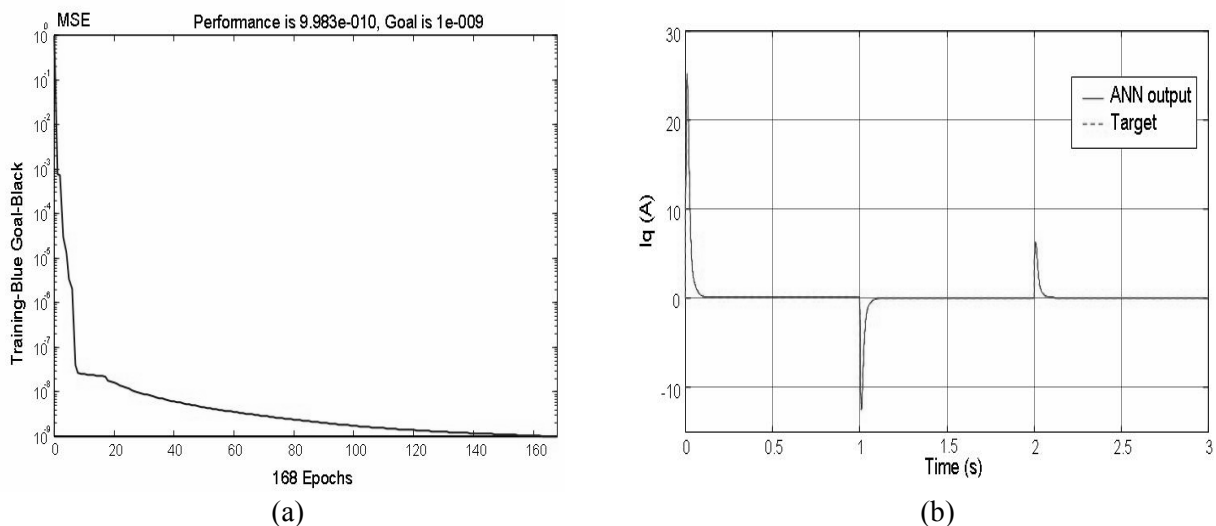


Figure VI.4 : Résultat de l'Apprentissage du Réseau de Neurones

(a) Convergence de l'erreur quadratique moyenne

(b) Test de Validation du Réseau

L'intérêt de ces tests, est de montrer que le réseau avec la structure déterminée précédemment, reproduit des exemples sur lesquels il n'a pas été entraîné. Ce qui valide le choix de la structure à quatre entrées et huit neurones en couche cachée.

VI.4 RESULTATS ET DISCUSSION

Une fois le régulateur neuronal est testé et validé, celui-ci est appliqué pour remplacer le régulateur PI afin de réaliser la commande en vitesse de la machine. Le schéma relatif à cette structure de commande est illustré par la figure VI.5.

Le comportement du système a été simulé en utilisant le logiciel de simulation Matlab-Simulink. Nous avons inséré un modèle de référence de la vitesse de consigne, pour générer une trajectoire de référence de la vitesse de rotation et réaliser un passage progressif de la vitesse de rotation au démarrage, ce qui permet un régime transitoire moins agité des grandeurs de la machine et une meilleure poursuite.

Les performances de la commande proposée ont été testées sur La machine M2, dont les paramètres sont donnés en annexe D.

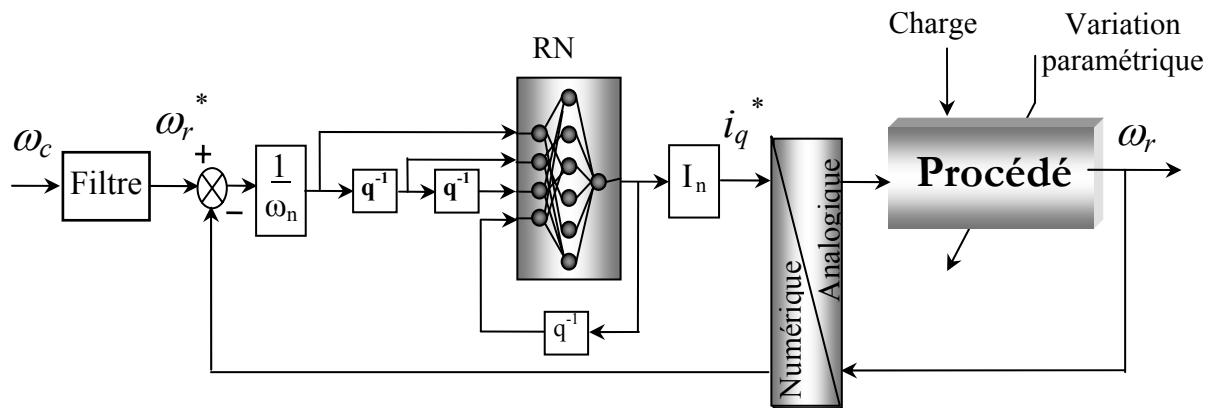


Figure VI.5 : Schéma de la commande en vitesse de la MSAP avec régulateur neuronal

L'évolution des variables de la machine, lors du démarrage utilisant le régulateur PI avec une vitesse de référence de 400 rad/s, est présentée dans la Figure VI.6. Le test comporte aussi l'application d'un échelon de couple résistant entre l'instant $k=0,3s$ et $k=0,8s$ et une variation de la consigne à 200 et 300 rad/s. la sortie commandée suit bien sa trajectoire de référence et la régulation est assurée face à la perturbation de charge.

Figure VI.7 présente la réponse du régulateur PI en présence d'une perturbation paramétrique due à une erreur sur la résistance ($R=5R_n$). Nous remarquons, que les performances du régulateur se détériorent considérablement.

L'utilisation du régulateur neuronal est explorée à partir de la Figure VI.8. Les mêmes conditions de tests avec PI sont utilisées. Les réponses montrent bien que la vitesse s'établit sans dépassement avec un faible temps de réponse et que le courant i_q se stabilise après des pics dus au démarrage. L'application des échelons de couple résistant ($C_r=5N.m$), permet d'observer le comportement du régulateur neuronal face à un rejet de perturbations.

La variation du paramètre R n'a pas affecté la qualité de la commande (Figure VI.9). La poursuite de la trajectoire de référence et le rejet de la perturbation paramétrique sont bien pris en charge par le régulateur neuronal de vitesse.

Pour mettre en évidence l'avantage du régulateur neuronal sur le PI, nous avons comparé leurs performances. Pour cela, la robustesse du régulateur neuronal par rapport aux perturbations paramétriques est testée pour une poursuite de trajectoire de vitesse de référence en présence des échelons de couple de charge. En Figure VI.10, nous avons comparé la vitesse de rotation pour les deux régulateurs en introduisant une erreur sur la résistance de la machine ($R=5R_n$). Une erreur sur l'inertie de la machine ($J_m=3J_m$) est testée en poursuite et régulation en Figure VI.11. Dans les deux cas, l'écart entre les deux réponses en termes de dépassement et rejet de perturbation est incontestable. Pour ces deux tests nous avons remarqué que l'action de la commande assure un établissement sans dépassement, une régulation et une poursuite de la référence.

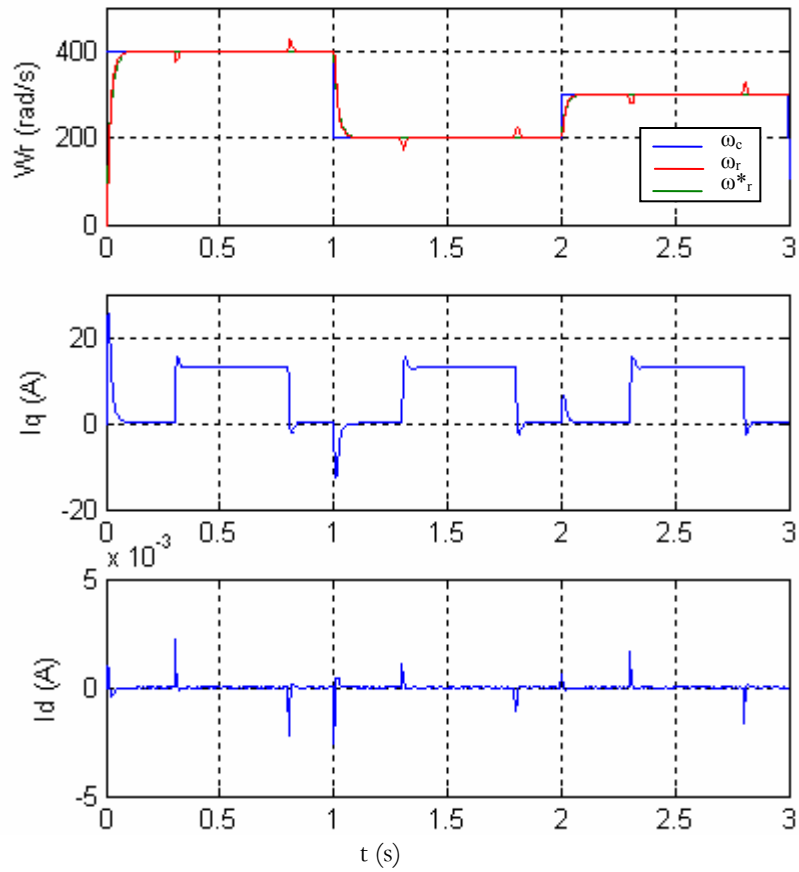


Figure VI.6: Poursuite de la trajectoire de référence en présence de la perturbation de charge utilisant le PI.

