

**UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**COMME EXIGENCE PARTIELLE  
DE LA MAITRISE EN ELECTRONIQUE INDUSTRIELLE**

**PAR  
YOUNES EL BITTIOUI**

**ADAPTATION DE LA CONSTANTE DE TEMPS ROTORIQUE  
DU MOTEUR À INDUCTION**

**Février 1999**

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

*À mon père Abdellah et à ma mère Fatiha  
À mes sœurs Fatima Zohra et Ilham  
À mes Frères Abdelilah, Noureddine et le petit Nabil  
À titre de reconnaissance, d'amour et d'affection*

## RÉSUMÉ

La constante de temps rotorique joue un rôle primordial dans un grand nombre de commandes à flux orienté basées sur le modèle inverse de la machine à induction. En effet, au moindre écart entre la valeur utilisée dans le modèle théorique et la valeur réelle, il en résulte une désorientation du flux et par conséquent une détérioration des performances dynamiques. Plusieurs méthodes d'adaptation de cette constante de temps rotorique ont été décrites dans la littérature. Dans ce mémoire, nous proposerons une nouvelle approche que nous comparerons aux méthodes existantes. Cette nouvelle approche est basée sur le modèle dq dans le repère stationnaire d'une machine alimentée en tension. La simulation a été effectuée dans l'environnement Matlab®/Simulink et la validation expérimentale sur DSP TMS 320C est aussi prévue. Le recours aux réseaux neuronaux dans la mise en œuvre pratique sera également considéré.

## REMERCIEMENTS

Ce travail a été effectué au sein du laboratoire d'électronique de puissance à l'Université du Québec à Trois-Rivières, dirigé par le professeur Ahmed CHÉRITI.

Je désire, avant tout, exprimer ma profonde reconnaissance envers le professeur Ahmed CHÉRITI qui m'a proposé d'élaborer ce projet et m'a fait l'honneur d'en être l'encadreur.

Je tiens à lui témoigner toute ma gratitude pour l'aide efficace et précieuse qu'il m'a apportée et par la même occasion je rends hommage à sa rigueur scientifique et technique, à son dévouement et à la patience dont il a fait preuve à mon égard.

Je ne peux pas oublier d'adresser mes sincères remerciements à Mr A.Ba-razzouk pour sa sincère amitié, qui, grâce à son exemplarité, son sérieux et sa parfaite maîtrise de la profession, m'ont été toujours utiles à la réalisation et l'achèvement de ce travail.

Je remercie mon grand ami Mohamed Smyej pour les longues journées qu'on a passées ensemble à échanger nos idées. Ses suggestions et ses critiques m'ont beaucoup aidé à franchir tant de difficultés.

Je remercie tous mes amis de Trois-Rivières et dont ma mémoire n'a pas pu restituer le nom.

# TABLE DES MATIÈRES

RÉSUMÉ.....	III
REMERCIEMENT.....	IV
LISTE DES FIGURES.....	VIII
LISTE DES SYMBOLES.....	X
<b>Introduction .....</b>	1
<b>Chapitre 1 Modélisation de la machine à induction.....</b>	4
1.1 Introduction.....	4
1.2 Modèle mathématique de la machine à induction réelle .....	5
1.2.1 Transformation d'axes.....	9
1.2.2 Modèle dq de la machine dans le repère stationnaire lié au stator...	11
1.2.3 Modèle dq de la machine dans le repère synchrone lié au stator ...	12
1.2.4 Modèle dq de la machine dans le repère stationnaire lié au rotor...	13
1.3 Simulation du modèle dq dans le repère stationnaire lié au rotor.....	13
1.4 Conclusion.....	16
<b>Chapitre 2 Principes de la commande vectorielle à flux rotorique orienté.....</b>	18
2.1 Introduction.....	18
2.2 Principes de la commande vectorielle.....	20
2.2.1 Méthodes directe et indirecte d'orientation de flux.....	20
2.2.1.1 Technique d'orientation directe de flux.....	21
2.2.1.2 Technique d'orientation indirecte de flux.....	23
2.2.2.3 Théorie d'orientation indirecte du flux rotorique.....	24
2.2.2.4 Implantation du flux rotorique.....	27
2.3 Analyse de la commande vectorielle à orientation du flux rotorique..	29
2.4 Conclusion.....	34
<b>Chapitre 3 Influence de la constante de temps rotorique sur le contrôle vectoriel.....</b>	35
3.1 Introduction.....	35
3.2 Limitation de la méthode indirecte d'orientation de flux.....	36
3.3 Effet de la variation de la constante de temps rotorique sur la fréquence des courants.....	40
3.4 Modèle thermique de la machine.....	42

3.4.1 Rappel théorique.....	42
3.4.2 Étude expérimentale.....	44
3.5 Conclusion.....	45
<b>Chapitre 4 Théorie des réseaux de neurones.....</b>	<b>47</b>
4.1 Introduction.....	47
4.2 Fondements biologiques.....	48
4.2.1 Le cerveau.....	48
4.2.2 Le neurone.....	49
4.2.3 Structure du neurone.....	49
4.3 Le neurone formel.....	51
4.3.1 Modèle de Mac Culloch et Pitts.....	51
4.3.2 Le modèle généralisé.....	51
4.3.3 Fonction d'activation.....	53
4.4 Architectures des réseaux de neurones.....	54
4.4.1 Réseau à couche simple.....	55
4.4.2 Réseaux multicouches.....	56
4.4.3 Réseaux récurrents.....	57
4.4.4 Structure "Latice".....	58
4.5 Apprentissage des réseaux de neurones.....	58
4.5.1 Modes d'apprentissage.....	59
4.5.2 Procédure d'apprentissage.....	60
4.5.3 Problèmes d'apprentissage.....	61
4.6 Algorithme de rétropropagation.....	62
4.6.1 Formulation.....	63
4.6.2 Algorithme de Levenberg-Marquardt.....	64
4.7 Différentes classes des réseaux de neurones.....	65
4.7.1 Perceptron.....	65
4.7.2 Réseau linéaire.....	65
4.7.3 Perceptron multicouche.....	66
4.7.4 Réseau RBF.....	67
4.7.5 Réseaux récurrents.....	67
4.7.6 Réseaux compétitifs.....	68
4.8 Application des réseaux de neurones dans l'électronique de puissance.....	69
4.8.1 Régulateur statique.....	69
4.8.2 Régulateur dynamique.....	70
4.8.3 Commande des machines électriques.....	70

4.9 Conclusion.....	71
<b>Chapitre 5 Adaptation de la constante de temps rotorique dans la commande vectorielle.....</b>	<b>72</b>
5.1 Introduction.....	72
5.2 Détermination de la constante de temps rotorique réelle du moteur....	73
5.3 Adaptation de la constante de temps rotorique par les réseaux de neurones.....	75
5.3.1 Choix du réseau de neurones.....	76
5.3.2 Apprentissage du réseau.....	76
5.3.2.1 Préparation de données.....	77
5.3.2.2 Choix du nombre de neurones.....	77
5.3.2.3 Choix des paramètres de l'algorithme d'apprentissage.....	77
5.3.2.4 Procédure d'apprentissage.....	78
5.3.3 Réseau estimateur de la constante de temps rotorique.....	78
5.4 Résultats de simulation.....	80
5.4.1 Variation de $\tau_r$ par échelon.....	80
5.4.2 Variation exponentielle de $\tau_r$ .....	83
5.5 Estimateur basé sur le filtre de Kalman.....	85
5.6 Conclusion.....	88
<b>Chapitre 6 Implantation pratique de la méthode sur DSP TMS320C31.....</b>	<b>89</b>
6.1 Introduction.....	89
6.2 Considérations pratiques.....	90
6.3 Implantation sur DSP TMS320C31.....	90
6.4 Résultats pratiques.....	93
6.4.1 Effet pratique de $\tau_r$ .....	93
6.4.1 Efficacité de la méthode d'adaptation.....	95
6.5 Conclusion.....	96
<b>Conclusion.....</b>	<b>97</b>
<b>Bibliographie.....</b>	<b>98</b>
<b>Annexe A : Programme d'apprentissage du réseau de neurones.....</b>	<b>101</b>
<b>Annexe B : Programme du contrôle vectoriel sans adaptation.....</b>	<b>103</b>
<b>Annexe C : Programme du contrôle vectoriel avec adaptation.....</b>	<b>107</b>

# LISTE DES FIGURES

## Chapitre 1

Figure 1.1. Représentation des enroulements de la machine asynchrone...	5
Figure 1.2. Transformation d'axes (3phases)-(dq).....	8
Figure 1.3. Organigramme d'implantation du modèle dq.....	14
Figure 1.4. Entraînement direct du modèle Simulink de la machine à induction.....	15
Figure 1.5. Réponses d'un moteur à induction couplé directement au réseau.....	16

## Chapitre 2

Figure 2.1. Machine à courant continu.....	19
Figure 2.2. Contrôle vectoriel direct.....	21
Figure 2.3. Systèmes d'axes dq.....	23
Figure 2.4. Modèle découplé de la machine à induction dans le cas d'une orientation du flux rotorique.....	26
Figure 2.5. Structure d'un contrôleur du flux rotorique.....	26
Figure 2.6. Schéma d'implantation d'un régulateur à orientation du flux rotorique .....	28
Figure 2.7. Transformation des consignes courant en tension.....	29
Figure 2.8. Modèle Simulink du contrôle vectoriel d'une machine à induction alimentée en tension .....	30
Figure 2.9. Réponse d'une commande scalaire .....	31
Figure 2.10. Flux rotatoire et sa référence .....	31
Figure 2.11. Couple et sa référence pour un flux constant.....	32
Figure 2.12. Flux rotatoire et sa consigne pour un couple constant .....	33
Figure 2.13. Couple et sa référence pour un flux variable .....	33

## Chapitre 3

Figure 3.1. Variation de la constante de temps rotatoire .....	37
Figure 3.2. Effet de la variation de la constante de temps rotatoire sur le flux .....	37
Figure 3.3. Effet de la variation de la constante de temps rotatoire sur le couple .....	38
Figure 3.4. Variation du couple en fonction de la constante du temps rotatoire .....	39
Figure 3.5. Variation du flux en fonction de la constante de temps rotatoire .....	40

Figure 3.6. Variation de la fréquence des courants statoriques en fonction de la constante de temps rotorique.....	42
Figure 3.7. Élévation de la température rotorique.....	44
Figure 3.8. Effet de la température sur la résistance rotorique.....	45
<b>Chapitre 4</b>	
Figure 4.1. Un neurone biologique simple.....	50
Figure 4.2. Formalisation mathématique d'un neurone.....	51
Figure 4.3. Fonction d'activation .....	54
Figure 4.4. Réseau à couche simple.....	56
Figure 4.5. Réseau à deux couches.....	57
Figure 4.6. Réseau récurrent.....	58
Figure 4.7. "Latice " à deux dimensions (3 neurones par 2).....	58
Figure 4.8. Évolution typique de l'erreur quadratique moyenne.....	61
<b>Chapitre 5</b>	
Figure 5.1. Principe d'adaptation de la constante de temps rotorique.....	75
Figure 5.2. Réseau de neurones en blocs Simulink.....	76
Figure 5.3. Réseau de neurones estimateur de la constante de temps rotorique.....	79
Figure 5.4. Résultat d'apprentissage.....	80
Figure 5.5. Variation programmée de la constante de temps rotorique ....	81
Figure 5.6. Modèle Simulink du contrôle vectoriel avec adaptation de $\tau_r$ ..	81
Figure 5.7. Réponse du flux avec adaptation.....	82
Figure 5.8. Réponse du couple avec adaptation.....	83
Figure 5.9. Variation exponentielle de la constante de temps rotorique....	84
Figure 5.10. — Flux rotorique sans adaptation ---- Flux rotorique avec adaptation.....	84
Figure 5.11. — Couple sans adaptation ----Couple avec adaptation.....	85
Figure 5.12. Sensibilité de l'estimateur basé sur le filtre de Kalman.....	88
<b>Chapitre 6</b>	
Figure 6.1. Montage pratique.....	91
Figure 6.2. Schéma de réalisation sur DSP.....	92
Figure 6.3. Niveau du flux dans la machine.....	93
Figure 6.4. Réponse du couple.....	94
Figure 6.5. Niveau du flux dans la machine pour 25% une réduction de $\tau_r$ de 25%.....	94
Figure 6.6. Réponse du couple pour pour 25% une réduction de $\tau_r$ de 25%.....	95

## LISTE DES SYMBOLES

$\sigma$	Coefficient de fuite total. $\sigma = \frac{(L_s L_r - M)^2}{L_s L_r}$
$\alpha$	Coefficient thermique de résistivité
$\omega_e^*$	Consigne de la fréquence des courants d'alimentation
$\tau_r^*$	Constante de temps rotorique nominale
$\Psi_r^*$	Consigne du flux rotorique
$\omega_{sl}^*$	Consigne de la fréquence de glissement
$\Psi_{dr}$	Composante directe du flux rotorique
$\Psi_{ds}$	Composante directe du flux statorique
$\omega_e$	Fréquence des courants d'alimentation
$\omega_{mec}$	Vitesse mécanique mesurée directement sur l'arbre
$\Psi_{qr}$	Composante en quadrature du flux rotorique
$\Psi_{qs}$	Composante en quadrature du flux statorique
$\tau_r$	Constante de temps rotorique
$\Psi_r$	Flux rotorique
$\omega_{sl}$	Fréquence de glissement
$[P(\theta)]$	Matrice de transformation 3 phases -- dq
$D$	Coefficient de frottement visqueux
$i_{ds}^*$	Consigne directe du courant statorique
$i_{qs}^*$	Consigne en quadrature du courant statorique
$i_{ar}$	Courant dans la phase a du rotor
$i_{as}$	Courant dans la phase a du stator
$i_{br}$	Courant dans la phase b du rotor
$i_{bs}$	Courant dans la phase b du stator
$i_{cr}$	Courant dans la phase c du rotor
$i_{cs}$	Courant dans la phase c du stator
$i_{dr}$	Composante directe du courant rotorique
$i_{ds}$	Composante directe du courant statorique
$i_{mr}$	Courant rotorique magnétisant
$i_{qr}$	Composante en quadrature du courant rotorique
$i_{qs}$	Composante en quadrature du courant statorique
$i_s$	Courant statorique

$J$	Moment d'inertie (charge et machine)
$L_r$	Inductance cyclique d'un enroulement rotorique
$L_{rr}$	Inductance propre d'un enroulement rotorique
$L_s$	Inductance cyclique d'un enroulement statorique
$L_{ss}$	Inductance propre d'un enroulement statorique
$M$	Inductance mutuelle entre les enroulements statoriques et rotoriques
$pp$	Nombre de paires de pôles
$R_r^*$	Résistance nominale d'un enroulement rotorique
$R_r$	Résistance d'un enroulement rotorique
$R_s$	Résistance d'un enroulement statorique
$T_{em}^*$	Consigne du couple électromagnétique
$T_{em}$	Couple électromagnétique
$T_L$	Couple de charge
$V_{as}$	Tension de la phase a du stator
$V_{bs}$	Tension de la phase b du stator
$V_{cs}$	Tension de la phase c du stator
$V_{ds}$	Composante directe de la tension du stator
$V_{qs}$	Composante en quadrature de la tension du stator
$[W_{ij}]$	Matrice des poids
$[b]$	Matrice des seuils

# Introduction

Durant ces dernières années plusieurs techniques de commande ont été développées pour permettre au variateur asynchrone d'atteindre des performances (vitesse, découplage...) élevées. Le contrôle vectoriel, qui permet un découplage entre les variables de commande, reste le plus utilisé vu les performances dynamiques élevées qu'il offre pour une large gamme d'applications. Pour surmonter la sensibilité de cette stratégie aux variations des paramètres du système et aux perturbations externes, des techniques modernes telles que le contrôle robuste, le contrôle adaptatif ainsi que l'estimateur à base du filtre de KALMAN ont été appliquées[1].

Dans la commande à flux orienté, la constante de temps rotorique est un paramètre non négligeable et les performances de cette commande dépendent considérablement de ce paramètre. Considérons par exemple une commande largement répandue qui est celle de l'orientation du flux rotorique. Cette commande se sert du modèle inverse de la machine pour déduire le flux et la fréquence de glissement. Or ces deux grandeurs dépendent de la constante de temps rotorique et le moindre écart entre les valeurs réelles et celles utilisées dans le modèle résulte en une perte d'orientation. D'où l'importance d'estimer et d'adapter la constante de temps rotorique en temps réel.

Le chapitre 1 présente une modélisation mathématique de la machine à induction dans le repère  $dq$  qui réduit l'étude de la machine à deux axes. Cette étude sera effectuée dans différents repères : stationnaires, synchrones liés au

stator ou au rotor. Une validation du modèle dq dans le repère stationnaire lié au rotor a été effectuée dans l'environnement Matlab®/Simulink.

Le chapitre 2 présente les principes de la commande vectorielle à flux rotorique orienté pour les deux méthodes directe et indirecte. L'étude, l'implantation et l'analyse de cette commande sont présentées et comparées à une commande scalaire.

Le chapitre 3 présente l'effet de la variation de la constante de temps rotorique sur les performances de la commande vectorielle à flux rotorique orienté. Il sera donc consacré à l'étude des limitations techniques dans le but de mieux comprendre le comportement de cet entraînement dans les conditions de fonctionnement plus réel.

Le chapitre 4 présente les généralités sur les réseaux de neurones artificiels (RNA). On commence par une présentation du fondement biologique qui est la base des RNA, puis une formulation mathématique d'un neurone, ainsi que l'architecture et les types de réseaux, en suite une brève présentation de l'algorithme de rétropropagation et à la fin quelques applications des RNA dans l'électronique de puissance sont présentées.

Le chapitre 5 présente une méthode d'adaptation de la constante de temps rotorique dans la commande vectorielle. Cette méthode utilise un réseau de neurones comme estimateur. Les résultats de cette adaptation sont comparés avec ceux d'une adaptation utilisant le filtre de Kalman.

Le chapitre 6 présente une implantation pratique sur DSP TMS320C31.  
Cette implantation nous permet de valider les résultats de simulation.

# *Chapitre 1*

## *Modélisation de la machine à induction*

### **1.1 Introduction**

Malgré qu'elles soient le plus souvent utilisées pour convertir l'énergie électrique en énergie mécanique, les machines à induction peuvent fonctionner aussi bien en moteur qu'en générateur. Dans le domaine des entraînements à vitesse variable, ces machines sont fréquemment utilisées dans les applications de grandes puissances.

La machine à induction est constituée d'un stator et d'un rotor. Dans le cas de la machine asynchrone à rotor bobiné, les enroulements rotoriques sont accessibles pour y connecter des résistances ou des convertisseurs statiques (cascade hyposynchrone). Tandis que dans le cas de la machine asynchrone à cage, les enroulements rotoriques sont en court-circuit sur eux mêmes. La machine asynchrone à cage est la plus utilisée dans le domaine des entraînements.

Lorsque le rotor de la machine est stationnaire, l'alimentation des enroulements statoriques par des sources de tensions triphasées équilibrées induit des courants dans les enroulements rotoriques à la même fréquence. L'interaction entre le flux produit dans l'entrefer et la force magnétomotrice créée au rotor, donne naissance au couple dans la machine. À la vitesse de synchronisme (c'est-à-dire à glissement nul), le rotor ne peut avoir aucune induction et par conséquent aucun couple n'est produit. En charge, la vitesse de la machine est différente de la vitesse de synchronisme, le flux d'entrefer tourne à la fréquence du glissement par rapport au rotor, ce qui contribue à induire des courants dans ce dernier et produit un couple électromagnétique pour compenser le couple imposé par la charge.

## 1.2 Modèle mathématique de la machine à induction réelle

La machine asynchrone est représentée à la figure 1.1 par six enroulements dans l'espace électrique; l'angle  $\alpha$  repère l'axe de la phase rotorique de référence  $R_a$  par rapport à l'axe de la phase statorique de référence  $S_a$

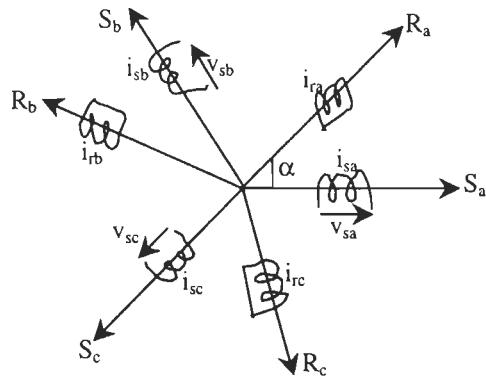


Figure 1.1. Représentation des enroulements de la machine asynchrone.

Les équations mathématiques décrivant la dynamique des six phases couplées de la machine sont [2] :

$$\frac{d}{dt} \begin{bmatrix} [L_{ss}] & [L_{sr}] \\ [L_{rs}] & [L_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} = \begin{bmatrix} [v_s] \\ [v_r] \end{bmatrix} - \begin{bmatrix} [R_{ss}] & [0] \\ [0] & [R_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} \quad (1-1)$$

$$T_{em} = \frac{1}{2} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix}^t \frac{\partial}{\partial \theta_r} \begin{bmatrix} [L_{sr}] & [L_{ss}] \\ [L_{rs}] & [L_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} \quad (1-2)$$

$$\begin{bmatrix} R_{ss} \\ R_{rr} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \quad \begin{bmatrix} L_{ss} \\ L_{rr} \end{bmatrix} = \begin{bmatrix} L_{AA} & L_{AB} & L_{AC} \\ L_{BA} & L_{BB} & L_{BC} \\ L_{CA} & L_{CB} & L_{CC} \end{bmatrix} \quad (1-3)$$

$$\begin{bmatrix} L_{sr} \\ L_{rs} \end{bmatrix} = \begin{bmatrix} L_{Aa} \cos(\theta_r) & L_{Ab} \cos(\theta_r + \phi) & L_{Ac} \cos(\theta_r - \phi) \\ L_{Ba} \cos(\theta_r - \phi) & L_{Bb} \cos(\theta_r) & L_{Bc} \cos(\theta_r + \phi) \\ L_{Ca} \cos(\theta_r + \phi) & L_{Cb} \cos(\theta_r - \phi) & L_{Cc} \cos(\theta_r) \end{bmatrix} \quad (1-4)$$

$$\begin{bmatrix} v_s \\ v_r \end{bmatrix} = \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix}, \quad \begin{bmatrix} i_s \\ i_r \end{bmatrix} = \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix}, \quad \begin{bmatrix} i_s \\ i_r \end{bmatrix} = \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}, \quad \begin{bmatrix} i_s \\ i_r \end{bmatrix} = \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad \text{et} \quad \phi = \frac{2\pi}{3} \quad (1-5)$$

Les lettres majuscules correspondent aux grandeurs statoriques. Les lettres minuscules correspondent aux grandeurs rotoriques.

Ces équations sont très générales et peuvent être considérablement réduites, si on introduit les hypothèses simplificatrices usuelles :

1. Les paramètres électriques de la machine ne dépendent ni du temps ni des conditions de fonctionnement;
2. La réciprocité des inductances mutuelles;
3. Les enroulements sont identiques du côté stator et du côté rotor;
4. La machine est alimentée par des sources parfaitement équilibrées;
5. Le neutre des enroulements est indépendant du neutre des alimentations.

Avec ces considérations, on aboutit au modèle mathématique simplifié suivant:

$$\frac{d}{dt} \begin{bmatrix} [L_{ss}] & [L_{sr}] \\ [L_{rs}] & [L_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} = \begin{bmatrix} [v_s] \\ [v_r] \end{bmatrix} - \begin{bmatrix} [R_{ss}] & [0] \\ [0] & [R_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} \quad (1-6)$$

$$[R_{ss}] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \quad ; \quad [R_{rr}] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}$$

L'hypothèse de la réciprocité des inductances mutuelles donne:

$$[L_{ss}] = \begin{bmatrix} L_{AA} & 0 & 0 \\ 0 & L_{BB} & 0 \\ 0 & 0 & L_{CC} \end{bmatrix} \quad ; \quad [L_{rr}] = \begin{bmatrix} L_{aa} & 0 & 0 \\ 0 & L_{bb} & 0 \\ 0 & 0 & L_{cc} \end{bmatrix} \quad (1-7)$$

$$[L_{sr}] = \frac{2M}{3} \begin{bmatrix} \cos(\theta_r) & \cos(\theta_r + \phi) & \cos(\theta_r - \phi) \\ \cos(\theta_r - \phi) & \cos(\theta_r) & \cos(\theta_r + \phi) \\ \cos(\theta_r + \phi) & \cos(\theta_r - \phi) & \cos(\theta_r) \end{bmatrix} \quad (1-8)$$

$$[L_{rs}] = [L_{sr}]^T \quad (1-9)$$

On note que même avec ce modèle simplifié, la résolution du système reste complexe à cause des coefficients dépendants du temps dans les matrices d'état et du couplage qui existe entre les phases de la machine. La formulation dans l'espace d'état d'un tel système nécessite l'inversion de la matrice  $[L_{sr}]$ .

Malheureusement celle-ci donne des termes en  $[\cos(\theta_r)]^{-1}$  qui créent des problèmes de calcul lorsque  $\cos(\theta_r)$  devient nul.

Le but est donc de chercher un modèle dont la matrice  $[L_{sr}]$  ne dépend pas explicitement de l'angle  $\theta_r$ , et dont l'inversion par le moyen de transformation mathématique ne donnera pas le problème mentionné. Ces transformations devront assurer l'invariance de puissance.

La théorie unifiée des machines électriques classiques, dite encore théorie généralisée, est basée sur la transformation de Park [3] qui rapporte les équations électriques statoriques et rotoriques à des axes perpendiculaires électriquement appelés d, pour direct, et q, pour quadrature comme sur la figure 1.2.

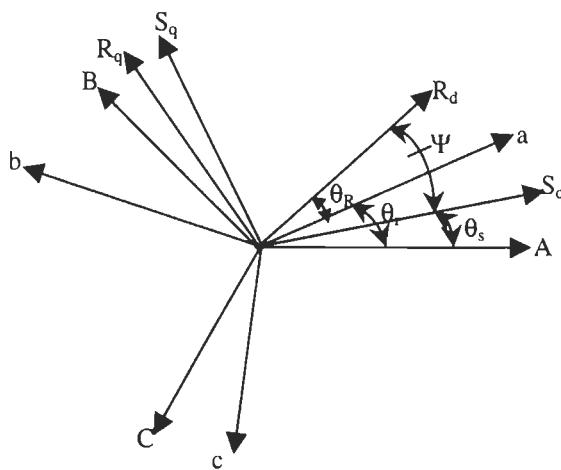


Figure 1.2. Transformation d'axes (3phases)-(dq).

- (A, B, C): Système triphasé statorique;
- (a, b, c) : Système triphasé rotorique,
- ( $S_d$ ,  $S_q$ ) : Système diphasé statorique;
- ( $R_d$ ,  $R_q$ ) : Système diphasé rotorique.

### 1.2.1 Transformation d'axes

Les transformations d'axes permettent non seulement de rendre les matrices d'état indépendantes de la position rotorique, mais aussi de simplifier l'analyse de la machine en considérant, après quelques simplifications, une machine biphasée équivalente au lieu de la machine triphasée originale.

L'une des transformations les plus utilisées est la transformation de Park ou le modèle dq généralisé, basée sur la matrice de transformation orthogonale suivante [3] :

$$[P(\theta)] = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos(\theta - \phi) & \cos(\theta + \phi) \\ -\sin(\theta) & -\sin(\theta - \phi) & -\sin(\theta + \phi) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (1-10)$$

Cette transformation permet d'obtenir les composantes d'qo des grandeurs triphasées, tout en assurant l'invariance de puissance suite à la transformation.

La composante homopolaire "o" n'intervient pas dans le cas de systèmes équilibrés car elle est nulle.

La formulation d'état relative à cette transformation, en supposant les systèmes équilibrés, est :

$$\begin{bmatrix} [v_s] \\ [v_r] \end{bmatrix} = \begin{bmatrix} [R_{ss}] & [0] \\ [0] & [R_{rr}] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} + \frac{d}{dt} [A] \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} + [\Omega] [A] \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} \quad (1-11)$$

où :

$$[v_s] = \begin{bmatrix} [v_{ds}] \\ [v_{qs}] \end{bmatrix}; [v_r] = \begin{bmatrix} [v_{dr}] \\ [v_{qr}] \end{bmatrix}; [R_{ss}] = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix}; [R_{rr}] = \begin{bmatrix} R_r & 0 \\ 0 & R_r \end{bmatrix} \quad (1-12)$$

$$[\Omega] = \begin{bmatrix} 0 & -\omega_s & 0 & 0 \\ \omega_s & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_R \\ 0 & 0 & \omega_R & 0 \end{bmatrix} \quad (1-13)$$

$$[A] = \begin{bmatrix} (L_{ss} - M_{ss}) & 0 & M\cos(\Psi) & -M\sin(\Psi) \\ 0 & (L_{ss} - M_{ss}) & M\sin(\Psi) & M\cos(\Psi) \\ M\cos(\Psi) & M\sin(\Psi) & (L_{rr} - M_{rr}) & 0 \\ -M\sin(\Psi) & M\cos(\Psi) & 0 & (L_{rr} - M_{rr}) \end{bmatrix} \quad (1-14)$$

$L_{ss}$  : inductance propre de l'enroulement statorique

$$M_{ss} = -\frac{L_{ss}}{2} : \text{inductance mutuelle de l'enroulement statorique} \quad (1-15)$$

$L_{rr}$  : inductance propre de l'enroulement rotorique

$$M_{rr} = -\frac{L_{rr}}{2} : \text{inductance mutuelle de l'enroulement rotorique} \quad (1-16)$$

$M$  : inductance mutuelle entre les enroulements statoriques et rotoriques

$$\text{avec: } \Psi = (\theta_r - \theta_s + \theta_R) \quad \text{et} \quad \frac{d\Psi}{dt} = (\omega_r - \omega_s + \omega_R) \quad (1-17)$$

$$\omega_r = \frac{d\theta_r}{dt}; \quad \omega_s = \frac{d\theta_s}{dt} \quad \text{et} \quad \omega_R = \frac{d\theta_R}{dt} \quad (1-18)$$

Le couple est donné par [6]:

$$T_{em} = \frac{3}{2} pp.M[i_{qs}i_{dr}\cos(\Psi) - i_{qs}i_{dr}\sin(\Psi) - i_{ds}i_{dr}\sin(\Psi) - i_{ds}i_{qr}\cos(\Psi)] \quad (1-19)$$

où  $pp$  est le nombre de paires de pôles de la machine.

Dans le cas où l'angle  $\Psi = 0$ , les axes dq du stator se superposent avec les axes dq du rotor et tournent à la même vitesse, alors on considère un seul système d'axes dq [4].