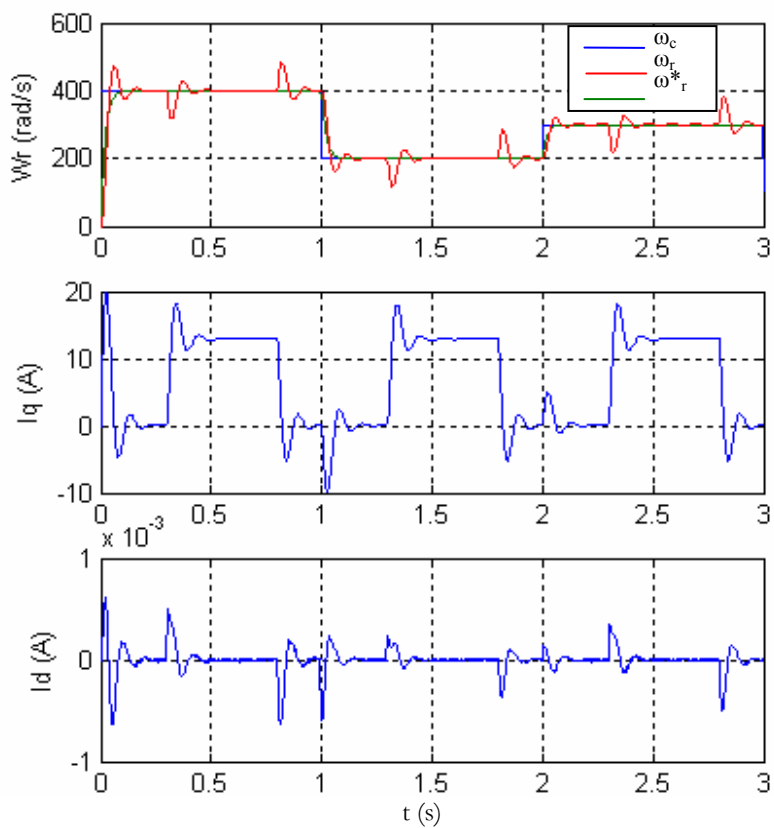


Figure VI.7 : Robustesse du PI vis à vis d'une erreur dans paramètre R , ($R=5R_n$).

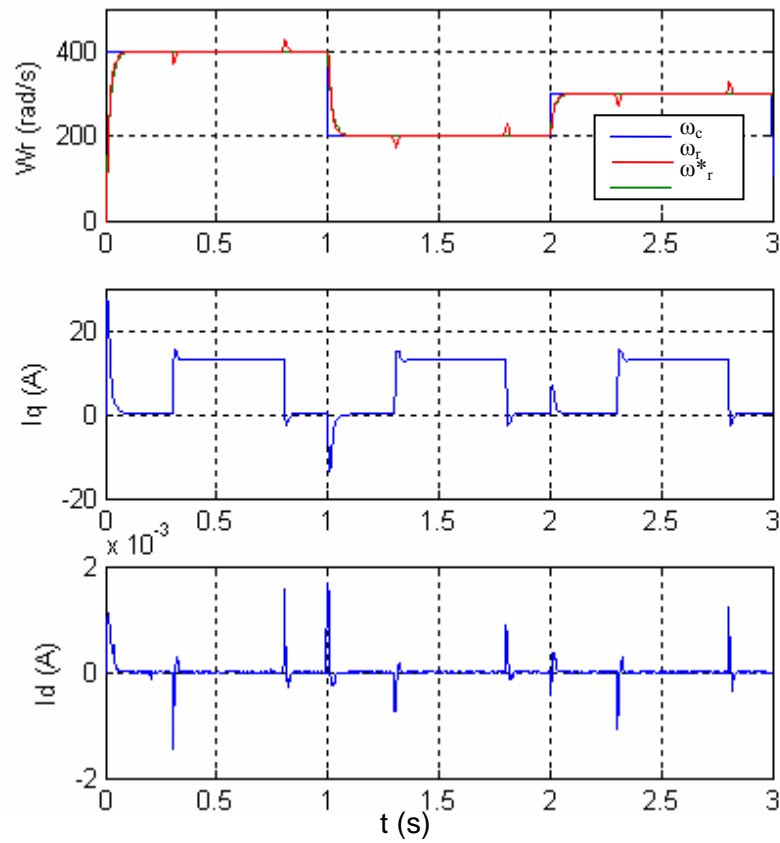


Figure VI.8: Poursuite de la trajectoire en présence de la perturbation de charge utilisant le correcteur RNA.

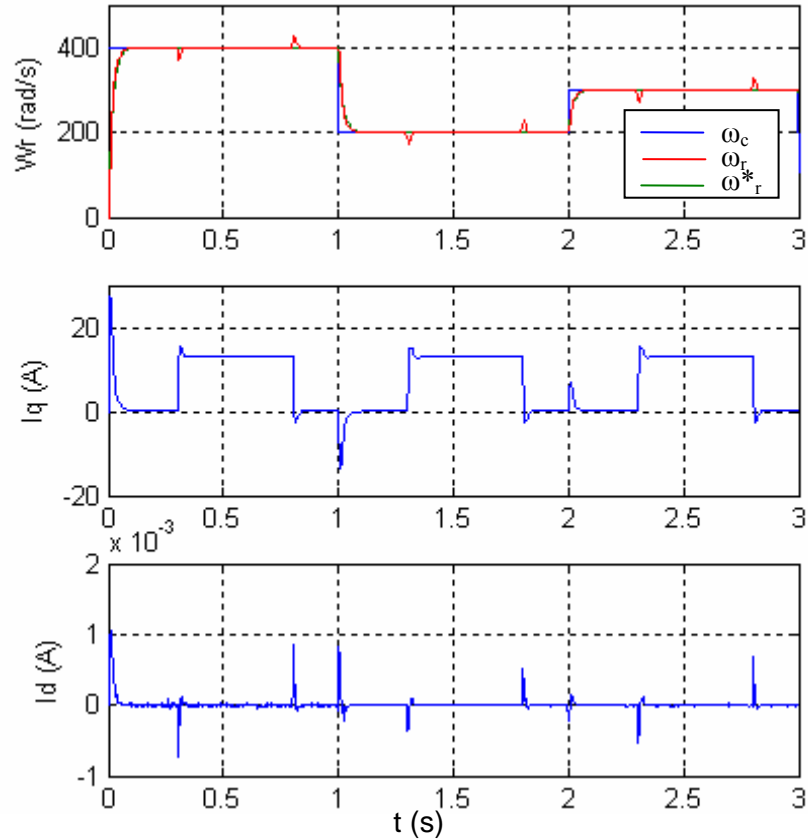


Figure VI.9: Robustesse du RNA vis à vis d'une erreur dans paramètre R ($R=5R_n$).

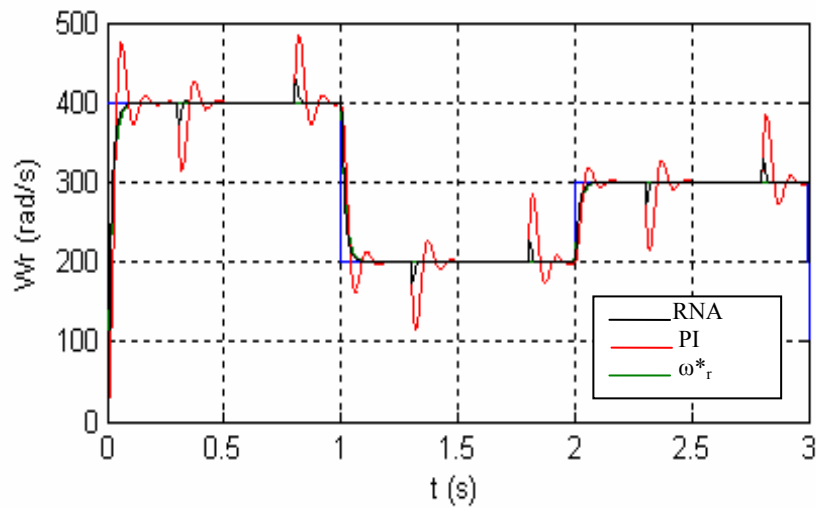


Figure VI.10: Comparaison entre la Robustesse du PI et du RNA à une erreur paramétrique ($R=5R$).

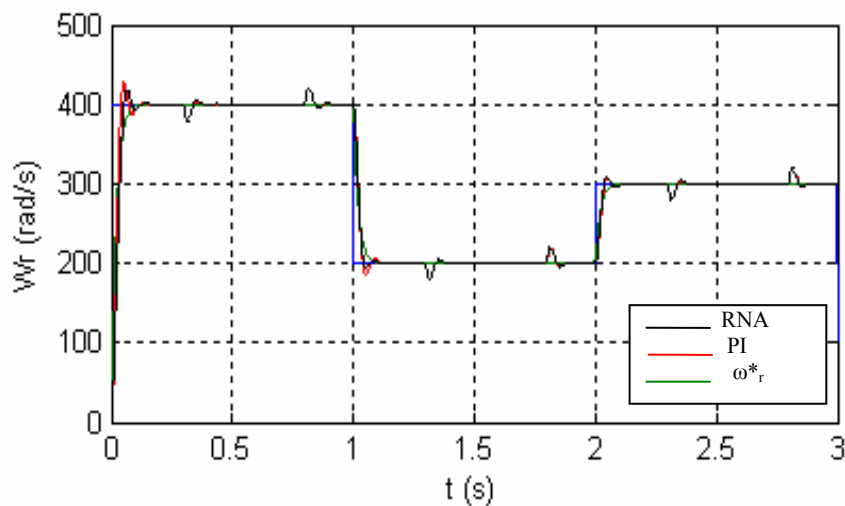


Figure VI.11 Comparaison entre la Robustesse du PI et du RNA à une erreur paramétrique ($J_m=3J_m$).