### 1.2.2 Modèle dq de la machine dans le repère stationnaire lié au stator

C'est le cas particulier où  $\Psi = 0$ , dans lequel le système d'axes dq est immobile par rapport au stator, l'axe "d" coïncide avec la phase A du stator (figure 1.2). Dans ce cas, on a:  $\theta_s = 0$  et  $\omega_s = 0$ , et comme  $\Psi = 0$  on a donc :

$$\theta_r = -\theta_R \text{ et } \omega_r = -\omega_R .$$

$$\begin{aligned} v_{ds} &= R_s i_{ds} + \frac{d}{dt} [(L_{ss} - M_{ss}) i_{ds} + M i_{dr}] \\ v_{qs} &= R_s i_{qs} + \frac{d}{dt} [(L_{ss} - M_{ss}) i_{qs} + M i_{qr}] \end{aligned} \quad (1-20)$$

$$\begin{aligned} v_{dr} &= R_r i_{dr} + \frac{d}{dt} [(L_{rr} - M_{rr}) i_{dr} + M i_{ds}] - \omega_r (L_{rr} - M_{rr}) i_{qr} - \omega_r M i_{qs} \\ v_{qr} &= R_r i_{qr} + \frac{d}{dt} [(L_{rr} - M_{rr}) i_{qr} + M i_{qs}] + \omega_r (L_{rr} - M_{rr}) i_{dr} + \omega_r M i_{ds} \end{aligned} \quad (1-21)$$

En posant:

$$(L_{ss} - M_{ss}) = \frac{3L_{ss}}{2} = L_s \quad \text{et} \quad (L_{rr} - M_{rr}) = \frac{3L_{rr}}{2} = L_r \quad (1-22)$$

où  $L_s$  et  $L_r$  sont respectivement les inductances cycliques des enroulements statoriques et rotoriques, on trouve:

$$\begin{aligned} v_{ds} &= R_s i_{ds} + \frac{d}{dt} [L_s i_{ds} + M i_{dr}] \\ v_{qs} &= R_s i_{qs} + \frac{d}{dt} [L_s i_{qs} + M i_{qr}] \end{aligned} \quad (1-23)$$

$$\begin{aligned}
v_{dr} &= R_r i_{dr} + \frac{d}{dt} [L_r i_{dr} + M i_{ds}] - \omega_r L_r i_{qr} - \omega_r M i_{qs} \\
v_{qr} &= R_r i_{qr} + \frac{d}{dt} [L_r i_{qr} + M i_{qs}] + \omega_r L_r i_{dr} + \omega_r M i_{ds}
\end{aligned} \tag{1-24}$$

Les équations des flux sont :

$$\begin{aligned}
\Psi_{ds} &= L_s i_{ds} + M i_{dr} \\
\Psi_{qs} &= L_s i_{qs} + M i_{qr} \\
\Psi_{dr} &= L_r i_{dr} + M i_{ds} \\
\Psi_{qr} &= L_r i_{qr} + M i_{qs}
\end{aligned} \tag{1-25}$$

et le couple est donné par :

$$T_{em} = \frac{3}{2} pp.M [i_{qs} i_{dr} - i_{ds} i_{qr}] \tag{1-26}$$

### 1.2.3 Modèle dq de la machine dans le repère synchrone lié au stator

C'est un cas particulier de  $\Psi = 0$ , dans lequel le système d'axes dq tourne à la même vitesse angulaire que le champs tournant (figure 1.2). Dans ce cas :  $\theta_s = \theta_e = \omega_e t$ ,  $\omega_s = \omega_e$ ,  $\theta_R = g\theta_e$  où  $g$  est le glissement et  $\theta_r = \theta_e - \theta_R = \theta_s(1-g)$ .

Les équations de la machine (1-23) et (1-24) deviennent :

$$\begin{aligned}
v_{ds} &= R_s i_{ds} + \frac{d}{dt} [L_s i_{ds} + M i_{dr}] - \omega_e L_s i_{qs} - \omega_e M i_{qr} \\
v_{qs} &= R_s i_{qs} + \frac{d}{dt} [L_s i_{qs} + M i_{qr}] + \omega_e L_s i_{ds} + \omega_e M i_{dr}
\end{aligned} \tag{1-27}$$

$$v_{dr} = R_r i_{dr} + \frac{d}{dt} [L_r i_{dr} + M i_{ds}] - (\omega_e - \omega_r) L_r i_{qr} - (\omega_e - \omega_r) M i_{qs} \tag{1-28}$$

$$v_{qr} = R_r i_{qr} + \frac{d}{dt} [L_r i_{qr} + M i_{qs}] + (\omega_e - \omega_r) L_r i_{dr} + (\omega_e - \omega_r) M i_{ds} \tag{1-28}$$

Les équations des flux (1-25) et du couple (1.26) ne changent pas.

#### 1.2.4 Modèle dq de la machine dans le repère stationnaire lié au rotor

C'est le cas particulier de  $\Psi = 0$  (figure 1.2), dans lequel le système d'axes dq est fixe par rapport au rotor. Dans ce cas:  $\theta_R = 0$ ,  $\omega_R = 0$ ,  $\omega_s = \omega_r$  et  $\theta_r = \theta_s$ .

Les équations de la machine (1-23) et (1-24) deviennent :

$$\begin{aligned} v_{ds} &= R_s i_{ds} + \frac{d}{dt} [L_s i_{ds} + M i_{dr}] - \omega_r L_s i_{qs} - \omega_r M i_{qr} \\ v_{qs} &= R_s i_{qs} + \frac{d}{dt} [L_s i_{qs} + M i_{qr}] + \omega_r L_s i_{ds} + \omega_r M i_{dr} \end{aligned} \quad (1-29)$$

$$\begin{aligned} v_{dr} &= R_r i_{dr} + \frac{d}{dt} [L_r i_{dr} + M i_{ds}] \\ v_{qr} &= R_r i_{qr} + \frac{d}{dt} [L_r i_{qr} + M i_{qs}] \end{aligned} \quad (1-30)$$

Les flux et le couple sont toujours donnés par les équations (1-25) et (1-26).

#### 1.3 Simulation du modèle dq dans le repère stationnaire lié au stator

La première application, désormais classique, du modèle de Park est sa mise en œuvre dans la simulation numérique du démarrage direct de la machine asynchrone alimenté par un réseau triphasé équilibré de tensions sinusoïdales et de fréquence fixe. Toutes les équations différentielles du modèle du moteur à induction dans le repère stationnaire lié au stator, nécessaires à l'étude du mouvement, peuvent être adaptées à l'intégration numérique. En manipulant les équations (1-23)-(1-25) et en utilisant les flux comme variables d'état on aura :

$$\begin{aligned}
 \frac{d\Psi_{ds}}{dt} &= v_{ds} - \frac{1}{\sigma\tau_s}(\Psi_{ds} - \frac{M}{L_r}\Psi_{dr}) \\
 \frac{d\Psi_{qs}}{dt} &= v_{qs} - \frac{1}{\sigma\tau_s}(\Psi_{qs} - \frac{M}{L_r}\Psi_{qr}) \\
 \frac{d\Psi_{dr}}{dt} &= -\frac{1}{\sigma\tau_r}(\Psi_{dr} - \frac{M}{L_s}\Psi_{ds}) - \omega_r\Psi_{qr} \\
 \frac{d\Psi_{qr}}{dt} &= -\frac{1}{\sigma\tau_r}(\Psi_{qr} - \frac{M}{L_s}\Psi_{qs}) + \omega_r\Psi_{dr}
 \end{aligned} \tag{1-31}$$

L'équation du couple :

$$T_{em} = \frac{3}{2}pp.M[i_{qs}i_{dr} - i_{ds}i_{qr}] \tag{1-32}$$

L'équation fondamentale de la mécanique :

$$J \frac{d\Omega}{dt} = T_{em} - T_L - D.\Omega \tag{1-33}$$

La figure 1.2 décrit le modèle dq implanté dans l'environnement Matlab<sup>®</sup>/Simulink.

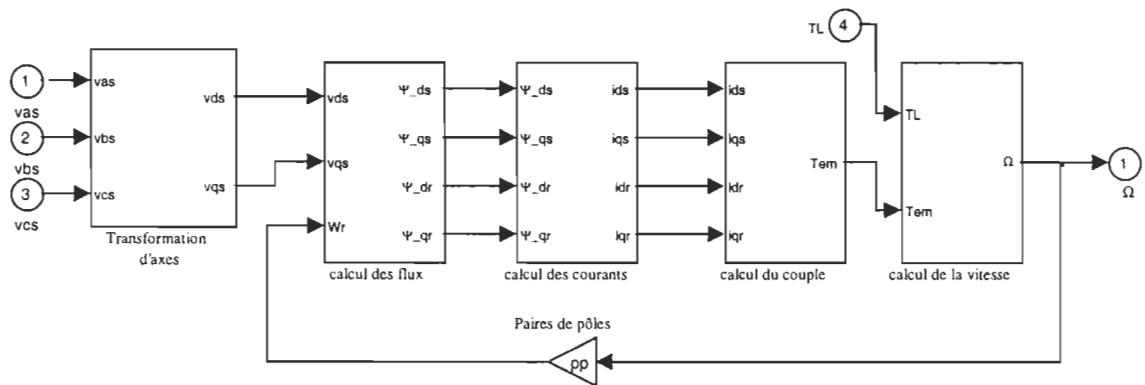


Figure 1.3. Organigramme d'implantation du modèle dq

Les transformations de deux phases à trois phases ou vice versa sont données par [3] :

$$\begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} \quad (1-34)$$

$$\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} \quad (1-35)$$

$$\begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ \cos(\theta_r + \frac{2\pi}{3}) & \sin(\theta_r + \frac{2\pi}{3}) \\ \cos(\theta_r - \frac{2\pi}{3}) & \sin(\theta_r - \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} i_{dr} \\ i_{qr} \end{bmatrix} \quad (1-36)$$

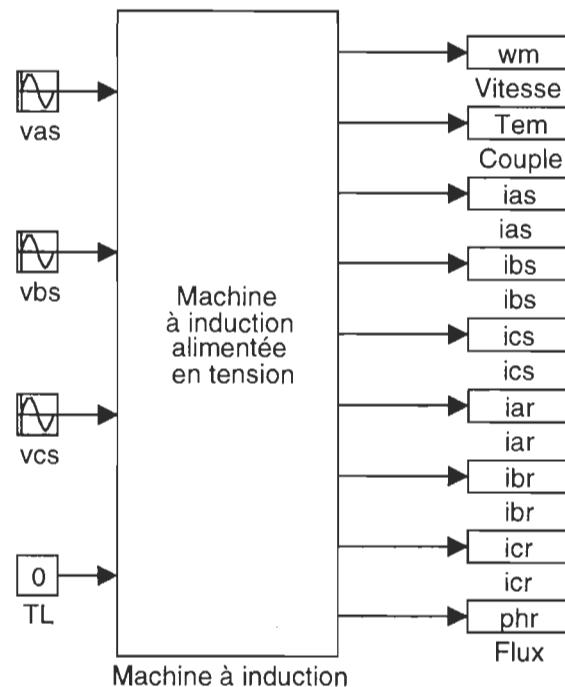


Figure 1.4. Modèle Simulink de l' entraînement direct de la machine à induction

Lors du démarrage à vide (figure 1.3), on constate d' emblée l' importance des courants statoriques et rotoriques pouvant être à l' origine de la destruction de

la machine par suréchauffement en cas de répétitions excessives. Pendant le régime transitoire, le couple est fortement pulsatoire ce qui explique le bruit engendré par la partie mécanique. Les résultats présentés sur la figure 1.5 confirment la validité du modèle qui peut être appliqué dans d'autres applications.

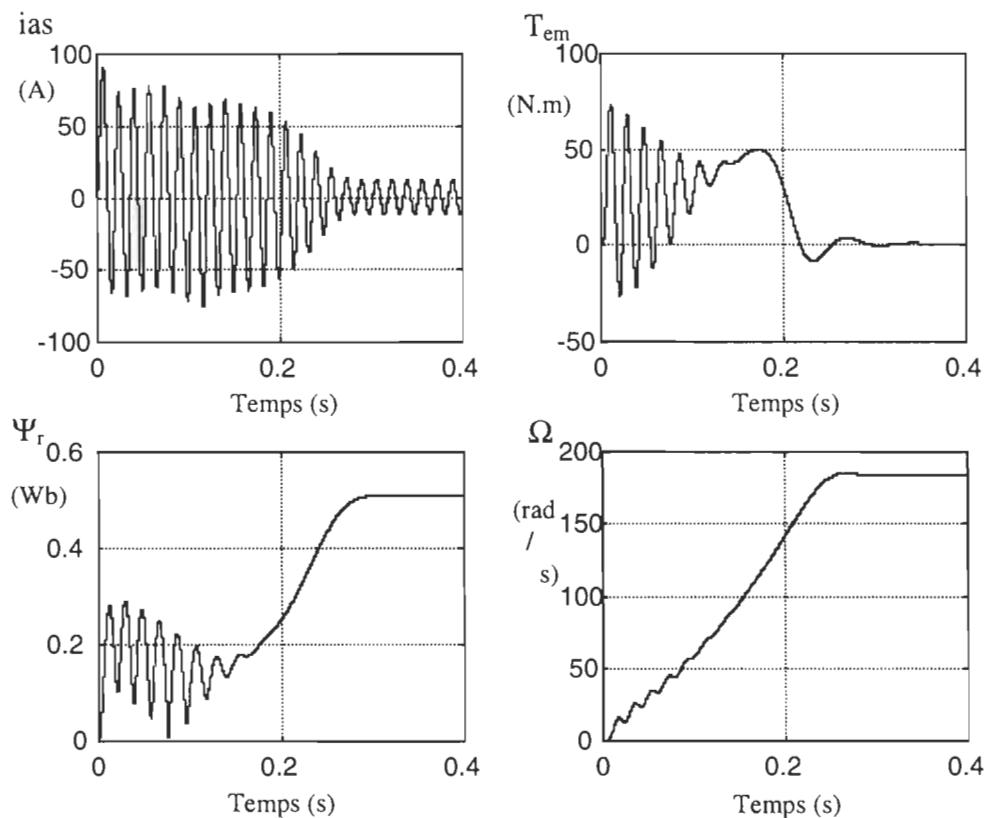


Figure 1.5. Réponses transitoires d'un moteur à induction couplé directement au réseau

## 1.4 Conclusion

Comme on l'a vu, il existe différentes possibilités concernant le choix de l'orientation du repère d'axes dq qui dépendent des objectifs de l'application :

- Axes tournant à la vitesse du rotor : étude des grandeurs statoriques,
- Axes liés au stator : étude des grandeurs rotoriques,

- c) Axes solidaires du champ tournant, repères liés au rotor ou au stator : étude de la commande.

C'est cette dernière solution qui fait correspondre des grandeurs continues aux grandeurs sinusoïdales du régime permanent. La conception du contrôle vectoriel par orientation du flux nécessite ce choix et les modèles d'action dépendent de la position du repère par rapport aux divers axes de flux.

## *Chapitre 2*

# *Principes de la commande vectorielle à flux rotorique orienté*

### **2.1 Introduction**

Les machines à courant continu occupent, encore de nos jours, la majeure partie du marché de la variation de vitesse et de la robotique. On utilise les machines à excitation série pour la traction électrique et celles à excitation séparée pour la robotique. Ces dernières ont une qualité intrinsèque : elles permettent un contrôle séparé du flux et du couple.

$$T_{em} = k \cdot \Psi_f \cdot I_a = K \cdot I_a \cdot I_f \quad (2-1)$$

avec :

$\Psi_f$  : flux imposé par le courant d'excitation  $I_f$  (indépendant de  $I_a$  si la réaction d'induit est négligeable),

$I_a$  : courant induit.

À flux constant, le couple peut être réglé en agissant sur  $I_a$ .

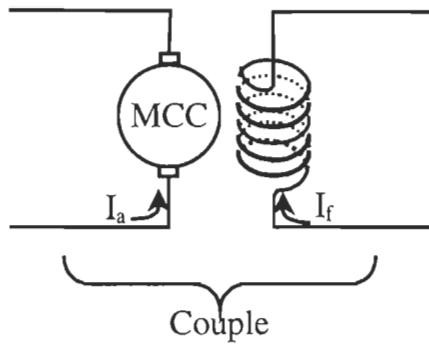


Figure 2.1 Machine à courant continu.

Cependant cette machine sensible nécessite un entretien relativement lourd et possède une puissance massique inférieure et un prix supérieur aux autres technologies de moteurs. On a donc cherché à les remplacer par les machines à induction.

Mais les commandes conventionnelles de la machine à induction n'assurent pas un découplage entre le flux et le couple que ce soit dans le cas d'une alimentation en courant ou en tension. Pour illustrer ceci, rappelons nous que dans les commandes scalaires ( $V/f$  constant) pour augmenter un peu le couple, par augmentation du glissement, il y a lieu à une réduction du flux. Le contrôle du couple ne se fait absolument pas indépendamment de celui du flux. Le couplage entre ces deux variables est responsable de la faible dynamique de ces techniques de commande. Pour réaliser des commandes découplées du flux et du couple d'un moteur à induction, on a recours aux techniques de la commande vectorielle.

## 2.2 Principes de la commande vectorielle

Cette technique qui tire son idée de base du modèle dq du moteur à induction dans le repère synchrone et de la façon dont le moteur à courant continu est contrôlé, est basée sur l'orientation du flux dans la machine selon l'axe d ce qui assure de ce fait une composante en quadrature en tout temps nulle [5-6].

Le couple d'un moteur à induction est donné par :

$$T_{em} = \frac{3}{2L_r} pp.M[i_{qs}\Psi_{dr} - i_{ds}\Psi_{qr}] \quad (2-2)$$

Donc, si on oriente le flux rotorique sur l'axe d ( $\Psi_{dr} = \Psi_r$  et  $\Psi_{qr} = 0$ ) d'un repère lié au champ tournant, alors le couple sera :

$$T_{em} = \frac{3}{2L_r} pp.M[i_{qs}\Psi_r] \quad (2-3)$$

et l'évolution du flux à partir de (1-31) est donnée par :

$$\tau_r \frac{d\Psi_r}{dt} + \Psi_r = M i_{ds} \quad (2-4)$$

Le courant  $i_{ds}$  fixe le flux et le courant  $i_{qs}$  le couple. On retrouve le comportement d'une machine à courant continu. La régulation du flux peut être soit directe soit indirecte.

### 2.2.1 Méthodes directe et indirecte d'orientation de flux

Dans ce chapitre, on va considérer un fonctionnement idéalisé de ces techniques en considérant que la mesure du flux s'effectue de façon précise et fiable dans le cas d'une commande directe. Pour la commande indirecte, on

assumera que les paramètres de la machine n'ont subi aucune variation au cours de son fonctionnement.

### 2.2.1.1 Technique d'orientation directe de flux

L'idée de base de l'orientation du flux est l'utilisation de l'angle du vecteur flux  $\theta_s$  (figure 1.2), pour calculer la phase (dans le repère stationnaire), de façon à orienter le flux selon l'axe d et assurer ainsi un découplage entre le flux et le couple comme le montre la figure 2.2. La plus grande difficulté, qui demeure toujours une sérieuse limitation de cette technique, est la mesure du flux surtout en ce qui concerne le flux rotorique qui n'est pas directement accessible.

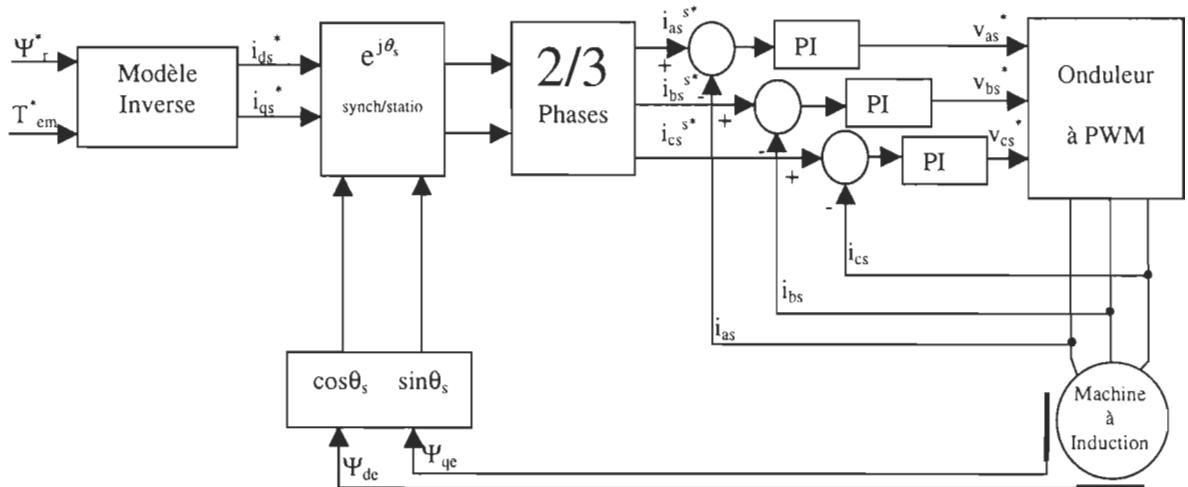


Figure 2.2. Contrôle vectoriel direct

À l'aide de capteurs de flux disposés dans l'entrefer de la machine, on effectue une mesure des composantes directe et transversale du flux dans la machine, soit :  $\Psi_{de}$  et  $\Psi_{qe}$ . On a :

$$\Psi_e = \Psi_{de} + j\Psi_{qe} \quad (2-5)$$

Dans le repère stationnaire de la machine, on a les relations :

$$\Psi_{de} = |\Psi_e| \cos(\theta_s) \quad (2-6)$$

$$\Psi_{qe} = |\Psi_e| \sin(\theta_s) \quad (2-7)$$

$$|\Psi_e| = \sqrt{\Psi_{de}^2 + \Psi_{qe}^2} \quad (2-8)$$

où  $\theta_s$  correspond à la position du vecteur flux dans le système d'axes dq.

Dans une commande vectorielle directe, nous avons besoin de  $\cos\theta_s$  et  $\sin\theta_s$  afin de réaliser la transformation des consignes de courant (exprimées dans le repère synchrone où elles sont des grandeurs continues) en leurs équivalentes dans le repère stationnaire (grandeur sinusoïdales de fréquence  $\omega_s = \frac{d\theta_s}{dt}$ ). À partir de (2-6), (2-7) et (2-8), il est évident que :

$$\cos(\theta_s) = \frac{\Psi_{de}}{\sqrt{\Psi_{de}^2 + \Psi_{qe}^2}} \quad (2-9)$$

$$\sin(\theta_s) = \frac{\Psi_{qe}}{\sqrt{\Psi_{de}^2 + \Psi_{qe}^2}} \quad (2-10)$$

Une fois le flux d'entrefer déterminé, on procède à la détermination du flux qu'on désire orienter. Les équations (2-11) et (2-12) donnent le flux rotorique en fonction de l'entrefer et des courants statoriques [6] :

$$\Psi_{dr} = \left(\frac{L_r}{M} + 1\right) \Psi_{de} - L_r i_{ds} \quad (2-11)$$

$$\Psi_{qr} = \left(\frac{L_r}{M} + 1\right) \Psi_{qe} - L_r i_{qs} \quad (2-12)$$

### 2.2.1.2 Techniques d'orientation indirecte de flux

À l'encontre de la méthode directe, la méthode indirecte est basée sur la détermination de la position du flux à partir du modèle de la machine. En inversant le modèle dq de la machine, comme on va le voir dans ce chapitre, on détermine la consigne de la pulsation du glissement. Cette dernière additionnée à la vitesse électrique du rotor donne la pulsation du champ tournant. L'intégration de cette dernière donne la position du flux.

On considère le système d'axes dq de la figure 2.3 :

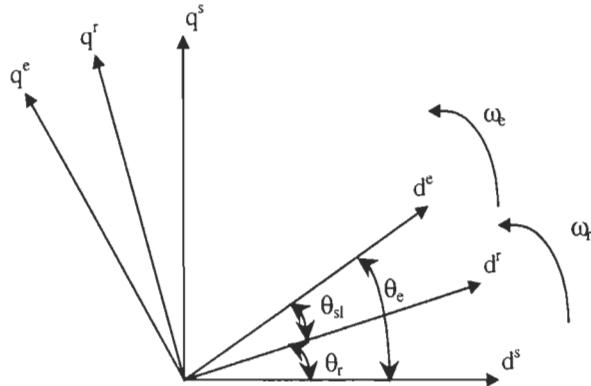


Figure 2.3. Systèmes d'axes dq

avec :

- $(d^s, q^s)$  : repère stationnaire lié au stator
- $(d^r, q^r)$  : repère stationnaire lié au rotor
- $(d^e, q^e)$  : repère synchrone.

Les équations des tensions de la machine exprimées dans le repère synchrone lié au stator et utilisant les flux comme variables d'état sont :

$$\frac{d\Psi_{ds}}{dt} = v_{ds} - \frac{1}{\sigma\tau_s} (\Psi_{ds} - \frac{M}{L_r} \Psi_{dr}) + \omega_e \Psi_{qs} \quad (2-13)$$

$$\frac{d\Psi_{qs}}{dt} = v_{qs} - \frac{1}{\sigma\tau_s} (\Psi_{qs} - \frac{M}{L_r} \Psi_{qr}) - \omega_e \Psi_{ds} \quad (2-14)$$

$$\frac{d\Psi_{dr}}{dt} = -\frac{1}{\sigma\tau_r} (\Psi_{dr} - \frac{M}{L_s} \Psi_{ds}) - \omega_r \Psi_{qr} + \omega_{sl} \Psi_{qr} \quad (2-15)$$

$$\frac{d\Psi_{qr}}{dt} = -\frac{1}{\sigma\tau_r} (\Psi_{qr} - \frac{M}{L_s} \Psi_{qs}) + \omega_r \Psi_{dr} - \omega_{sl} \Psi_{dr} \quad (2-16)$$

$$T_{em} = \frac{3}{2} pp.M [\Psi_{qs} \Psi_{dr} - \Psi_{ds} \Psi_{qr}] \quad (2-17)$$

Les courants dans la machine sont donnés par les relations :

$$\begin{aligned} i_{ds} &= \frac{1}{\sigma L_s} \left( \Psi_{ds} - \frac{M}{L_r} \Psi_{dr} \right) \\ i_{qs} &= \frac{1}{\sigma L_s} \left( \Psi_{qs} - \frac{M}{L_r} \Psi_{qr} \right) \\ i_{dr} &= \frac{1}{\sigma L_r} \left( \Psi_{dr} - \frac{M}{L_s} \Psi_{ds} \right) \\ i_{qr} &= \frac{1}{\sigma L_r} \left( \Psi_{qr} - \frac{M}{L_s} \Psi_{qs} \right) \end{aligned} \quad (2-18)$$

### 2.2.1.3 Théorie d'orientation indirecte du flux rotorique

Calculons les expressions des flux statoriques en fonction des flux rotoriques et des courants statoriques. Nous avons :

$$\Psi_{ds} = \sigma L_s i_{ds} + \frac{M}{L_r} \Psi_{dr} \quad (2-19)$$

$$\Psi_{qs} = \sigma L_s i_{qs} + \frac{M}{L_r} \Psi_{qr} \quad (2-20)$$

En remplaçant ces équations dans (2-15) et (2-16), on aura :

$$\frac{d\Psi_{dr}}{dt} = -\frac{1}{\tau_r}(\Psi_{dr} - Mi_{ds}) + \omega_{sl}\Psi_{qr} \quad (2-21)$$

$$\frac{d\Psi_{qr}}{dt} = -\frac{1}{\tau_r}(\Psi_{qr} - Mi_{qs}) - \omega_{sl}\Psi_{dr} \quad (2-22)$$

L'expression des courants rotoriques en fonction des flux rotoriques et des courants statoriques est donnée par :

$$i_{dr} = \frac{\Psi_{dr} - Mi_{ds}}{L_r} \quad (2-23)$$

$$i_{qr} = \frac{\Psi_{qr} - Mi_{qs}}{L_r} \quad (2-24)$$

En remplaçant (2-23) et (2-24) dans l'expression du couple donnée par (2-17), on aura :

$$T_{em} = \frac{3}{2L_r} pp.M [i_{qs}\Psi_{dr} - i_{ds}\Psi_{qr}] \quad (2-25)$$

Le modèle découplé de la machine en orientation du flux rotorique, comme présenté à la figure 2.4, est donné par les équations suivantes :

$$\Psi_r = \left( \frac{M}{1 + s\tau_r} \right) i_{ds} \quad (2-26)$$

$$\omega_{sl} = \left( \frac{M}{\tau_r \cdot \Psi_r} \right) i_{qs} \quad (2-27)$$

$$T_{em} = \frac{3 \cdot pp \cdot M}{2 \cdot L_r} \cdot \Psi_r \cdot i_{qs} \quad (2-28)$$

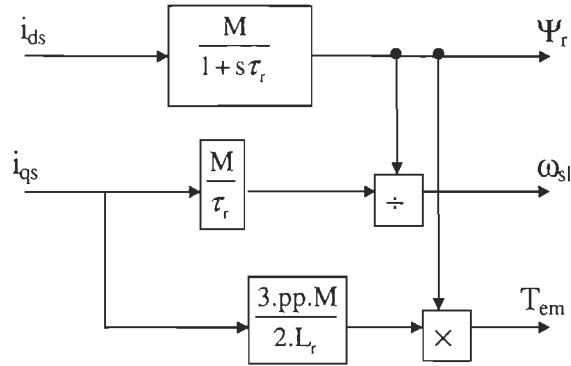


Figure 2.4. Modèle découplé de la machine à induction dans le cas d'une orientation du flux rotorique.

Quant au contrôleur à orientation du flux rotorique basé sur le modèle inverse donné par la figure 2.5, nous avons :

$$i_{ds}^* = \left( \frac{1 + s\tau_r}{M} \right) \Psi_r^* \quad (2-29)$$

$$i_{qs}^* = \left( \frac{2.L_r}{3.pp.M} \right) \frac{T_{em}^*}{\Psi_r^*} \quad (2-30)$$

$$\omega_{sl}^* = \left( \frac{M}{\tau_r \cdot \Psi_r^*} \right) i_{qs}^* \quad (2-31)$$

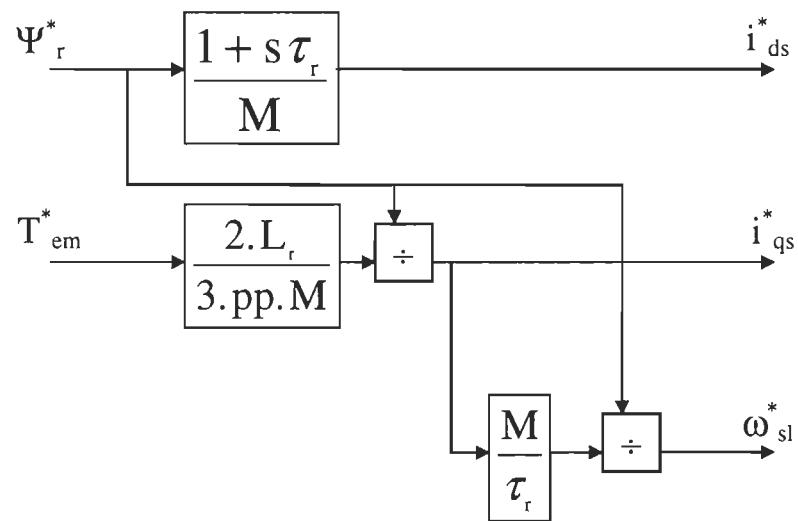


Figure 2.5. Structure d'un contrôleur du flux rotorique.

#### 2.2.1.4 Implantation de l'orientation indirecte du flux rotorique

À partir des consignes de flux et de couple, le contrôleur à orientation du flux calcule les deux commandes de courant et de glissement à imposer à la machine. Les consignes de courant sont exprimées dans le repère synchrone. Elles doivent être converties en grandeurs de phase (a,b,c). Pour ce faire, nous avons besoin d'effectuer une transformation  $(d^e-q^e) \rightarrow (d^s-q^s)$  permettant de retrouver les grandeurs stationnaires (liées au stator) puis une transformation de  $(d^s-q^s) \rightarrow (a,b,c)$  permettant de retrouver les trois commandes de courant  $(i_{as}^*, i_{bs}^*, i_{cs}^*)$ .