L'ensemble de ces résultats, montre les performances et la robustesse que procure le régulateur neuronal de vitesse pour la commande de la MSAP. En effet, il apparaît bien que le RNA de vitesse maintient la qualité de l'asservissement et du réglage aussi bien en présence d'une perturbation paramétriques qu'une perturbation de la charge.

VI.5 APPLICATION D'UN REGULATEUR NEURONAL MULTIVARIABLE (MIMO)

La commande à régulateur de vitesse neuronal permet d'avoir de meilleures performances que la commande à régulateurs classiques. Cependant, la machine synchrone à aimants permanents est un système multientrée-multisortie (MIMO) et son schéma de commande est multivariable et très encombrant car il contient deux régulateurs de courants et un de vitesse. Pour cela, un nouveau contrôleur neuronal multisorti est proposé pour prendre en charge le caractère multivariable du schéma de commande et remplacer les trois régulateurs du schéma de commande (Figure VI.12).

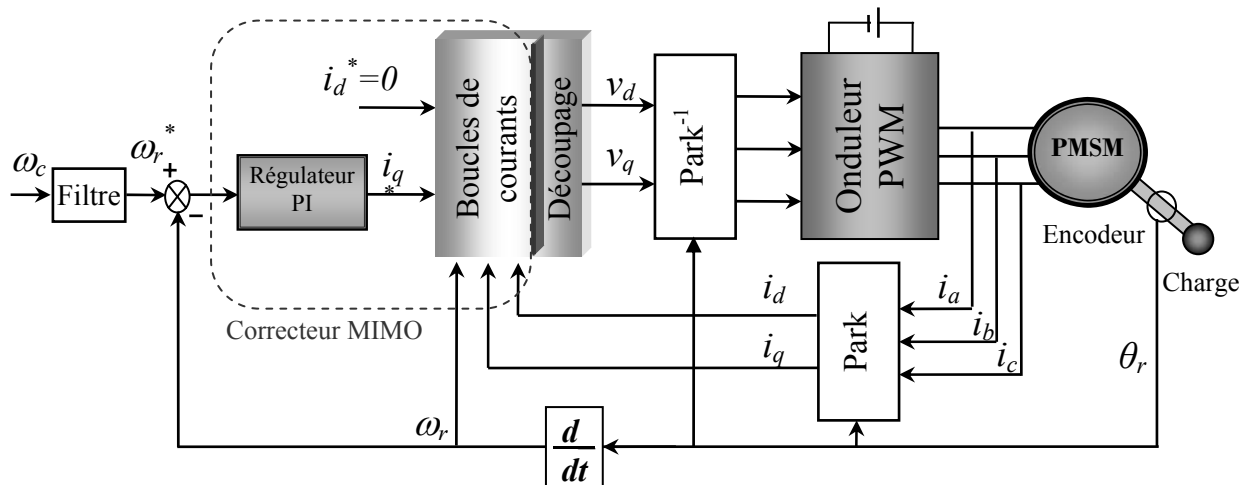


Figure VI.12 : Structure multivariable de la commande de la MSAP

VI.5.1 Architecture du réseau de neurones

Après avoir effectué plusieurs tests sur différentes structures, l'architecture du réseau MIMO adoptée, en tenant compte de la complexité du réseau et du temps de propagation de l'information dans ce dernier, contient huit entrées, une couche cachée avec quatre neurones et deux sorties comme montre dans Figure VI.13.

La fonction de transfert de type sigmoïde est retenue pour la couche cachée et une fonction de transfert linéaire les neurones de la couche de sortie. L'algorithme d'apprentissage utilisé est le même que celui pour le réseau précédent.

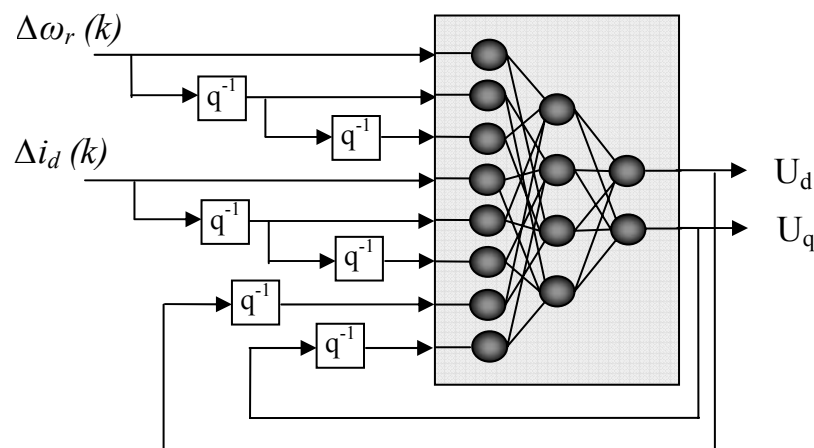


Figure VI.13 : Architecture du Contrôleur Neuronale Multivariable

VI.5.2 Apprentissage du RN

Les données utilisées pour l'apprentissage du réseau de neurone représentent tests de variation de la consigne de vitesse et d'une application du couple résistant. La fonction de transfert de type sigmoïde est retenue pour la couche cachée et une fonction de transfert linéaire les neurones de la couche de sortie. L'algorithme d'apprentissage utilisé est le même que celui pour le réseau précédent.

L'erreur quadratique moyenne obtenue au bout de 250 epoch pour la structure sélectionnée est de: $EQM = 1,38 \cdot 10^{-5}$.

VI.5.3 Résultats et discussion

La figure VI.14 montre une réponse en poursuite de vitesse de référence en présence d'une perturbation de charge. Le résultat observé en terme de dépassement et de rejet de perturbation est pratiquement le même que celui obtenu avec le RNA de vitesse.

Un deuxième test de poursuite, incluant une variation de la consigne avec application de couple résistant sur plusieurs niveaux de vitesse de rotation, est présenté dans la figure VI.15. Les courbes obtenues montrent bien l'efficacité du RNA-MIMO à assurer la poursuite.

La robustesse de ce régulateur par rapport aux perturbations paramétriques est testée pour une poursuite de la variation de vitesse de référence en présence des échelons de couple de charge (Figure VI.16). Le paramètre de la machine varié est le moment d'inertie J_m . Comme il est constaté, un petit dépassement apparaît à l'établissement de la réponse mais sans affecter la qualité de la commande. Par ailleurs le RNA assure la poursuite et le réglage sans être affecté par l'augmentation de J_m .

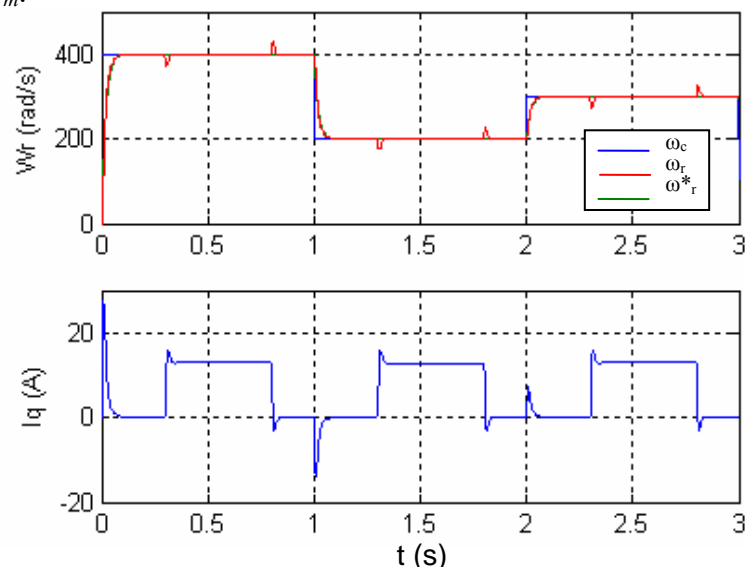


Figure VI.14 : Poursuite de la trajectoire avec couple de charge utilisant le RNA MIMO.