Le passage du référentiel synchrone au référentiel stationnaire se fait en considérant :

$$\theta_e = (\theta_r + \theta_{sl}) \quad (2-32)$$

$$\theta_r = \int \omega_r \cdot dt = \int pp \cdot \omega_m \cdot dt \quad (2-33)$$

$$\theta_{sl} = \int \omega_{sl} \cdot dt \quad (2-34)$$

$$\begin{bmatrix} i_{ds}^{s*} \\ i_{qs}^{s*} \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \cdot \begin{bmatrix} i_{ds}^{e*} \\ i_{qs}^{e*} \end{bmatrix} \quad (2-35)$$

La position rotorique est obtenue par une mesure de la vitesse de la machine, tandis que l'angle  $\theta_{sl}$  est donné par l'intégration du glissement calculé par le contrôleur à orientation du flux.

La matrice de passage entre le référentiel stationnaire et le référentiel de phase (a,b,c) est donnée par l'inverse de la matrice de transformation orthogonale de PARK.

$$\begin{bmatrix} i_{as}^* \\ i_{bs}^* \\ i_{cs}^* \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_{ds}^{s*} \\ i_{qs}^{s*} \end{bmatrix} \quad (2-36)$$

**Remarque :** La consigne du flux est considérée constante ce qui donne :  $(1 + s\tau_r).\Psi^* = \Psi^*$

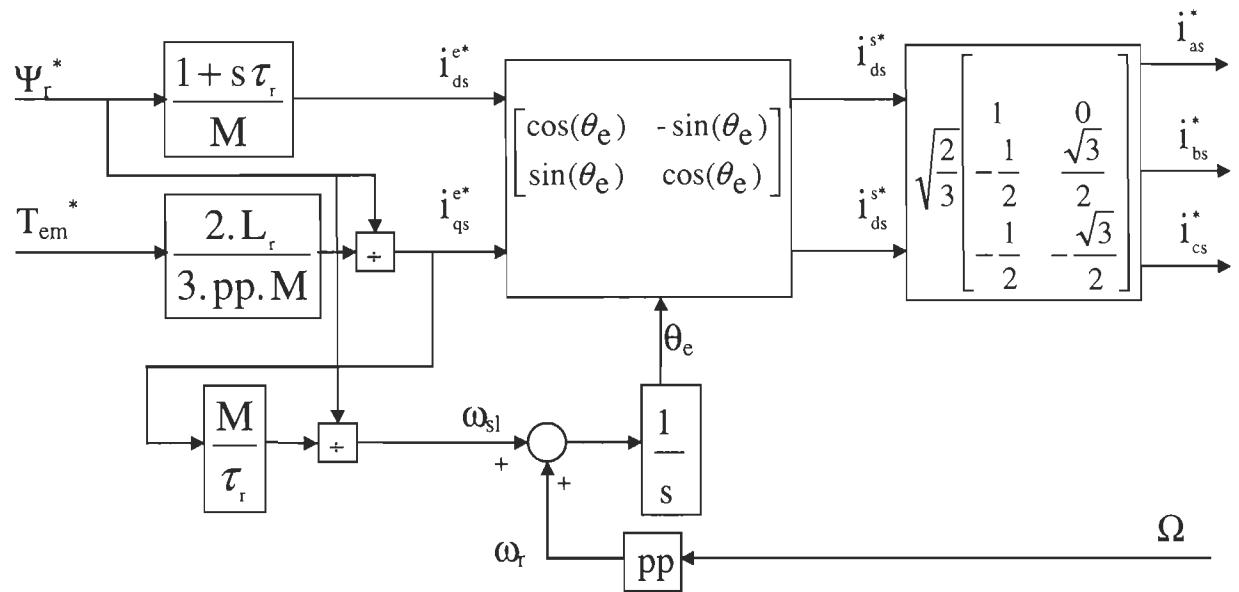


Figure 2.6 Schéma d'implantation d'un régulateur à orientation du flux rotorique

Les flux statoriques sont déterminés à partir des flux rotoriques par les équations (2-19) et (2-20). En injectant ces dernières dans les équations de tensions et en considérant la condition d'orientation du flux rotorique, nous déduisons :

$$v_{ds}^{e*} = (R_s + s\sigma L_s).i_{ds}^{e*} - \omega_e \sigma L_s \cdot i_{qs}^{e*} + \frac{M}{L_r} \cdot \frac{d\Psi_r}{dt} \quad (2-37)$$

$$v_{qs}^{e*} = (R_s + s\sigma L_s).i_{qs}^{e*} + \omega_e \sigma L_s \cdot i_{ds}^{e*} + \omega_e \cdot \frac{M}{L_r} \Psi_r \quad (2-38)$$

$(i_{ds}^*, i_{qs}^*)$  sont donnés par les équations (2-29) et (2-30),  $\omega_{sl}$  est déterminée à partir de (2-31).

Les consignes de tension (2-37) et (2-38) présentent un élément dérivé ce qui peut causer une instabilité. De ce fait, à partir des consignes de courant  $(i_{ds}^*, i_{qs}^*)$  calculées pour une alimentation en courant, on soustrait les courants statoriques captés de la machine après transformation pour fournir les signaux d'entrées aux correcteurs de type PI. La présence d'un intégrateur suffit pour assurer la précision en régime permanent [6-18]. Les sorties des correcteurs constitueront les consignes de tension  $(v_{ds}^*, v_{qs}^*)$  comme il est représenté sur la figure 2.7.

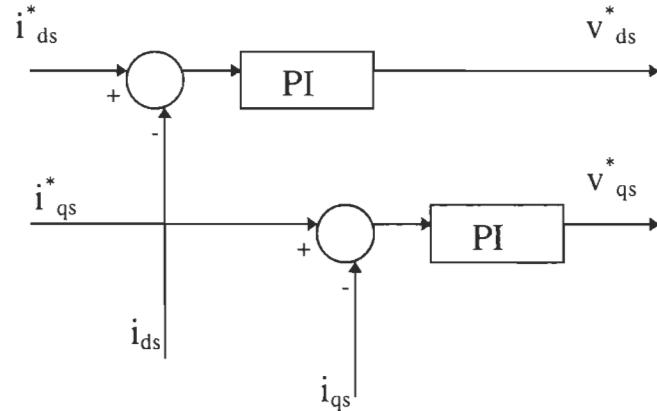


Figure 2.7 Transformation consignes courant en tension

### 2.3 Analyse de la commande vectorielle à orientation du flux rotorique

Le modèle est simplifié car, en tant que première approximation, le convertisseur a été supposé idéal. De ce fait, les consignes des tensions statoriques sont appliquées directement à la machine (figure 2.8). Cette approximation est tout à fait valide dans la mesure où les convertisseurs utilisés dans ce genre d'entraînement réagissent instantanément.

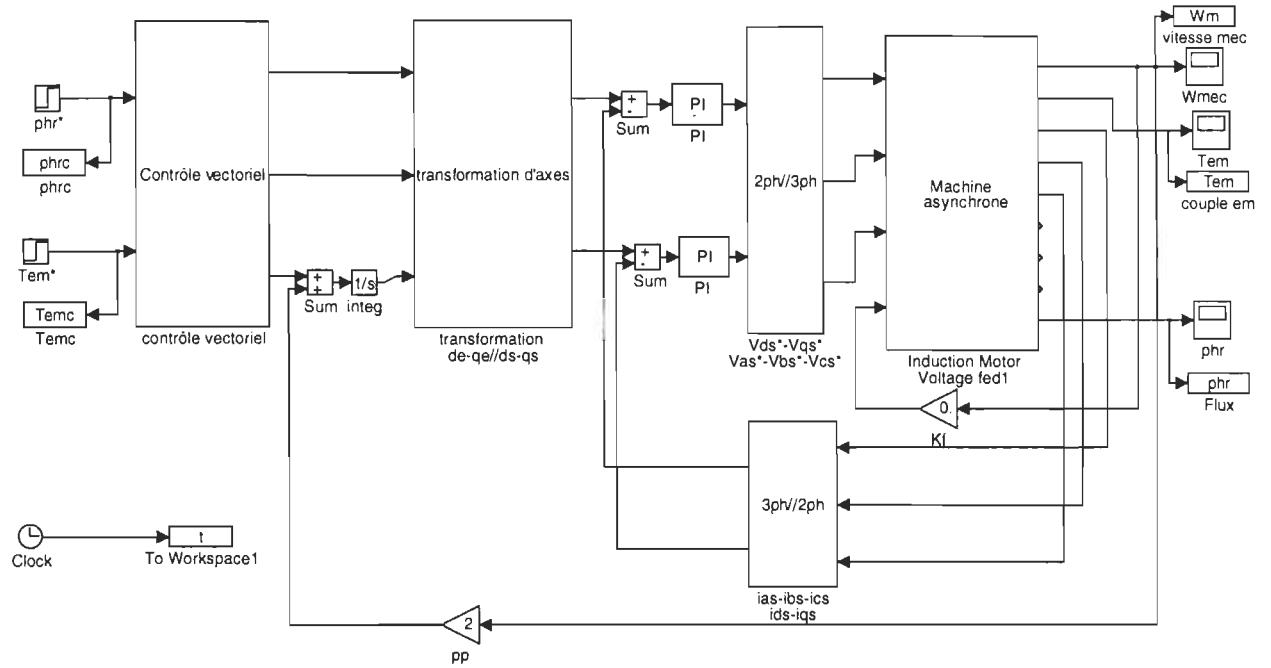
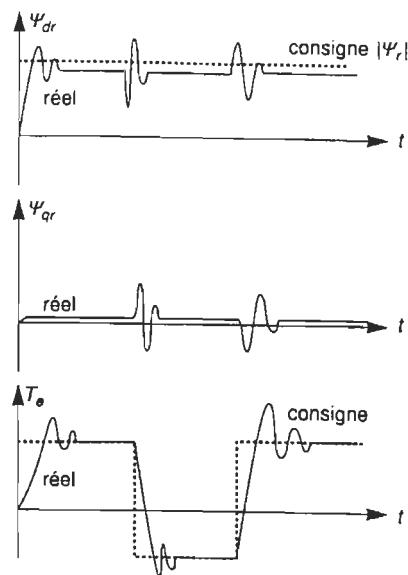


Figure 2.8. Modèle Simulink du contrôle vectoriel d'une machine à induction alimentée en tension.

Pour évaluer les performances en boucle ouverte, les essais suivants ont été appliqués :

- Réponse indicielle du couple électromagnétique du moteur pour une variation de la consigne du couple à flux constant;
  - Réponse indicielle du flux dans le moteur pour une variation de la consigne de flux à couple constant.



Couplage du flux et du couple  
avec une commande scalaire.

Figure 2.9. Réponse d'une commande scalaire [20]

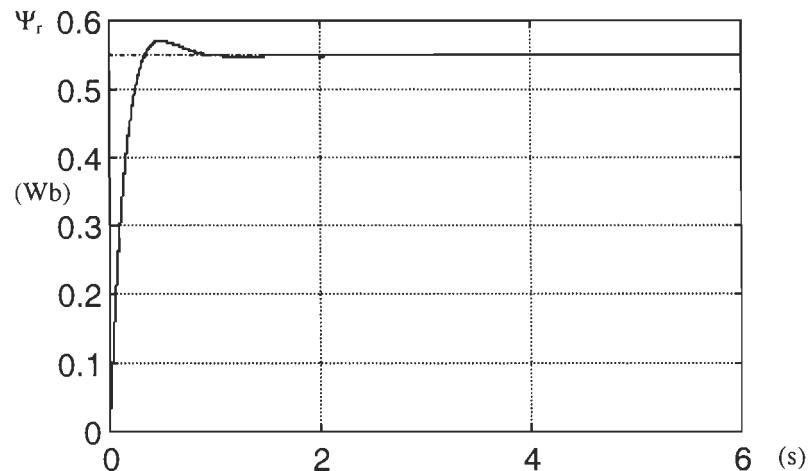


Figure 2.10. Flux rotorique et sa référence

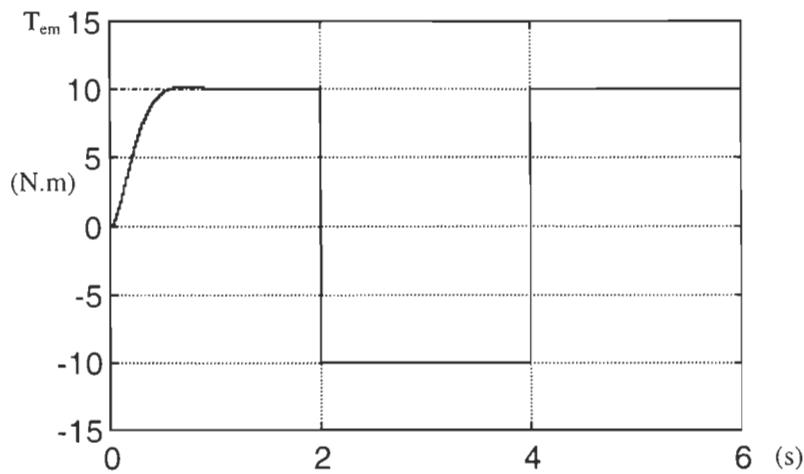


Figure 2.11. Couple et sa référence pour un flux constant

Sur les figures 2.10 et 2.11, on remarque qu'à flux constant, le couple électromagnétique produit par la machine répond quasi instantanément à toute variation de la consigne de couple. Un point très intéressant à remarquer c'est que le flux dans la machine n'est pas affecté par la perturbation du couple ce qui veut dire qu'à flux constant, le système est parfaitement découplé. Dans le cas d'une commande scalaire (figure 2.9) à flux constant, le couple répond avec un certain retard et après quelques oscillations avant de rejoindre sa consigne. De plus, le couple dans ce cas influence le flux même lorsqu'il atteint son régime permanent.

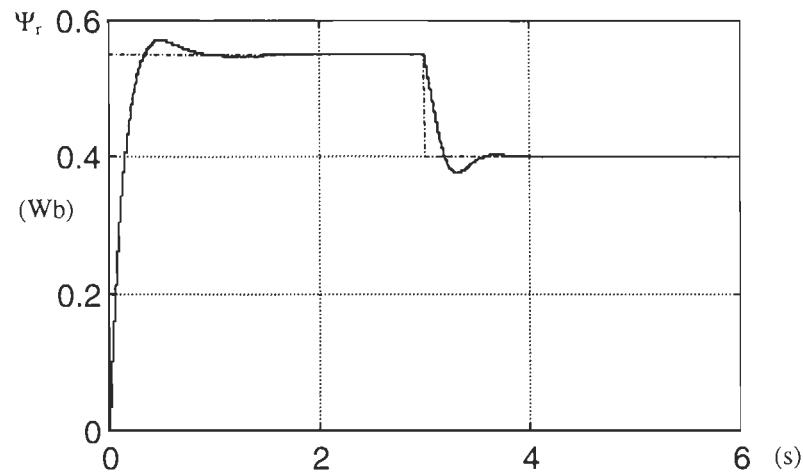


Figure 2.12. Flux rotorique et sa consigne pour un couple constant

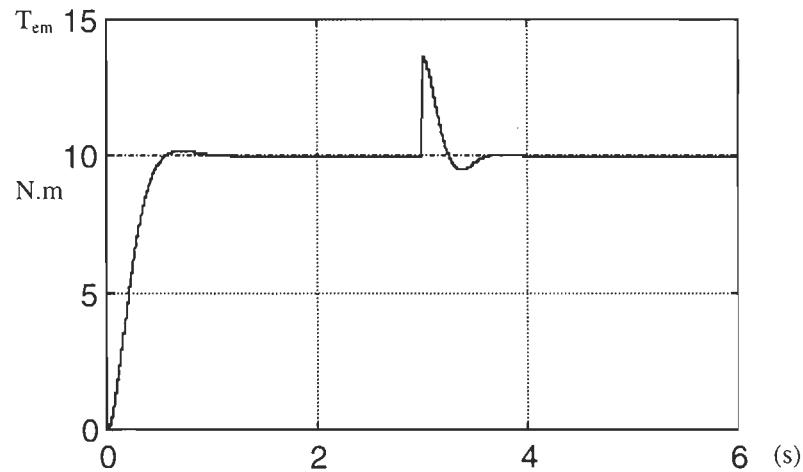


Figure 2.13. Couple et sa référence pour un flux variable

La figure 2.12 montre la réponse du système à une variation de la consigne de flux. Ce dernier rejoint sa consigne avec une constante de temps  $\tau_r$ . Ceci vient du fait que lors du développement du modèle du régulateur, le flux a été supposé constant et de ce fait un terme dérivé a été omis. Cette supposition est tout à fait valide, car dans un entraînement de ce type le flux est maintenu constant à sa valeur nominale en bas de la vitesse synchrone,

réglage dans laquelle les meilleures performances sont assurées. Au delà de la vitesse de synchronisme, là où le flux doit être affaibli, le terme dérivé doit être pris en compte bien que son implantation pratique soit onéreuse (problèmes liés aux circuits déivateurs). Sur la figure 2.13, le couple électromagnétique est aussi affecté par la variation de flux. Ceci est tout à fait normal vu que le régulateur à découplage a été développé en considérant le flux constant.