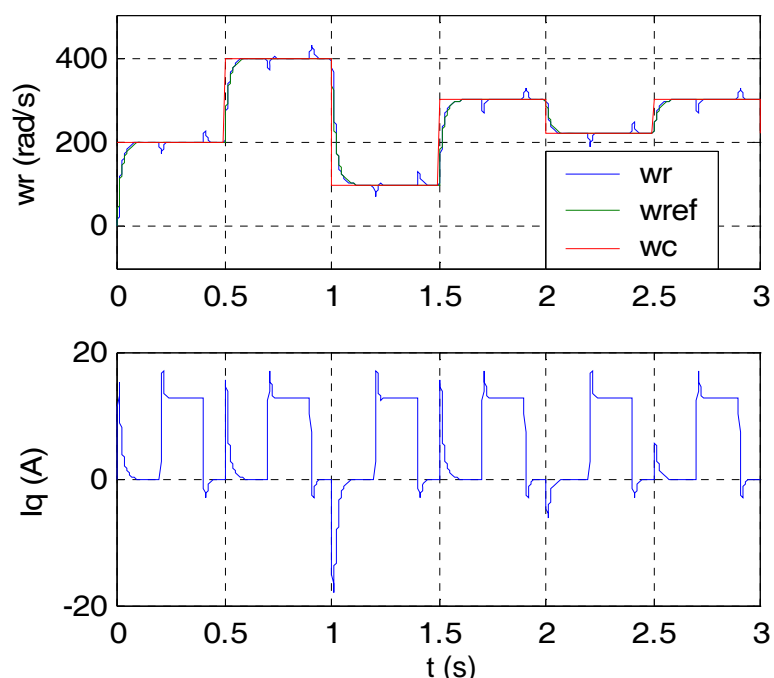


Figure VI.15 : Test de poursuite sur plusieurs niveaux de vitesse utilisant le RNA MIMO

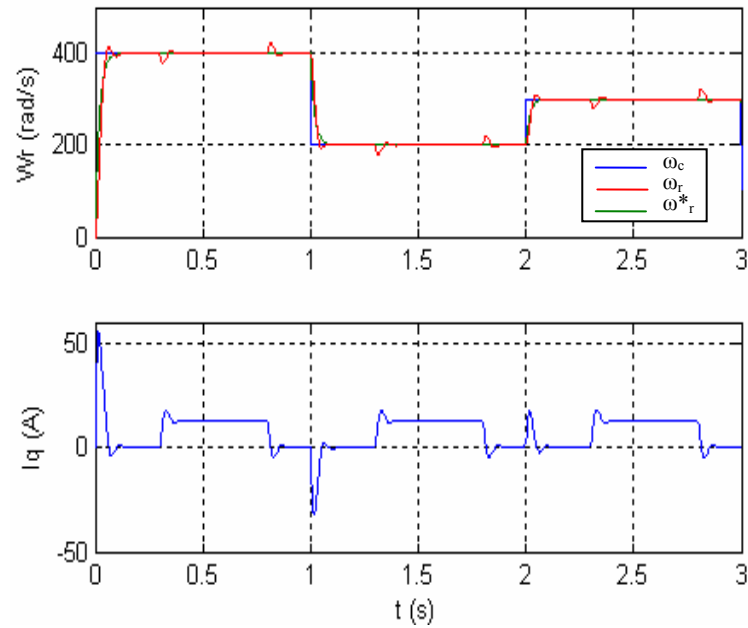


Figure VI.16 : Test de Robustesse du RNA MIMO vis à vis d'une variation de $3J_m$.

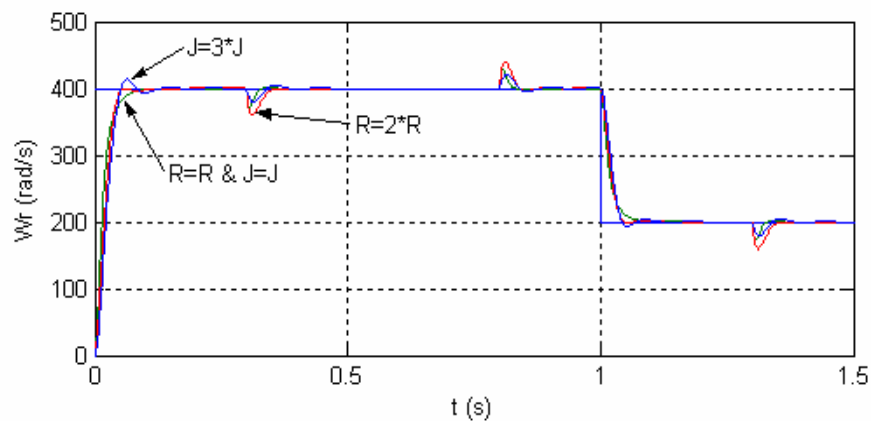


Figure VI.17 : Robustesse du RNA MIMO vis à vis de variations paramétriques.

La figure. VI.17 présente une comparaison entre plusieurs tests de robustesse de la commande utilisant le RNA multivariable aux variations paramétriques de la machine incluant les conditions de poursuite et de régulation. Une première courbe traduit la réponse du système avec la variation du moment d'inertie. La deuxième concerne la variation de la résistance statorique. La troisième courbe montre la réponse en vitesse en présence de variation des deux paramètres R et J_m .

De ce résultat, il est évident que le régulateur neuronal est capable d'assurer la poursuite et la régulation mais trouve des difficultés à maintenir les performances de la commande. Ceci est dû à la complexité du block du schéma globale de la commande remplacé par le RNA MIMO qui nécessite des investigations supplémentaires en terme des entrées et de la structure du réseau.

VI.6 CONCLUSION

La commande par réseau de neurones artificiel appliquée à la MSAP est présentée dans ce travail. L'approche adoptée consiste à remplacer le régulateur conventionnel de vitesse PI par un régulateur neuronal. Les capacités et l'efficacité du régulateur proposé ont été établies avec des tests de simulation en présence de variations paramétriques et des perturbations charge. Afin de comparer les performances du PI et du RNA, les tests sont effectués dans les mêmes conditions. Les résultats obtenus ont clairement montré l'avantage des réseaux de neurones dans le maintien de l'indice de performance.

Ces avantages sont traduits par la robustesse du RNA par rapport au PI vis-à-vis des variations des paramètres de la machine et la perturbation du couple de charge. Cependant, le choix de la structure interne du réseau reste aléatoire, ce qui n'est pas toujours facile.

La synthèse d'un régulateur neuronale multivariable, a permis de substituer les trois régulateurs PI du schéma de la commande vectorielle de la MSAP. Des résultats intéressants sont obtenus en régulation et poursuite de vitesse de la machine en présence des perturbations de charge. Les tests de robustesse, quand à la variation paramétrique que subit le modèle de la MSAP, se sont révélés prometteurs.

Par ailleurs, ce régulateur MIMO a permis de réduire le temps d'exécution et la complexité du schéma de commande vu qu'il remplace les régulateurs classiques ce qui est très important. Il est également préférable d'utiliser la commande à régulateur global qui s'avère plus simple et plus rapide donc plus efficace.

CONCLUSION GENERALE

CONCLUSION GENERALE

Dans ce travail, la commande adaptative et la commande par réseau de neurones artificiel ont été appliquées à l'asservissement de vitesse de la machine synchrone à aimants permanents.

Dans la première partie du travail, nous avons présenté la modélisation en vue de la simulation du variateur de vitesse constitué par l'ensemble MSAP-convertisseur statique. Malgré les hypothèses simplificatrices introduites, le modèle de la MSAP est très complexe (couplé, non linéaire et multivariable). La difficulté majeure, rencontrée dans cet actionneur, réside dans le fait que le couple électromagnétique et le flux sont des variables fortement couplées. L'utilisation de la transformation de Park, a permis d'aboutir à un modèle plus simplifié.

Le principe de la commande vectorielle retenue pour la MSAP a été appliqué au modèle obtenu. L'utilisation de la stratégie à i_d nul a permis de commander le couple électromagnétique par le courant i_q . Cette technique permet d'avoir un modèle semblable à celui de la machine à courant continu à excitation séparée. Dans le schéma de cette commande, les boucles de courants et celle de la vitesse sont assurées par des régulateurs classiques. Des tests de robustesse pour la boucle de vitesse dans le cas des variations paramétriques ont montré les limites des régulateurs utilisés.

Pour pallier ce problème, la commande adaptative a été introduite pour la boucle de vitesse dans un premier temps. L'approche directe du régulateur auto-ajustable a été retenue. Deux stratégies de commande ont été étudiées, la commande à erreur de prédiction minimale pondérée (EPMP) et la commande à variance minimale généralisée (VMG). Ce régulateur procède en deux étapes : estime en ligne les paramètres de la loi de commande, puis, génère la commande en utilisant une loi basée sur l'une des deux stratégies. L'algorithme d'estimation paramétrique choisi est celui des moindres carrés récursifs.

L'utilisation de la technique à EPMP obtenue en introduisant une pondération dans la structure du critère, de façon à prendre en charge la commande d'une large variété de modèle, a donné lieu à des résultats nettement meilleurs et le comportement transitoire de la MSAP s'est amélioré. Il suffit pour cela de faire le bon choix des polynômes de pondération. L'ensemble des résultats obtenus a montré satisfaction quand aux performances atteintes par le système. Celles-ci sont traduites par la robustesse de la commande vis-à-vis de la variation des paramètres de la machine et du couple de charge. Par ailleurs, la commande assure une bonne poursuite des variations du point de fonctionnement même en présence de perturbation de charge.

La commande auto-ajustable à variance minimale généralisée a été ensuite testée sur la machine. L'introduction du bruit de mesure sur la sortie de vitesse a permis la vérification de l'efficacité de cette technique à maintenir les performances désirées de la commande selon l'évolution des conditions opératoires. Les résultats obtenus ont montré la robustesse de la commande vis-à-vis de la variation des paramètres du système et de charge en présence du bruit de mesure. Il est montré aussi que cette commande réduit considérablement les oscillations des grandeurs de la machine en agissant sur les polynômes de pondération.

La commande par réseau de neurones artificiel appliquée à la MSAP a été abordée dans la seconde partie de ce travail. L'approche adoptée consiste à remplacer le régulateur conventionnel de vitesse PI par un régulateur neuronal. Après étude, un choix judicieux de l'architecture, du nombre d'entrées et de sorties, et les données de l'apprentissage, nous conduit à construire le régulateur neuronal. Les capacités et l'efficacité du régulateur proposé ont été établies avec des tests de simulation en présence de variations paramétriques et de perturbations de charge. Une comparaison des performances du PI et du RNA a clairement montré l'avantage des réseaux de neurones dans le maintien de l'indice de performance. Ces avantages sont traduits par la robustesse du RNA par rapport au PI vis-à-vis des variations des paramètres de la machine et la perturbation du couple de charge. Cependant, le choix de la structure interne du réseau reste aléatoire.

L'application des réseaux de neurones a été ensuite étendue à la régulation des deux composantes de Park du courant statorique de la MSAP. Une nouvelle structure multivariable du régulateur neuronale a été proposée. La synthèse de ce contrôleur est établie pour substituer le block des trois régulateurs du schéma de la commande en vitesse de la machine. Des résultats intéressants sont obtenus par ce contrôleur neuronal multi entrées-multi sorties, notamment en réponse de poursuite de la vitesse de référence en présence de perturbations de charge et des variations paramétriques. Cette commande par RNA-MIMO est simple et moins encombrante et possède aussi l'avantage d'un temps d'exécution relativement faible. Cependant, les tests de robustesse se sont révélés peu efficaces à cause de la sensibilité du contrôleur à la variation de certains paramètres. Ceci est du principalement à la difficulté de conception de sa structure d'une part et à la détermination de ses paramètres par apprentissage d'autre part.

Des travaux restent à faire pour concevoir une architecture adéquate pour la commande neuronale multivariable. En effet, une des voies à explorer est d'utiliser les plans d'expérience pour optimiser l'ensemble des données entrées-sorties du fichier d'apprentissage et ceux pour ne pas sous entraîner ou surentraîner le réseau jusqu'à devenir spécialisé. Une deuxième voie à envisager est d'associer aux réseaux de neurones la commande adaptative et là, plusieurs configurations sont possibles.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [ARA98] D. Arab Khabouri "Contribution à la commande vectorielle des machines synchrones à aimants permanents sans capteurs mécaniques ", Thèse de Doctorat de l'INP Lorraine, Nancy, 1998.
- [ÅST73] K.J. Astrom, B. Wittenmark, "on self tuning regulators", *Automatica*, Vol. 9, pp. 185-199, 1973.
- [ÅST89] K.J. Astrom, B. Wittenmark, *Adaptive control*", Addison Wesley, 1989.
- [BAG99] D. L Baghlii "Contribution à la commande de la machines asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques", Thèse de Doctorat de l'UHP, Nancy-1, 1999.
- [BER02] Nicolas BERNARD, "Machine Synchrone : de la boucle ouverte à l'autopilotage", *Revue 3EI*, n° 30, septembre 2002, pp. 24-39.
- [BOU95] M.S. Boucherit "Sur l'application de l'automatique moderne dans la commande des machines électriques", Thèse de Doctorat d'Etat, ENP, 1995.
- [BOU99] A. Bouscayrol, T. Communal, "Approche Globale de la Commande Dynamique de Machines Electriques", *Revue 3EI*, No. 17, Juin. 1999, pp. 73-79.
- [CHA87] V. Chalam, *Adaptive Control Systems: Technics and Applications*, Marcel Dekker Inc, New York. basel, 1989.
- [CIR02] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, *Neural and Fuzzy Logic Control of Drives and Power Systems*, Newnes press, Great Britain, 2002.
- [CLA75] D.W. Clarke, P.J. Gawthrop, "Self-tuning controller", *IEE proc.* , Vol. 122, No. 9, pp. 929-934, 1975.
- [CLA79] D.W. Clarke, P.J. Gawthrop, "Self-tuning control", *IEE proc.* , Vol. 126, No. 6, pp. 633-640, 1979.
- [DAU00] Emmanuel Daucé, "Adaptation dynamique et apprentissage dans des réseaux de neurones récurrents aléatoires", Thèse de Doctorat de l'ENSAE, Toulouse, 2000.
- [DAV90] E. Davalo, P. Naim, " *Les réseaux de neurones* ". Edition Eyrolles, Paris, 1990.
- [DRE98] G. DREYFUS, "Les réseaux de neurones", *Mécanique Industrielle et Matériaux*, n°51 (septembre 1998).
- [FRE92] James A. Freeman, David M. Skapura, " *Neural Networks Algorithms, Applications and Progaming TechniquesI*", Addison Wesley, 1992.
- [FU 91] Yayuan Fu, "Commande découplées et adaptatives des machines asynchrones triphasées", Thèse de doctorat de l'université de Montpellier 2, France, 1991.
- [FU 95] Yaqun Fu, "Commande vectorielle des machines à courant alternatif : étude des problèmes liés à la numérisation des régulateurs", Thèse de doctorat de l'université Catholique de Louvain, Louvain, 1995.
- [GOO84] G.C. Goodwin, K.S. Sin, " *Adaptive prediction filtering and control*", Prentice Hall, 1984.
- [GRE94] G.Grellet. G. Clerc, " *Actionneurs électriques, Principes – Modèles - Commande*", Edition Eyrolles, Paris, 1994.
- [HAR98] Y. HARKOUSS, "Application de réseaux de neurones à la modélisation de composants et de dispositifs micro-ondes non linéaires", Thèse de doctorat de l'Université de Limoges, décembre 1998.
- [IBR02] Z. Ibrahim and E. Levi, "A Comparative Analysis of Fuzzy Logic and PI Speed Control in High-Performance AC Drives Using Experimental Approach", *IEEE Trans. on Industry Applications*, Vol. 38, No. 5, September/October 2002, pp. 1210-1218.

- [KAD00] A. Kaddouri, "Etude d'une commande non linéaire adaptative d'une machines synchrones à aimants permanents", Thèse de Doctorat de l'Université Laval, Québec, 2000.
- [KIM03] Y. S. Kim, S. Kyoon and Y. A. Kwon, "MRAS Based Sensorless Control of Permanent Magnet Synchronous Motor", *SICE Annual Conference*, Fukui, Japan, August 4-6, 2003, pp. 1632-1637.
- [KUK99] Dragan Kukolj, Filip Kulić and Emil Levi, "Artificial Intelligence Based Gain Scheduling of PI Speed Controller in DC Motor Drives", *ISIE'99*, Bled, Slovenia, 1999, pp. 425-429.
- [LAJ99] M. Lajoie-Mazenc, P. Viarouge, "Alimentation des machines synchrones", *Techniques de l'Ingénieur, Traité Génie Electrique*, D3630, 1999.
- [LAN79] I.D. Landau, *Adaptive control – the model reference approach*. New York: Marcel Dekker Inc, 1979.
- [LAN80] I.D. Landau, "Dualité asymptotique entre les systèmes de commande adaptative avec modèle et les régulateurs à variance minimale auto-ajustables", *R.A.I.R.O. Automatique/Systems Analysis and Control*, Vol.14, No.2, pp.189-204, 1980.
- [LAN81] I.D. Landau, "Model reference adaptive controllers and stochastic self-tuning regulator – A unified approach", *Journal of Dynamic Systems, Measurement and control*, Vol.103, pp.404-416, 1981.
- [LAN86] I.D.Landau, *Commande adaptative : Aspects Pratiques et Théoriques*, Masson, 1986.
- [LAN93a] I.D. Landau, *Identification et commande des systèmes*, (seconde édition), Hermès, Paris, 1993.
- [LAN93b] I.D. Landau, F. Rolland, "Identification and digital control of electrical drives", *Control Eng Practice*, Vol. 1, No. 3, pp. 539-546, 1993.
- [LIN97] F.J. Lin, "Real-Time IP Position Controller Designe with Torque Feedforward Control for PM Synchronous Motor", *IEEE Trans. on Industry Electronics*, Vol. 44, No. 3, June. 1997, pp. 398-406.
- [LIN98] Faa-Jeng Lin, Rong-Jong Wai and Hong-Pong Chen, "A PM Synchronous Servo Motor Drive with an On-Line Trained Fuzzy Neural Network Controller", *IEEE Trans. on Energy Conversion*, Vol. 13, No. 4, December 1998, pp. 319-325.
- [LIU98] T. LIU, I. HUSAIN et M. ELBULUK, "Torque Ripple minimization with on-line parameter estimation using neural networks in permanent magnet synchronous motors", *Proceeding de IEEE, Annual Meeting 1998*, pp.375-380.
- [LOU99] J.P. Louis, C. Bergmann, "Commande numérique des machines synchrones", *Techniques de l'Ingénieur, Traité Génie Electrique*, D3644, 1999.
- [MCC43] Mc Culloch W.S., Pitts, "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*. 5, (1943) 115-133.
- [MED98] D. MEDDAH, "Identification et commande neuronales des systèmes non linéaires: application aux systèmes robotisés", Thèse de doctorat, de l'université de pierre et marie curie, 1998.
- [MIL92] W. T. Miller, R. S. Sutton, and P. J. Werbos, "*Neural Networks for Control*", MIT Press, 1992.
- [MIN04] Liu Mingji, Cai Zhongqin, Cheng Ximing and Ouyang Minggao, "Adaptive Position Servo Control of Permanent Magnet Synchronous Motor", *Proc. of American Control Conference, Boston, Massachusetts*, June/July 2004, pp. 84-89.
- [NAH01] Babak NAHID MOBARAKEH, "Commande vectorielle sans capteur mécanique des machines synchrones à aimants: méthodes, convergence, robustesse identification "en ligne" des paramètres", Thèse de Doctorat de l'INP Lorraine, Nancy, 2001.
- [NAI04a] A. Nait Seghir, M.S. Boucherit, "Control of Permanent Magnet Synchronous Motor Using