## 2.4 Conclusion

En comparaison avec les commandes scalaires, le contrôle vectoriel reste parmi les meilleures stratégies commandes de des machines asynchrones, qui assure un découplage efficace entre les variables de commande soient le flux et le couple tout en manipulant l'un sans influencer l'autre.

## *Chapitre 3*

# *Influence de la constante de temps rotorique sur le contrôle vectoriel*

### **3.1 Introduction**

Nous avons vu dans le chapitre 2 que la commande vectorielle effectuait un découplage des moteurs à induction grâce à une orientation directe ou indirecte du flux dans la machine. Ceci permet donc d'assurer à ce type de machines extrêmement non-linéaires un modèle simple, découplé et pouvant être contrôlé de façon analogue à celle d'une machine à courant continu à excitation séparée. Ces performances sont atteintes dans la mesure où le flux est maintenu constant. En effet, en maintenant le flux constant dans la machine, le couple du moteur à induction se comporte en  $K.\Psi_r.i_{qs}$ , et peut être contrôlé indépendamment du flux à l'aide de la composante quadrature du courant statorique  $i_{qs}$ . De cette façon, le couple de la machine est commandé directement et sa réponse transitoire est instantanée dès que le flux est établi dans le moteur.

Bien que l'implantation de cette technique à orientation de flux soit attractive, elle souffre cependant d'un certain nombre de limitations aussi bien

pour la méthode directe que celle indirecte. Ces limitations sont d'ordre technique dans le cas d'une orientation directe et d'ordre fondamental dans le cas de la commande indirecte. Cette dernière limitation est liée au comportement de la machine en fonction de ses conditions de fonctionnement.

Ce chapitre sera donc consacré à l'étude de ces limitations dans le but de mieux comprendre le comportement de ces entraînements dans des conditions de fonctionnement plus réalistes. On termine le chapitre par une étude thermique de la machine à induction.

### 3.2 Limitations de la méthode indirecte d'orientation de flux

La méthode classique d'orientation indirecte du flux rotorique est une méthode de commande de type action directe. Elle souffre, de ce fait, de l'effet de la variation des paramètres de la machine sur la réponse dynamique du système et particulièrement celle de la constante de temps rotorique. La variation de cette constante de temps peut être due soit à la variation de la résistance rotorique en raison de l'échauffement du moteur, soit à la saturation de l'inductance rotorique, soit à l'effet combiné de ces deux paramètres. Pour illustrer ceci, analysons les résultats d'une simulation effectuée pour évaluer l'effet de ces variations sur les performances d'un entraînement à flux orienté [7].

Pour différentes valeurs de la constante de temps rotorique programmées dans le modèle du moteur (figure 3.1), les réponses d'un entraînement à flux orienté sont montrées aux figures 3.2 et 3.3.

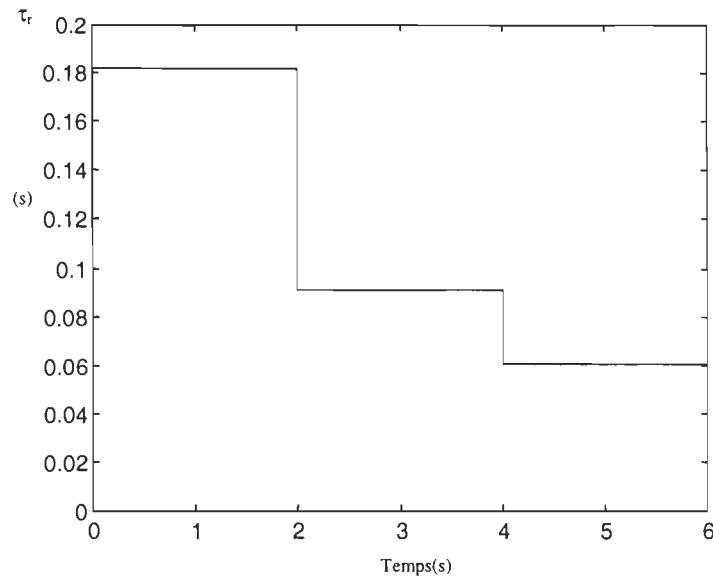


Figure 3.1. Variation de la constante de temps rotorique.

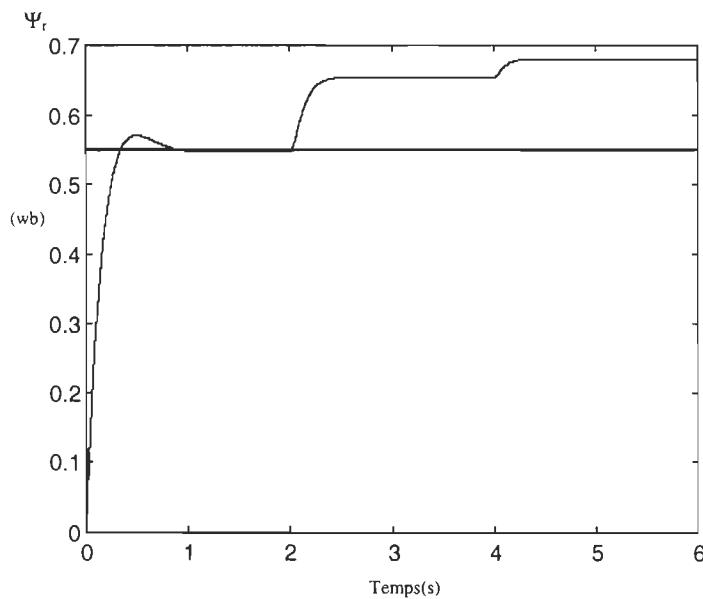


Figure 3.2. Effet de la variation de la constante de temps rotorique sur le flux .

La réponse dynamique du couple produit par la machine (en régime établi) montre que celui-ci baisse tout le temps alors que le flux dans la

machine augmente tout le temps dans la machine lorsque la constante de temps rotorique subit une baisse jusqu'à la saturation.

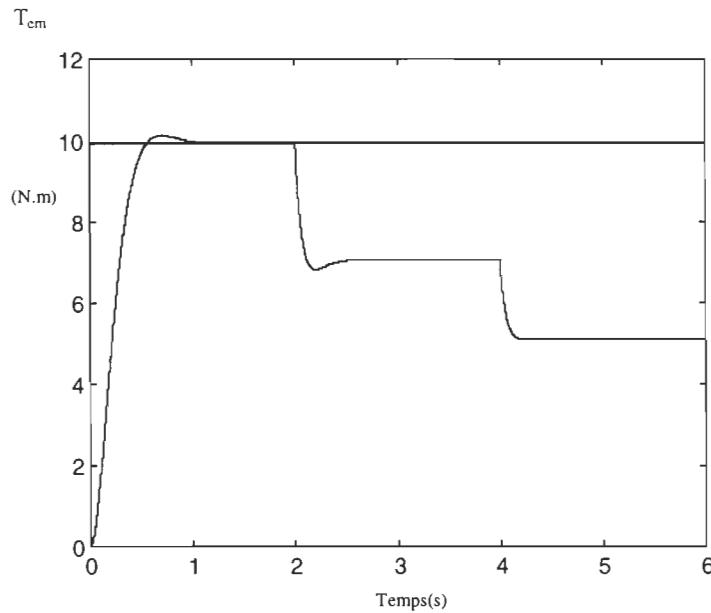


Figure 3.3 Effet de la variation de la constante de temps rotorique sur le couple

Pour analyser l'effet de la variation de la constante de temps rotorique sur la réponse du flux et du couple d'un entraînement à flux orienté, en régime établi, nous considérons les expressions du flux et du couple dans le repère synchrone données par les relations (2-21) et (2-22), nous obtenons :

$$i_{ds} = \frac{\Psi_{dr} - \tau_r \cdot \omega_{sl} \cdot \Psi_{qr}}{M} \quad (3-1)$$

$$i_{qs} = \frac{\Psi_{qr} - \tau_r \cdot \omega_{sl} \cdot \Psi_{dr}}{M} \quad (3-2)$$

On peut tirer les expressions des flux et du couple en fonction des courants (3-1) et (3-2) et de la constante de temps rotorique  $\tau_r$ .

$$\Psi_{dr} = \frac{M \cdot i_{ds} + \tau_r \cdot \omega_{sl} \cdot M \cdot i_{qs}}{1 + (\tau_r \cdot \omega_{sl})^2} \quad (3-3)$$

$$\Psi_{qr} = \frac{M \cdot i_{qs} + \tau_r \cdot \omega_{sl} \cdot M \cdot i_{ds}}{1 + (\tau_r \cdot \omega_{sl})^2} \quad (3-4)$$

$$\Psi_r = \sqrt{\Psi_{dr}^2 + \Psi_{qr}^2} = M \sqrt{\frac{i_{ds}^2 + i_{qs}^2}{1 + \omega_{sl}^2 \cdot \tau_r^2}} \quad (3-5)$$

$$T_{em} = \frac{3 \cdot pp \cdot M}{2 \cdot L_r} (\Psi_{dr} i_{qs} - \Psi_{qr} i_{ds}) = \frac{3 \cdot pp \cdot M^2 \cdot \omega_{sl}}{2 \cdot L_r} \left[ \frac{\tau_r (i_{ds}^2 + i_{qs}^2)}{1 + \omega_{sl}^2 \cdot \tau_r^2} \right] \quad (3-6)$$

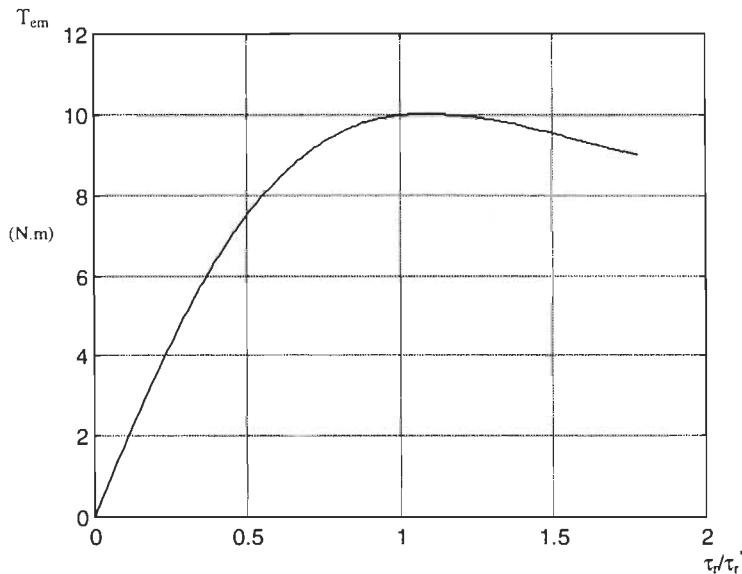


Figure 3.4. Variation du couple en fonction de la constante du temps rotorique

Les figures 3.4 et 3.5 montrent les courbes de la variation du couple et du flux en fonction de la constante de temps rotorique pour des consignes de flux et de couple constantes. Le flux varie d'une façon monotone, alors que le couple augmente pour une certaine valeur de cette constante de temps rotorique puis décroît d'une façon monotone.

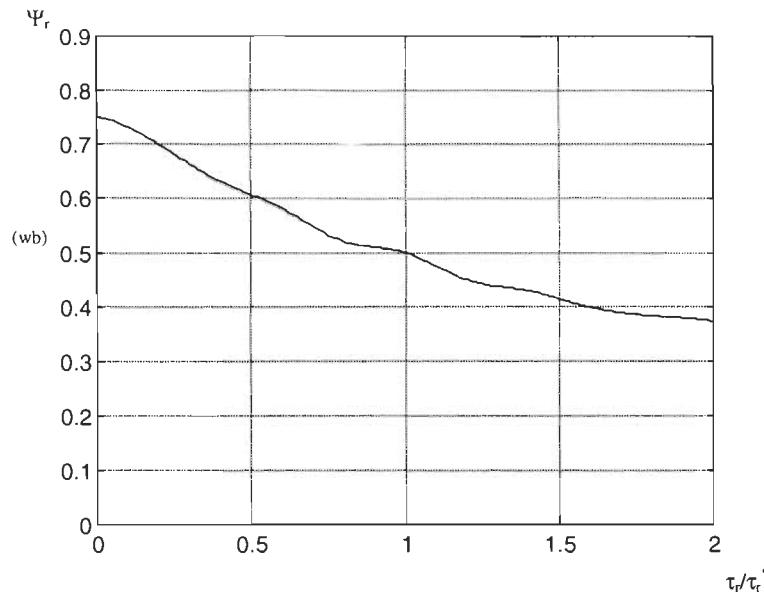


Figure 3.5. Variation du flux en fonction de la constante de temps rotorique

Lorsque le flux est maintenu constant dans la machine, les inductances de la machine peuvent être considérées constantes et égales à la valeur correspondant au niveau de flux de la machine à ce point de fonctionnement. Dans de telles circonstances, la perte de découplage de la machine est principalement due à la variation de la résistance rotorique. Cette dernière augmente en raison de l'échauffement et du phénomène de l'effet de peau à haute fréquence.

### 3.3 Effet de la variation de la constante de temps rotorique sur la fréquence des courants statoriques

Les équations de découplage dans une orientation du flux rotorique sont données par :

$$i_{ds}^* = \left( \frac{1 + s\tau_r}{M} \right) \Psi_r^* \quad (3-7)$$

$$i_{qs}^* = \left( \frac{2 \cdot L_r}{3 \cdot pp \cdot M} \right) \frac{T_{em}^*}{\Psi_r^*} \quad (3-8)$$

$$\omega_{sl}^* = \left( \frac{M}{\tau_r \cdot \Psi_r^*} \right) i_{qs}^* \quad (3-9)$$

En considérant seulement la variation de la résistance rotorique  $R_r$ , on s'aperçoit que l'amplitude des courants  $I_{ds}$  et  $I_{qs}$  n'est pas influencée et que seule la commande du glissement est affectée à cause du terme  $\tau_r$ .

Par conséquent, la fréquence des courants statoriques de l'entraînement à flux rotorique orienté ( $\omega_s = \omega_{sl}^* + \omega_r$ ) se trouve évidemment affectée.

La figure 3.6 montre la courbe de la variation de la fréquence des courants statoriques en fonction de la constante de temps rotorique pour un couple de charge proportionnel à la vitesse. Comme on peut le constater la fréquence des courants statoriques augmente pour certaines valeurs de  $\tau_r$  puis décroît d'une façon monotone.