- Neural Networks”, *Proceeding of the MS'04, Lyon-Villeurbanne, 5-7, July*, pp 3.9-3.12, 2004.
Selected at AMSE & University of Lyon conference for publication in special issue 2004 “Best Book”, pp 157-165.
- [NAI04b] **A. Nait Seghir**, M.S. Boucherit, “Adaptative Speed Control of Permanent Synchronous Motor”, *IEEE International Conference on Electrical, Electronic and Computer Engineering, IEEE - ICEEC'04, Cairo-Egypt, 5th – 7th, September, 2004*.
- [NAI04c] **A. Nait Seghir**, M.S. Boucherit, “Commande Adaptative en Vitesse de la Machine Synchrone à Aimants Permanents”, *Proceeding of the CEE'04, Batna, 4th - 6th, October*, pp 129 – 132, 2004.
- [NAI05] **A. Nait Seghir**, M.S. Boucherit, “Self tuning control of Permanent Magnet Synchronous Motor”, *International conference on Modelling and Simulation (General applications and Embedded Systems engineering) ESIGELEC, Rouen- France, 6-8 juillet 2005*.
- [NAI06a] **A. Nait Seghir**, M.S. Boucherit, “A Direct Adaptive Control of Permanent Magnet Synchronous Motor”, *International Review of Electrical Engineering (I.R.E.E.)*, Vol. 1, N. 2, May-June 2006. pp. 304-310.
- [NAI06b] **A. Nait Seghir**, M.S. Boucherit, “Application des Réseaux de Neurones pour la Commande de la Machine Synchrone à Aimants Permanents”, *Proceeding of the CEE'06, Batna, 7th - 8th, November*, pp 13 – 18, 2006.
- [NAI06c] **A. Nait Seghir**, M.S. Boucherit, “Artificial Neural Network to Improve Control of Permanent Magnet Synchronous Motor”, *6th WSEAS International Conference on POWER SYSTEMS (POWER'06)*, Tenerife, Espagne, December 16-18, 2006.
- [NAI07] **A. Nait Seghir**, M.S. Boucherit, “A New Neural Network Based Approach for Speed Control of PM Synchronous Motor”, *WSEAS Transactions On Circuits and Systems*, Issue 1, Vol. 6, January 2007. pp 87-93, ISSN: 1109-2734.
- [NAI96] **A. Nait Seghir**, M.S. Boucherit, D. Boukhetala et L. Nezli, “Commande adaptative d'une machine synchrone autopilotée”, *COMAEI'96, Vol.3*, pp.161-165, Tlemcen, 1996.
- [NAI97a] **A. Nait Seghir**, *Identification et Commande adaptative d'une machine synchrone autopilotée alimentée par onduleur de tension*. Thèse de Magister, ENP, Alger, 1997.
- [NAI97b] **A. Nait Seghir**, M.S. Boucherit, D. Boukhetala et L. Nezli, “Commande auto-ajustable en vitesse d'une machine synchrone autopilotée alimentée en tension”, *Actes des 2^{èmes} JST'97, 05 & 06 Mai, INH, Boumerdes, Vol.2*, pp.397-400, 1997.
- [NAI98] **A. Nait Seghir**, M.S. Boucherit, D. Boukhetala et L. Nezli, “Commande adaptative à erreur de prédiction minimale pondérée d'un ensemble onduleur de tension-machine synchrone”, *Proceeding of the ICEL'98, Oran, 5th & 6th, December*, pp 242-246, 1998.
- [NAJ82] K.Najim, *Commande adaptative des processus industries*, Masson, 1982.
- [OUA04] M. Ouassaid, M. Cherkaoui and Y. Zidani, “A Nonlinear Speed Control for a PM Synchronous Motor Using an Adaptive Backstepping Control Approach”, *IEEE International Conference on Industrial Technology (ICIT)*, 2004, pp. 1287-1292.
- [PAJ05] T. Pajchrowski and K. Zawirski, “Synthesis of robust speed controller based on artificial neural network for permanent magnet synchronous motor”, *Proc. PELINCEC'05, Warsaw, Poland*, 2005, paper ID41 on CD-ROM.
- [PER03] L. Personnaz et I. Rivals, *Réseaux de Neurones Formels pour la Modélisation, la commande et la classification*, CNRS Editions, Paris 2003.
- [PIL89] P. Pillay and R. Krishnan, “Modeling, Simulation and Analysis of Permanent-Magnet Motor Drives, Part I: the Permanent-Magnet Motor Drive”, *IEEE Trans. on Industry Applications, Vol. 25, No. 2, March./April. 1989*, pp. 265-273.
- [RAH98] M. A. Rahman and M. A. Hoque, “On-Line Adaptive Artificial Neural Network Based Vector Control of permanent Magnet Synchronous Motors”, *IEEE Trans. on Energy*

- Conversion*, Vol. 13, No. 4, December. 1998, pp. 311-318.
- [REN95] J. M. Renders, *Algorithme génétique et réseaux de neurones, Application à la commande des processus*, Hermes, Paris, 1995.
- [ROB95] B. Robyns, Y. Fu, F. Labrique et H. Buyse, “Commande numérique de moteurs synchrones à aimants permanents de faible puissance”, *J. Phys*, III. T5, N°8 August 1995, pp. 1255-1268.
- [ROS60] Rosenblatt : “Perceptron simulation experiments”. *Proceedings of the I. R. E.* 1960, pp. 167-192.
- [SAL98] KH.I. SALEH, M.A. BADR, A.S ELWER et S. WAHSH: “Analysis of controlled permanent magnet synchronous motors using artificial neural network”, *IEEE, Trans. On Energy conversion*, vol. 5 September 1998.
- [SEI88] P.F.Seixas, “Commande d’une machine synchrone autopilotée”, Thèse de Doctorat de l’INP Toulouse, 1988.
- [SHE99] Tsong-Terng Sheu and Tien-Chi Chen, “Self-tuning Control of Induction Motor Drive Using Neural Network Identifier”, *IEEE Trans. on Energy Conversion*, Vol. 14, No. 4, December 1999, pp. 881-886.
- [SHI03] Seung C.Shin and Zeungnam Bien, “Adaptive Neural Network -Based Predictive Control for Nonlinear Dynamical Systems”, *Intelligent Automation and Soft Computing*, Vol. 9, No. 1, 2003, pp. 31-42.
- [SIL79] H. M. Silveira, I. D. Landau, “Une nouvelle méthode pour la synthèse des systèmes adaptatifs avec modèle de référence”, *R.A.I.R.O. Automatique/Systems Analysis and Control*, Vol.13, No.1, pp.39-56, 1979.
- [SOL00] M. Solazzi and A. Uncini, “On-line Quality Control of Permanent Magnet Motor Using Neural Networks”, *IEEE Trans. on Industry Applications*, Vol. 36, No. 3, May/June. 2000, pp. 265-273.
- [SOL99] Mirko Solazzi and Aurelio Uncini, “On-line Quality Control of DC Permanent Magnet Motor Using Neural Networks”, *XI Italian Workshop on Neural Nets (WIRN’99), Vietri Sul Mare (SA), Italy*, May 20-22, 1999.
- [TAM97] Shin’ichi Tamura and Masahiko Tateishi, “Capabilities of a four-Layered Feedforward Neural Network: Four Layers Versus Three”, *IEEE Trans. on Neural Networks*, Vol. 8, No. 2, March 1997, pp. 251-255.
- [YAN03] Simon X. Yang, Hao Li and John Shi, “Modeling of Supercritical Fluid Extraction by Neural Networks”, *Intelligent Automation and Soft Computing*, Vol. 9, No. 1, 2003, pp. 3-12.
- [ZHI01] Zhihong Su and K. Khorasani, “A Neural- Network- Based Controller for a Single- Link Flexible Manipulator Using the Inverse Dynamics Approach”, *IEEE Trans. on Industrial Electronics*, Vol. 48, No. 6, December 2001, pp. 1074-1086.
- [ZHU01] Guchuan Zhu, Azeddine Kaddouri, Louis-A Dessaint and Ouassima Akhrif, “A Nonlinear State Observer for the Sensorless Control of a Permanent-Magnet AC Machine”, *IEEE Trans. on Industrial Electronics*, Vol. 48, No. 6, December 2001, pp. 1098-1108.

ANNEXES

- A- ALGORITHME D'ADAPTATION PARAMÉTRIQUE
- B- FORME PREDICTIVE
- C- TRANSFORMATION DE PARK
- D- PARAMETRES DES MACHINES UTILISEES

ANNEXE A

ALGORITHME D'ADAPTATION PARAMÉTRIQUE

A.1 ALGORITHME D'ADAPTATION PARAMÉTRIQUE

Le calcul de la commande et l'estimation des paramètres du modèle du procédé, ou ceux du régulateur, sont deux tâches séparées. La synthèse de l'algorithme d'estimation paramétrique se fait donc indépendamment de la méthode de commande choisie.

Nous présentons ici deux algorithmes d'adaptation paramétriques, à savoir:

- ☐ L'algorithme du gradient simple.
- ☐ L'algorithme des moindres carrés récursif.

Ces algorithmes sont utilisés pour l'estimation en temps réel des paramètres inconnus et éventuellement variables dans le temps du modèle du processus ou du régulateur.

Considérons le système décrit par le modèle déterministe suivant :

$$A(q^{-1})y(t) = B(q^{-1})u(t) \quad (\text{A.1})$$

où A et B sont des polynômes en q^{-1} d'ordre n et m respectivement. q^{-1} est l'opérateur retard défini par :

$$q^{-1}y(t) = y(t-1)$$

Nous introduisons le vecteur des paramètres:

$$\theta = [a_1, \dots, a_n, b_1, \dots, b_m]^T$$

et le vecteur de mesures

$$\phi(t-1) = [-y(t-1), \dots, -y(t-n), u(t-1), \dots, u(t-m)]^T$$

Le modèle (A.1) est réécrit sous la forme de régression :

$$y(t) = \phi(t-1)^T \theta \quad (\text{A.2})$$

A chaque pas d'échantillonnage, la nouvelle estimation des paramètres du modèle, sera obtenue en minimisant un critère d'erreur. Le calcul récursif de θ sera obtenu par une expression générale de la forme:

$$\begin{pmatrix} \textit{nouvelle} \\ \textit{estimation} \\ \textit{des paramètres} \end{pmatrix} = \begin{pmatrix} \textit{estimation} \\ \textit{precedente} \end{pmatrix} + \begin{pmatrix} \textit{gain} \\ \textit{d'adaptation} \end{pmatrix} \cdot \begin{pmatrix} \textit{fonction} \\ \textit{des mesures} \end{pmatrix} \cdot \begin{pmatrix} \textit{fonction} \\ \textit{de l'erreur} \\ \textit{de prediction} \end{pmatrix} \quad (\text{A.3})$$

Le gain d'adaptation paramétrique est calculé suivant l'approche considérée.

A.1.1 Algorithme du gradient simple (ou de projection)

Considérons le modèle décrit par l'équation (A.2)

$$y(t) = \phi(t-1)^T \theta$$

Le critère quadratique à minimiser à chaque pas est:

$$J(t) = [y(t) - \phi^T(t-1)\hat{\theta}(t-1)]^2 \quad (\text{A.4})$$

Les estimés des paramètres seront calculés par:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \alpha \phi(t-1) \quad (\text{A.5})$$

où α est choisi de manière à avoir:

$$\begin{aligned} y(t) &= \phi^T(t-1)\hat{\theta}(t) \\ y(t) &= \phi^T(t-1)\hat{\theta}(t-1) + \alpha \phi^T(t-1)\phi(t-1) \end{aligned} \quad (\text{A.6})$$

Ce qui donne:

$$\alpha = \frac{1}{\phi^T(t-1)\phi(t-1)} [y(t) - \phi^T(t-1)\hat{\theta}(t-1)] \quad (\text{A.7})$$

La formule de calcul des paramètres θ sera alors:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\phi(t-1)}{\phi^T(t-1)\phi(t-1)} [y(t) - \phi^T(t-1)\hat{\theta}(t-1)] \quad (\text{A.8})$$

C'est l'algorithme de "Kaczmarz" [ÅST89]. Les données sont générées par l'équation (A.2).

Pour éviter le problème de la division par zéro, qui surgit quand $\phi(t)$ est nul, l'algorithme de projection est modifié dans la pratique de la manière suivante:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + a(t) \frac{\phi(t-1)[y(t) - \phi^T(t-1)\hat{\theta}(t-1)]}{c + \phi^T(t-1)\phi(t-1)} \quad (\text{A.9})$$

où $c > 0$ et $0 < a(t) < 2$.

$a(t)$ est une constante de gain calculée par:

$$\begin{aligned} a(t) &= 1 \text{ si } \theta_{n+1}, \text{ calculer avec } a(t)=1, \dots, 0 \\ a(t) &= \gamma \text{ sinon} \end{aligned}$$

avec $\varepsilon < \gamma < 2-\varepsilon$, $\gamma \neq 1$ et $0 < \varepsilon < 1$.

La convergence globale de cet algorithme a été démontrée dans [LAN93]. Cet algorithme a la propriété d'avoir un gain d'adaptation scalaire ne tendant pas vers zéro. Il est donc utilisé pour l'identification et la régulation adaptative de systèmes stationnaires ou à paramètres variables dans le temps. Il possède l'avantage de la simplicité, d'où une mise en oeuvre plus simple, mais sa convergence très lente. En pratique, il est utilisé pour les systèmes dont les paramètres varient lentement dans le temps.

A.1.2 Algorithme des moindres carrés récursif

La structure du modèle est celle donnée par l'équation (A.1).

De la même manière qu'au paragraphe précédent, nous utilisons le modèle sous la forme de régression est donné par l'équation (A.2)

$$y(t) = \phi^T(t-1)\theta$$

Le critère d'estimation va consister en la minimisation de l'erreur de sortie:

$$J_N(t) = \sum_{k=t-N}^t [y(t) - \phi^T(t-1) \hat{\theta}(t)]^2 \quad (\text{A.10})$$

N étant le nombre de mesures disponibles.

Pour trouver la solution récursive du problème on fait appel au lemme d'inversion matricielle. Les estimés satisfont les équations récursives:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)[y(t) - \phi^T(t-1)\hat{\theta}(t-1)] \\ K(t) &= \frac{p(t-1)\phi(t-1)}{1 + \phi^T(t-1)p(t-1)\phi(t-1)} \\ p(t) &= p(t-1) - \frac{p(t-1)\phi(t-1)\phi^T(t-1)p(t-1)}{1 + \phi^T(t-1)p(t-1)\phi(t-1)} \end{aligned} \quad (\text{A.11})$$

$P(t)$: Matrice de covariance.

$K(t)$: est le gain d'adaptation.

L'algorithme est initialisé par un choix adéquat de $p(0)$. On la prend souvent sous la forme $p(0) = \alpha I$. Où α est une grande valeur (10^4) si on ne dispose pas d'information initiale sur les paramètres. Ceci permet d'opérer un changement rapide sur les paramètres $\theta(t)$ initiaux. Par contre, si on dispose de l'estimation initiale des paramètres on prend α petit (≤ 1).

Dans le cas où le procédé est invariant dans le temps, la capacité d'adaptation est choisie décroissante et s'annulant exponentiellement de manière à ce que l'estimation reste insensible aux nouvelles données, quand les paramètres ont convergé vers leurs vraies valeurs.

En pratique, on préfère un algorithme qui poursuit les variations non brusques des paramètres. Une capacité d'adaptation plus importante doit être maintenue tout au long de l'estimation. Plusieurs techniques permettent d'y parvenir. Nous citons deux principales méthodes les plus utilisées :

- Facteur d'oubli variable avec trace constante.
- Gain décroissant avec trace constante.

Nous présentons, dans ce qui suit la première méthode.

Le gain d'adaptation est exprimé par:

$$\begin{aligned} K(t) &= \frac{p(t-1)\phi(t-1)}{\frac{\lambda_1(t)}{\lambda_2(t)} + \phi^T(t-1)p(t-1)\phi(t-1)} \\ p(t) &= [p(t-1) - K(t)\phi^T(t-1)p(t-1)] / \lambda_1(t) \end{aligned} \quad (\text{A.12})$$

$\lambda_1(t)$ et $\lambda_2(t)$ sont des facteurs introduits pour permettre de fixer un compromis entre la précision de l'estimation et la capacité de poursuite des paramètres dans le cas des systèmes variants dans le temps.

$\lambda_2(t)$ est fixé à 1, $\lambda_1(t)$ est calculé de manière à assurer:

$$\text{tr } P(t) = \text{tr } P(t-1) = \text{tr } P(0) = n.GI$$

n : nombre de paramètres.

GI : gain initial.

Dans ce cas P et K ne tendent pas vers zéro. Les valeurs typiques de $\lambda_l(0)$ sont de 0.95 pour les variations rapides à 0.99 pour les variation lentes. Nous pouvons interpréter $P(0)$ comme une précision de l'estimation initiale.

A.2 ESTIMATION DANS LE CAS STOCHASTIQUE

Dans le cas où le système est stochastique, l'estimation avec les moindres carrés devient biaiser. Une possibilité d'éviter ce problème, est de modéliser la corrélation de l'erreur et d'estimer les paramètres qui la décrivent [ÅST89].

Considérons le modèle:

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t) \quad (\text{A.13})$$

Où A , B et C sont des polynômes en q^{-1} .

$\{e(t)\}$ est un bruit blanc

Les paramètres du polynôme C décrivent la corrélation de la perturbation. Ce modèle ne peut pas être converti en un modèle de régression car $\{e(t)\}$ n'est pas connue. Cependant un modèle de régression peut être obtenu par une approximation convenable.

Le vecteur des paramètres sera:

$$\theta = [a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l]$$

et le vecteur des mesures

$$\phi(t) = [-y(t), \dots, -y(t-n+1); u(t), \dots, u(t-m+1); \varepsilon(t), \dots, \varepsilon(t-l+1)]^T$$

avec $\varepsilon(t) = y(t) - \phi^T(t-1)\theta(t-1)$

$e(t)$ est alors approximée par l'erreur de prédiction. Le modèle peut être approximé par :

$$y(t) = \phi^T(t-1)\theta \quad (\text{A.14})$$

et l'algorithme, donné par (A.12) est alors appliqué. La méthode obtenue est appelée: Moindres Carrées Etendue (MCE).