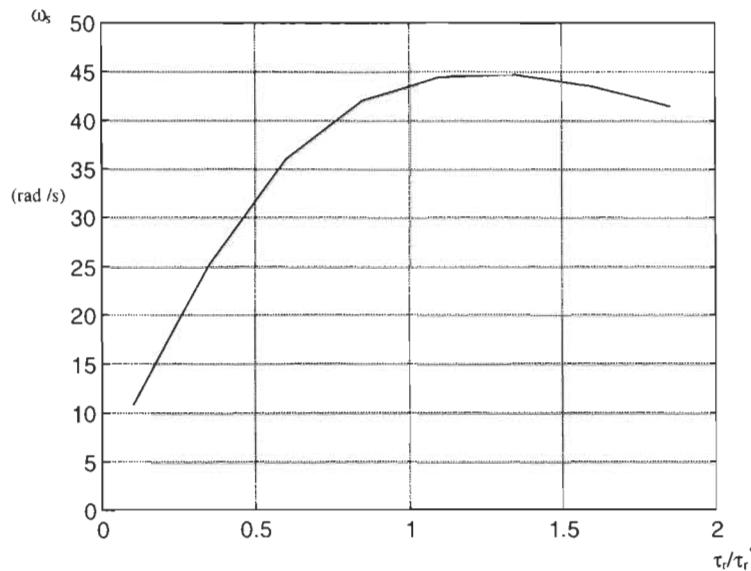


Figure 3.6. Variation de la fréquence des courants statoriques en fonction de la constante de temps rotorique

### 3.4 Modèle thermique de la machine

Une étude expérimentale de la variation de la résistance rotorique, en fonction de la température, peut nous donner une idée sur le taux de variation de cette résistance.

#### 3.4.1 Rappel théorique

Dans une machine à induction, la transformation de l'énergie mécanique en énergie électrique ou inversement s'accompagne toujours d'une perte d'énergie qui est dissipée sous forme calorifique dans les différentes parties de la machine. Les pertes d'énergie se répartissent en trois catégories : les pertes dans le cuivre, les pertes dans le fer et les pertes mécaniques. En dehors de ces trois catégories, il existe aussi d'autres pertes supplémentaires comme il sera décrit par la suite [19]:

### ***Pertes dans le fer***

Ce sont les pertes qui se produisent dans les masses de fer constituant le circuit magnétique de la machine. Elles représentent les pertes par hystéresis et courants de Foucault.

### ***Pertes cuivre***

Ce sont les pertes joules dans les enroulements statoriques et rotoriques. Elles sont proportionnelles au carré du courant qui traverse les enroulements.

### ***Pertes mécaniques***

Elles sont composées des pertes par frottements des tourillons dans les paliers et des pertes par frottements dans l'air (pertes de ventilation). Elles varient avec la vitesse de l'arbre.

### ***Pertes harmoniques***

Avec l'apparition des onduleurs modernes, on a pu diminuer l'effet des pertes harmoniques. Leurs effets dans le cuivre et dans le fer (effet de peau), peuvent être réunis avec les pertes dans le fer et les pertes cuivre.

### ***Pertes supplémentaires***

Ces pertes sont les pertes additionnelles dans le fer et les pertes à hautes fréquences dans le cuivre. Elles sont causées par les fuites dans l'entrefer et les pulsations du flux à haute fréquence. Ces pertes dépendent approximativement de la charge et elles sont supposées être incluses dans les pertes dans le fer et les pertes cuivre.

### 3.4.2 Étude expérimentale

Le moteur à induction est entraîné par un onduleur triphasé fonctionnant à 10 kHz. Le moteur entraîne un alternateur débitant sur une charge triphasée. On mesure la température du rotor à chaque 15 min puis on calcule la résistance rotorique par la loi :

$$R_r = R_a (1 + \alpha \cdot \Delta \theta) \quad (3-10)$$

où :

$R_a$  : résistance rotorique à la température ambiante,

$\alpha = 0.0039$  pour l'aluminium.

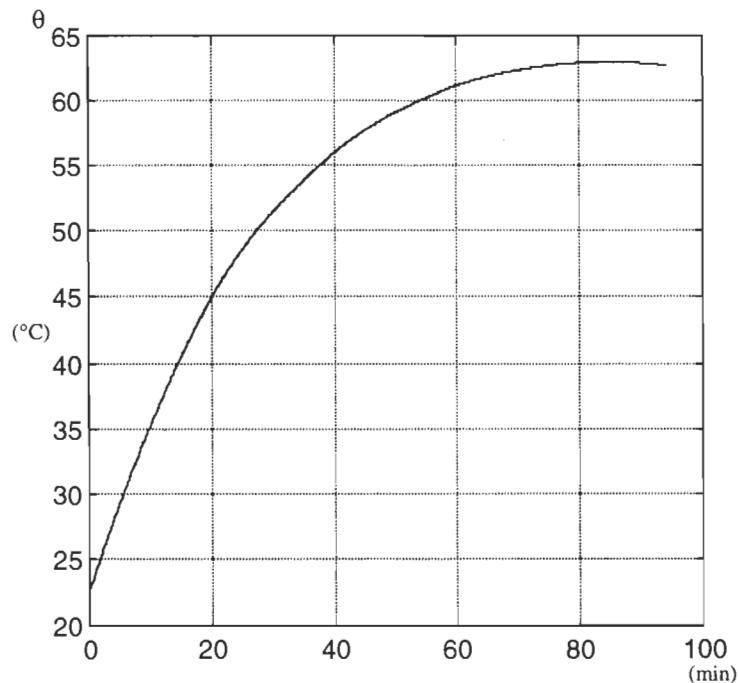


Figure 3.7. Évolution de la température rotorique

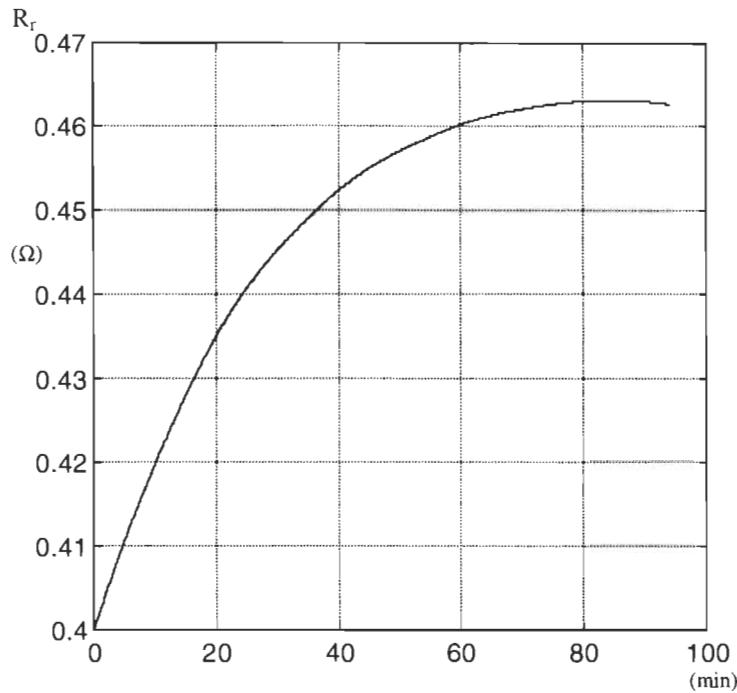


Figure 3.8. Effet de la température sur la résistance rotorique

La figure 3.7 montre l'augmentation de la température dans le rotor, cette dernière est mesurée à l'aide d'un thermomètre à thermocouple d'une grande précision pour différents intervalles de temps. La figure 3.8 montre que durant un fonctionnement de 1 h 40 min, la résistance rotorique a augmenté d'une façon exponentielle jusqu'à 15% de sa valeur nominale à la température ambiante.

### 3.5 Conclusion

Cette technique d'orientation indirecte du flux rotorique, à priori attractive, possède un inconvénient majeur. Le comportement de la machine et de sa commande vectorielle s'est fortement dégradé suite à la variation des paramètres liés à la température, à la fréquence (résistance rotorique) ou à la

saturation (inductance). Souvent une adaptation en ligne de ces paramètres est nécessaire pour maintenir un niveau de découplage et de performance acceptable.

Cette constatation justifie les efforts réalisés par les concepteurs pour définir des procédures conduisant à une connaissance aussi précise que possible de l'évolution de  $\tau_r$ , un paramètre très sensible à la température, à l'état magnétique de la machine et à la fréquence (effet de peau).

## *Chapitre 4*

# *Théorie des réseaux de neurones*

### **4.1 Introduction**

Récemment plusieurs techniques d'estimation ont vu le jour et ont été appliquées pour l'estimation de la constante de temps rotorique dans le contrôle vectoriel. Parmi les techniques les plus utilisées, on trouve les estimateurs basés sur le filtre de Kalman et sur la méthode des moindres carrées, mais toutes ces méthodes restent dépendantes des paramètres de la machine. Une nouvelle technique basée sur les réseaux de neurones, complètement indépendante des paramètres de la machine, sera présentée au chapitre suivant. Dans ce chapitre, on va présenter quelques généralités sur les réseaux de neurones artificiels.

Les études des réseaux de neurones artificiels (RNA) datent depuis les années 1940. Grâce aux développements des recherches sur le cerveau et la disponibilité des outils de simulation, Mc Culloch et Pitts [9] étudièrent un

ensemble de neurones formels interconnectés. Ce réseau, déjà développé à l'époque, permettait d'effectuer quelques opérations logiques simples. Jusqu'aux années 1980, la recherche était freinée par la limitation théorique du perceptron. Peu après cette époque, Hopfield [8] lança de nouveau la recherche dans ce domaine après avoir montré l'analogie entre les RNA et les systèmes physiques (en 1982).

Les RNA sont des assemblages fortement connectés d'unités de calcul (neurones formels). Le neurone formel est une formulation mathématique simplifiée d'un neurone biologique. Les RNA ont la capacité de mémorisation, de généralisation et surtout d'apprentissage qui est le phénomène le plus important.

Après les années 1990, quelques travaux scientifiques ont vu le jour dans le domaine de l'électronique de puissance, parmi ces applications, on trouve la commande des machines électriques et des convertisseurs de puissance.

## 4.2 Fondements biologiques

### 4.2.1 Le cerveau

Le système nerveux central est composé d'un ensemble d'éléments de base appelés neurones. Ces neurones sont fortement interconnectés entre eux et on dit que le cerveau en possède environ cent milliards. C'est le centre de régulation et de communication de l'organisme; nos pensées, nos actions, nos émotions attestent son activité. Le système nerveux remplit trois fonctions étroitement liées: il reçoit de l'information sur le changement qui se produit

tant à l'extérieur qu'à l'intérieur de l'organisme; il traite l'information et détermine l'action à entreprendre à tout moment; et enfin il fournit une réponse motrice (commande) qui active les muscles ou les glandes [8].

#### **4.2.2 Le neurone**

Le neurone est l'élément de base du cerveau (voir figure 4.1). Il possède des fonctions spécialisées, comme recevoir des signaux en provenance d'autres neurones, intégrer ces signaux, produire un influx nerveux, le conduire et le transmettre à une autre cellule capable de le recevoir.

#### **4.2.3 Structure du neurone**

Chaque neurone est constitué des éléments suivants: le corps cellulaire, les dendrites et l'axone. Le corps cellulaire contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et des autres molécules qui assurent la survie du neurone. Les dendrites sont des petites ramifications de quelques dixièmes de microns de diamètre et d'une longueur de quelques dizaines de microns. Enfin l'axone qui est la fibre nerveuse sert de moyen de transport pour les signaux émis par le neurone. Sa longueur varie d'un millimètre à plus d'un mètre, il se ramifie à son extrémité pour permettre la communication avec d'autres cellules.

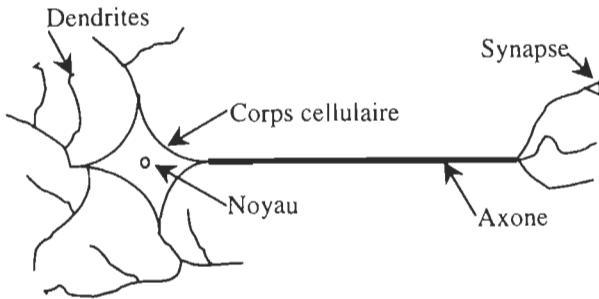


Figure 4.1. Un neurone biologique simple

La communication entre neurones est très complexe. Le neurone émet une impulsion nerveuse du corps cellulaire, c'est une décharge électrique qui prend chemin le long de l'axone vers les terminaisons axonales. Là, elle affectera tous les neurones reliés à ce neurone émetteur par l'intermédiaire de jonctions (appelées synapses) entre les terminaisons axonales et les dendrites des autres neurones. La synapse est le point de contact entre les neurones où le signal électrique est converti en un signal biologique. Une substance biologique appelée neurotransmetteur traverse la courte distance entre les deux cellules et excite les neurones qui la reçoivent. Les impulsions excitatrices convergent en un temps très court sur un même neurone, celui-ci produira en général à son tour une impulsion nerveuse et ainsi de suite. L'effet inverse existe, un neurone peut inhiber un autre neurone pour réduire sa tendance à produire des impulsions nerveuses [9-10].

## 4.3 Le neurone formel

### 4.3.1 Modèle de Mac Culloch et Pitts

La première modélisation de neurone formel date des années quarante. Elle a été présentée par Mc Culloch et Pitts [11], c'est un automate à seuil dont l'unité de sortie est binaire; la fonction d'activation utilisée est en seuil. Ce neurone est actif si la sortie est à 1 et inactif si la sortie est à 0. Ce modèle ne peut résoudre que des problèmes simples tels que les fonctions logiques ET, OU, NON etc.

### 4.3.2 Le modèle généralisé

Le neurone formel est la cellule fondamentale d'un réseau de neurones artificiels. Par analogie avec le neurone biologique, le neurone formel doit être apte à accomplir les trois tâches suivantes: collecter, traiter les données qui viennent des neurones émetteurs et transmettre les messages aux autres neurones. La figure 4.2 présente un neurone formel simple.

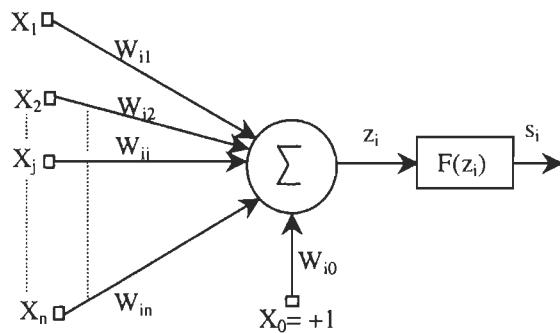


Figure 4.2. Formalisation mathématique d'un neurone

Le neurone formel possède:

- Plusieurs entrées pour collecter les messages provenant des autres neurones formels, le signal  $x_j$  provenant du neurone  $j$  connecté au neurone  $i$  doit être pondéré par le poids  $W_{ij}$  (l'indice  $i$  réfère le neurone récepteur et l'indice  $j$  réfère au neurone émetteur). Si le produit du poids et de l'entrée est positif, le neurone sera excitateur, et si le produit du poids et de l'entrée est négatif, le neurone sera inhibiteur.  $W_{i0}$  est appelé "biais" et sert à ajuster la fonction d'activation, le signal d'entrée correspondant est  $x_0$  et est égal à +1.
- la sommation des signaux constitue une combinaison linéaire des signaux d'entrées  $x_j$  pondérés par les poids  $W_{ij}$ .
- une fonction mathématique limite l'amplitude de la sortie du neurone et est appelée fonction d'activation. Cette dernière limite la sortie en un intervalle fini de valeurs, en général  $[0,1]$  ou  $[-1,1]$ .

La formulation mathématique d'un neurone peut alors s'écrire comme suit [10]:

$$y_i = f(z_i) = f\left(\sum_{j=1}^n W_{ij}x_j - b_i\right) \quad (4-1)$$

$x_i$  entrée élémentaire du neurone  $i$ ,

$z_i$  entrée totale du neurone  $i$ ,

$y_i$  sortie du neurone  $i$ ,

$W_{ij}$  poids de connexion reliant le neurone  $i$  à l'entrée  $x_j$ ,

$b_i$  seuil du neurone  $i$ , qui est égale  $-W_{i0}$ ,

$f(x)$  fonction de décision non linéaire, dite d'activation,

$n$  nombre de neurones  $j$  connectés au neurone  $i$ .

### 4.3.3 Fonction d'activation

Les fonctions d'activation utilisées dans les modèles connexionnistes d'aujourd'hui sont variées. On peut identifier quatre principaux types de fonctions les plus connues [12-13]: binaire à seuil, rampe avec saturation, sigmoïde et gaussienne.