A.3 APPLICATION AU MODELE DE LA MSAP

Le modèle dynamique linéaire de la machine synchrone à aimants permanents obtenu après orientation du champ est donné par la fonction de transfert suivante

$$F_{BF}(s) = \frac{\omega_r}{v_q} = \frac{b'}{a_1' s^2 + a_2' s + 1} \quad (\text{A.15})$$

$$\text{Tels que } b' = \frac{P \cdot \phi_f}{R \cdot f + P^2 \cdot \phi_f^2}; \quad a_1' = \frac{L_q \cdot J}{R \cdot f + P^2 \cdot \phi_f^2}; \quad a_2' = \frac{L_q \cdot f + R \cdot J}{R \cdot f + P^2 \cdot \phi_f^2}$$

La fonction de transfert échantillonnée correspondante à la discrétisation de l'équation (A.15) est donnée par :

$$F(Z^{-1}) = \frac{z^{-d} \cdot (b_1 + b_2 z^{-1})}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (\text{A.16})$$

Les valeurs des paramètres de la fonction de transfert pour $d = 1$ et $T_e = 2ms$ sont données par le tableau suivant:

Tableau A.1 : Paramètres du modèle de la Machine

a_1	a_2	b_1	b_2
1.8471	0.9741	1.9613	1.9441

Afin de vérifier la validité de l'identification des paramètres par l'algorithme des moindres carrés récursifs, nous avons tracé l'évolution des paramètres a_i et b_i à la figure IV.1. Nous remarquons que les paramètres estimés convergent vers leurs valeurs réelles avec une erreur d'ordre 10^{-4} .

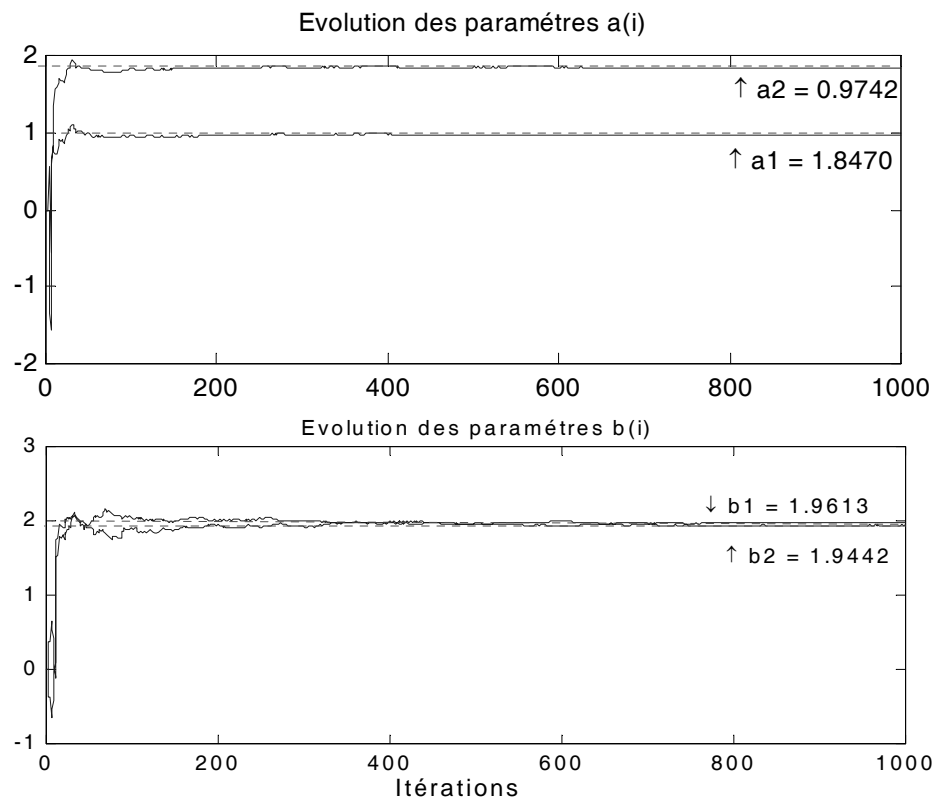


Figure A.1 Evolution des paramètres estimés.

ANNEXE B

FORME PREDICTIVE

Considérons, le système donné par le modèle ARMAX suivant:

$$A(q^{-1})y(t) = q^{-d} B(q^{-1})u(t) + C(q^{-1})e(t) \quad (\text{B.1})$$

En utilisant l'équation (III.12) dans (B.1) on obtient

$$A(q^{-1})P_D(q^{-1})\phi_y(t) = P_N(q^{-1})b(q^{-1})u(t-d) + P_N(q^{-1})C(q^{-1})e(t) \quad (\text{B.2})$$

Introduisons l'identité polynomiale suivante:

$$\frac{C(q^{-1})P_N(q^{-1})}{A(q^{-1})P_D(q^{-1})} = F(q^{-1}) + q^{-d} \frac{G(q^{-1})}{A(q^{-1})P_D(-1)} \quad (\text{B.3})$$

obtenue par la division du polynôme $P_N.C$ sur le polynôme $A.P_D$ jusqu'à l'ordre d .

$$F(q^{-1}) = f_0 + f_1 q^{-1} + \dots + f_{d-1} q^{-(d-1)}$$

$$G(q^{-1}) = g_0 + g_1 q^{-1} + \dots + g_{na-1} q^{-(na-1)}$$

En multipliant les deux membres de l'équation (B.2) par $F(q^{-1})$ on obtient:

$$\begin{aligned} F(q^{-1})A(q^{-1})P_D(q^{-1})\phi_y(t) &= F(q^{-1})P_N(q^{-1})B(q^{-1})u(t-d) \\ &+ F(q^{-1})P_N(q^{-1})C(q^{-1})e(t) \end{aligned} \quad (\text{B.4})$$

En introduisant l'identité (B.3) dans l'équation (B.4) on aboutit à :

$$\begin{aligned} [C(q^{-1})P_N(q^{-1}) - q^{-d}G(q^{-1})]\phi_y(t) &= F(q^{-1})P_N(q^{-1})B(q^{-1})u(t-d) \\ &+ F(q^{-1})P_N(q^{-1})C(q^{-1})e(t) \end{aligned} \quad (\text{B.5})$$

En réarrangeant cette équation on obtient:

$$\begin{aligned} C(q^{-1})P_N(q^{-1})[\phi_y(t) - F(q^{-1})e(t)] &= q^{-d}G(q^{-1})\phi_y(t) \\ &+ F(q^{-1})P_N(q^{-1})B(q^{-1})u(t) \end{aligned} \quad (\text{B.6})$$

$\phi_y(t+d)$ sera donné par :

$$\phi_y(t+d) = \frac{G(q^{-1})}{C(q^{-1})P_N(q^{-1})}\phi_y(t) + \frac{F(q^{-1})B(q^{-1})}{C(q^{-1})}u(t) + F(q^{-1})e(t+d) \quad (\text{B.7})$$

Utilisons (III.11), l'équation (B.7) devient:

$$\phi_y(t+d) = \frac{G(q^{-1})}{C(q^{-1})P_D(q^{-1})}y(t) + \frac{F(q^{-1})B(q^{-1})}{C(q^{-1})}u(t) + F(q^{-1})e(t+d) \quad (\text{B.8})$$

Le prédicteur est alors donné par:

$$\hat{\phi}_y(t+d) = \frac{G(q^{-1})}{C(q^{-1})P_D(q^{-1})}y(t) + \frac{F(q^{-1})B(q^{-1})}{C(q^{-1})}u(t) \quad (\text{B.9})$$

ANNEXE C

TRANSFORMATION DE PARK

La transformation de Park est utilisée pour supprimer la non linéarité des équations de la MSAP. Elle consiste à remplacer les enroulements de phases (a, b, c), dont les axes magnétiques sont immobiles par rapport au stator, par deux enroulements (d, q) dont les axes magnétiques sont solidaires au rotor et tournant avec lui avec une vitesse ω .

La matrice de Park qui assure la transformation du courant, flux et tension est:

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (C.1)$$

tel que la position du rotor est définie par :

$$\theta = \int_0^t \omega(\tau) d\tau \quad (C.2)$$

Nous appliquons donc un changement de variables défini par la matrice $P(\theta)$ au modèle électrique de la machine :

$$\begin{cases} [I_{dq}] = [P(\theta)][I_{abc}] \\ [V_{dq}] = [P(\theta)][V_{abc}] \\ [\psi_{dq}] = [P(\theta)][\psi_{abc}] \end{cases} \quad (C.3)$$

La matrice inverse de la matrice de Park $P(\theta)$ notée $P^{-1}(\theta)$, est donnée par l'équation suivante :

$$P^{-1}(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & -\sin\theta & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (C.4)$$

Les grandeurs (a, b, c) se calcules par:

$$\begin{cases} [I_{abc}] = [P^{-1}(\theta)][I_{dq}] \\ [V_{abc}] = [P^{-1}(\theta)][V_{dq}] \\ [\psi_{abc}] = [P^{-1}(\theta)][\psi_{dq}] \end{cases} \quad (C.5)$$

ANNEXE D
PARAMETRES DES MACHINES UTILISEES

Tableau D.1 : Paramètres de la Machine M1.

Puissance nominale	100 W
Tension nominale	28 V
Couple résistant	0.05 N.m
Nombre de paires de Pôles (P)	2
Inductance axe-d, L_d	12,1 mH
Inductance axe q, L_q	12,1 mH
Résistance statorique, R	3,4 Ω
Moment d'inertie, J_m	0.0001 N.m.s ² /rad
Coefficient de frottement, f_m	0.00005 N.m/rad/sec
Flux des aimants, ψ_f	0.013 volts/rad/sec

Tableau D.2 : Paramètres de la Machine M2.

Puissance nominale	2 kW
Vitesse nominale	4000 r/min
Nombre de paires de Pôles (P)	3
Inductance axe-d, L_d	3,2 mH
Inductance axe q, L_q	3,2 mH
Résistance statorique, R	1 Ω
Moment d'inertie, J_m	6 10^{-4} kg m ²
Coefficient de frottement, f_m	9,5 10^{-5} Nm s/rad
Flux des aimants, ψ_f	0,39 Nm/A