#### *Fonction binaire à seuil*

$$f(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (4-2)$$

Cette fonction d'activation a été utilisée par McCulloch et Pitts, le seuil introduit une non linéarité dans le comportement du neurone, c'est le modèle tout ou rien (figure 4.3a).

#### *Fonction à rampe avec saturation*

$$f(x) = \begin{cases} -1 & \text{si } x \leq -1 \\ x & \text{si } |x| < 1 \\ +1 & \text{si } x \geq 1 \end{cases} \quad (4-3)$$

Cette fonction représente un compromis entre la fonction linéaire et la fonction seuil : entre ses deux bornes, elle confère au neurone une combinaison linéaire de l'entrée. A la limite, la fonction linéaire est équivalente à la fonction seuil (figure 4.3b).

#### *Fonction sigmoïde*

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (4-4)$$

où  $\alpha > 0$  en général  $\alpha = 1$ . La fonction sigmoïde est une fonction continue qui maintient la sortie dans l'intervalle  $[0,1]$ . Son avantage principal est

l'existence de sa dérivée en tout point. Elle est employée en général dans le perceptron multicouches (figure 4.3c).

### Fonction Gaussienne

$$f_i(x) = e^{-\frac{d_i(\chi)^2}{\sigma_i^2}} \quad (4-5)$$

avec  $d_i(\chi) = \|x - c_i\|$

$c_i$  est la position du centre de la fonction appelé noyau,

$d_i(\chi)$ : la distance entre le vecteur  $x$  et le centre  $c_i$  de la fonction-noyau,

$\sigma_i$  : la taille du champ récepteur.

Elle est souvent employée dans les réseaux RBF ("Radial Basis Function") où chaque neurone est conçu pour résoudre de préférence un ensemble de valeurs (figure 4.3d).

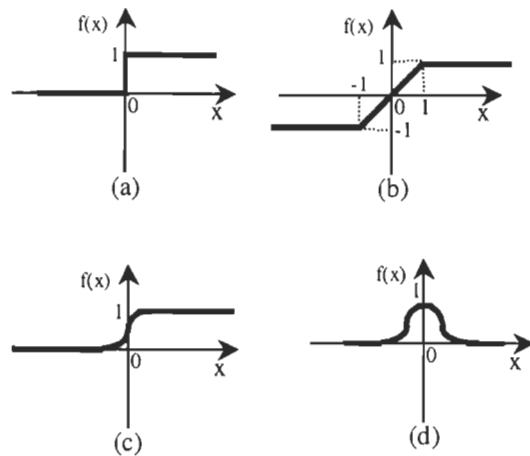


Figure 4.3. Fonctions d'activation binaire avec seuil (a), rampe avec saturation (b), sigmoïde (c), gaussienne (d).

### 4.4 Architectures des réseaux de neurones

L'architecture des réseaux de neurones artificiels peut aller d'une connectivité totale, où tous les neurones sont reliés les uns aux autres, à une

connectivité locale, où les neurones ne sont reliés qu'à leurs plus proches voisins. Il y a plusieurs classifications des réseaux de neurones suivant le nombre de couches, le mode de connexion ou le type de couches. Principalement, nous pouvons identifier quatre classes d'architectures de réseaux [8-12].

#### 4.4.1 Réseau à couche simple

Le réseau à couches est un réseau dont les neurones sont organisés en couches, la forme la plus simple est le réseau à une seule couche. Tous les signaux d'entrées sont propagés des noeuds d'entrées vers la couche de neurones de sortie.

Le nombre de neurones d'entrée (noeuds) et de sortie est en général lié au problème à résoudre. Les entrées seront propagées à travers la matrice des poids  $W$  pour ensuite obtenir la réponse de sortie (figure 4.4).

$$W = \begin{bmatrix} W(1,1) & W(1,2) & \dots & W(1,R) \\ W(2,1) & W(2,2) & \dots & W(2,R) \\ \vdots & \vdots & \ddots & \vdots \\ W(s,1) & W(s,2) & \dots & W(s,R) \end{bmatrix} \quad b = \begin{bmatrix} b(1) \\ b(2) \\ \vdots \\ b(s) \end{bmatrix} \quad (4-6)$$

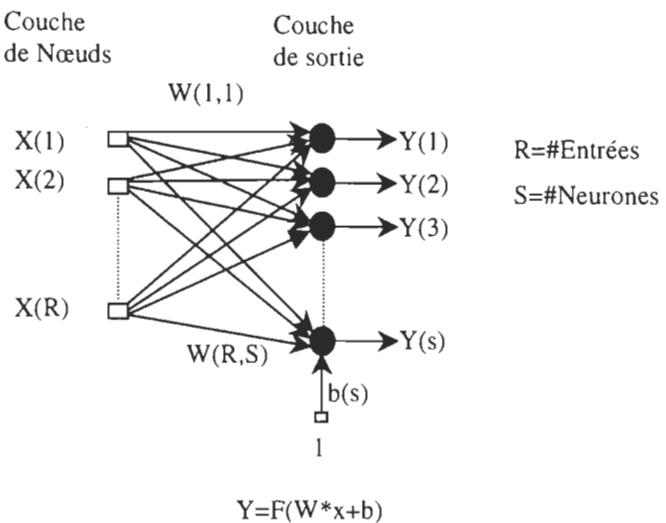


Figure 4.4. Réseau à couche simple

#### 4.4.2 Réseaux multicouches

La seconde classe est caractérisée par une ou plusieurs couches. A chaque couche correspond une matrice de poids  $\mathbf{W}$  et un vecteur de seuil  $\mathbf{b}$ , et on peut avoir aussi des fonctions d'activation différentes pour chaque couche. La fonction des couches cachées intervient entre les entrées  $\mathbf{x}$  et la couche de sortie  $\mathbf{y}$ . Elle permet de résoudre des problèmes plus complexes que le réseau à couche simple.

La figure 4.5 montre un réseau à deux couches, une cachée avec  $s_1$  neurones et une couche de sortie avec  $s_2$  neurones. La connexion entre les neurones peut être totale ou partielle (d'une couche à la suivante).

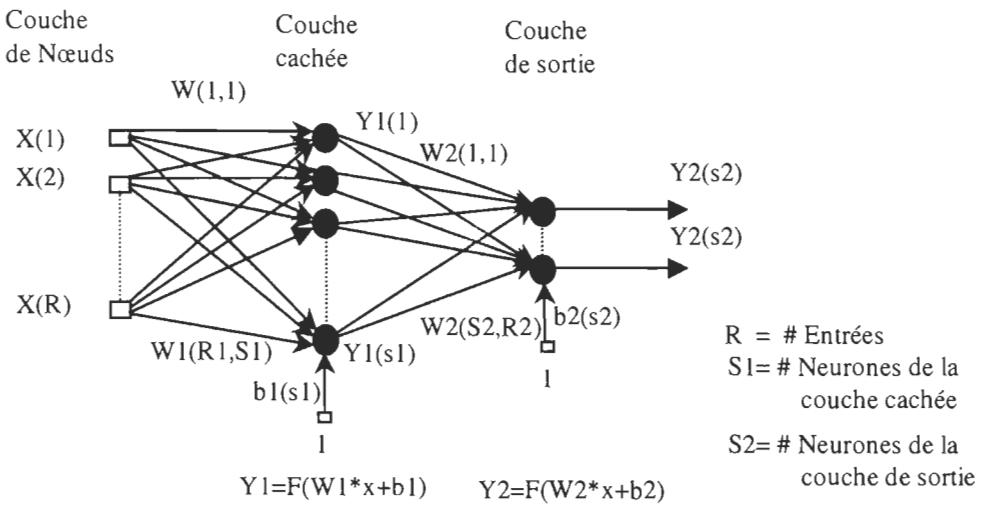


Figure 4.5. Réseau à deux couches (une couche cachée)

#### 4.4.3 Réseaux récurrents

Nous appelons réseaux récurrents les réseaux pouvant comporter des boucles. Ces réseaux se distinguent des autres réseaux par la connexion des sorties de neurones avec leurs entrées. La sortie d'un neurone peut être connectée avec l'entrée du même neurone ou avec celles des autres neurones. L'importance de ces réseaux est qu'ils permettent d'apprendre la dynamique de systèmes, c'est-à-dire qu'ils peuvent imiter le comportement temporel en insérant des délais dans les boucles, reliant l'entrée à la sortie du réseau ou dans des couches internes. La figure 4.6 illustre un réseau récurrent dont les sorties sont bouclées avec les entrées de tous les neurones.

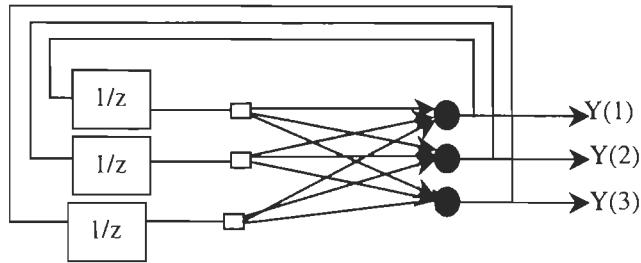


Figure 4.6. Réseau récurrent

#### 4.4.4 Structure "Lattice"

C'est un réseau dont les neurones sont organisés en lignes et en colonnes, où toutes les entrées de neurones sont reliées aux nœuds d'entrées. Ces réseaux peuvent être d'une, de deux ou de plusieurs dimensions. La figure 4.7 montre un réseau de lattice à deux dimensions (de 3 par 2).

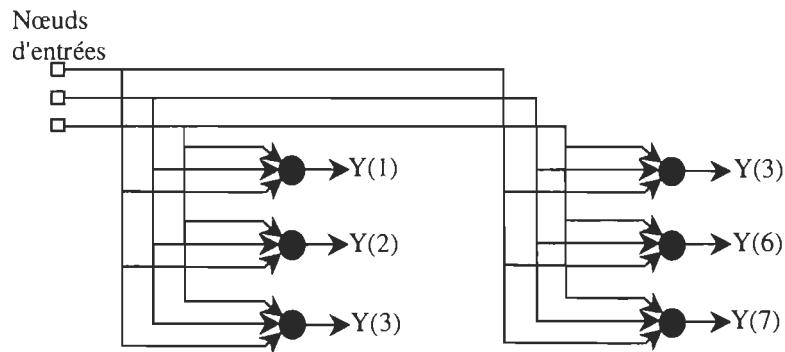


Figure 4.7. "Lattice" à deux dimensions (3 neurones par 2)

### 4.5 Apprentissage des réseaux de neurones

On peut définir l'apprentissage comme la capacité d'emmagasiner des informations (acquisition des propriétés) qui peuvent être rappelées par la suite. Les connaissances d'un réseau connexionniste sont mémorisées dans les poids de connexions qui seront déterminés lors de l'apprentissage. Le but

de l'apprentissage pour un réseau est de trouver un ensemble de poids synoptiques qui minimisent l'erreur entre la sortie du réseau et le résultat désiré. C'est la caractéristique principale des réseaux de neurones [8-12].

#### **4.5.1 Modes d'apprentissage**

Les modes d'apprentissages connexionnistes sont:

- Supervisé: des associations entrées/sorties sont présentées au réseau par un professeur, c'est-à-dire que l'apprentissage dispose d'un comportement de référence vers lequel il tente de faire converger le réseau.
- Non supervisé: cet apprentissage concerne les réseaux ne disposant que des informations d'entrées et qui les regroupent en fonction d'un critère interne dans le but de créer des classes.
- Par cœur : l'information se mémorise parfaitement et définitivement. Dans ce cas, l'erreur est quasi nulle entre la sortie du réseau et celle désirée.
- En ligne : le réseau est en phase d'apprentissage continual, à chaque nouvelle action le réseau apprend et s'ajuste en réalisant la tâche qui lui est assignée.
- Hors ligne : le réseau est entraîné durant la phase d'apprentissage à l'aide d'un ensemble de données qui permettent de fixer les poids de connexions. Une fois les poids calculés, le réseau est fonctionnel et il peut mener la tâche qui lui a été assignée sans modification des poids.

#### 4.5.2 Procédure d'apprentissage

En général, l'apprentissage se fait sur une période relativement longue, durant laquelle celles d'entrées et les données de sorties peuvent être présentées au réseau un grand nombre de fois. Du point de vue calcul, l'objectif de l'apprentissage est de minimiser l'erreur quadratique globale [8].

$$E = \frac{1}{2} \sum_k e_k^2 \quad (4-7)$$

E: l'erreur quadratique globale

$e_k$ : l'erreur entre la valeur désirée et la sortie  $x$

k: nombre d'itérations.

À chaque itération, il faut trouver un  $\Delta W_{kj}$  pour ajuster de nouveau les poids.

$$W_{ij}(k+1) = W_{ij}(k) + \Delta W_{ij}(k) \quad (4-8)$$

Il y a quatre étapes dans la procédure d'apprentissage :

- 1 - Initialisation des poids du réseau ;
- 2 - Présentation des données et propagation d'activation ;
- 3 - Calcul d'erreur ;
- 4 - Calcul du vecteur de corrélation.

Les étapes 2-3-4 sont répétées jusqu'à la fin de l'apprentissage (figure 4.8).

En général, on divise les données d'entrées en deux parties distribuées aléatoirement : une banque d'apprentissage avec laquelle on entraîne le réseau et une banque de généralisation pour le test du réseau à la fin de l'apprentissage. L'arrêt du processus d'apprentissage s'effectue lorsque les performances en test sont optimales, c'est-à-dire avant que le réseau rentre dans l'apprentissage par cœur [8].

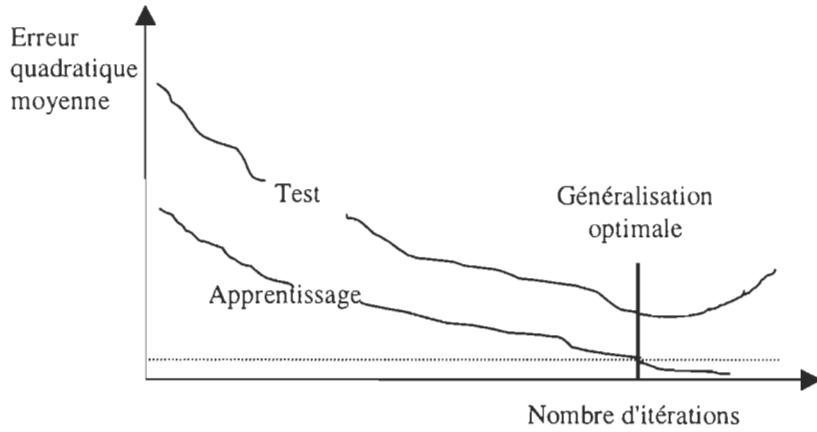


Figure 4.8. Évolution typique de l'erreur quadratique moyenne.

#### 4.5.3 Problèmes d'apprentissage

Malgré la diversité des algorithmes d'apprentissage il n'y a pas de méthode systématique pour la détermination d'un réseau de neurones, en particulier le choix de l'architecture du réseau, le nombre de neurones, le nombre de couches ou le choix des paramètres internes de l'algorithme comme l'erreur quadratique et le nombre d'itérations. Nous pouvons citer quelques problèmes concernant l'apprentissage [9]:

- **Insuffisance de la règle d'apprentissage:** en principe, les réseaux de neurones sont capables de calculs élaborés, en revanche rien ne garantit que la règle d'apprentissage soit capable de tirer profit du plein potentiel du réseau.
- **Minima locaux:** ce problème est rencontré lorsque l'apprentissage converge vers une solution sous optimale. Ce type de problème est difficile à résoudre car généralement la surface d'erreur est inconnue.

- **Choix du nombre de neurones:** le nombre de neurones cachés est particulièrement important parce qu'il détermine la capacité de calcul du réseau. Un nombre insuffisant de neurones cachés peut compromettre la capacité du réseau à résoudre le problème. Inversement, trop de neurones permettent au réseau d'apprendre par cœur au détriment des performances de la généralisation. Même les paramètres propres à l'algorithme d'apprentissage sont difficiles à choisir comme par exemple le pas d'apprentissage.
- **Surapprentissage:** c'est lorsque l'apprentissage d'un réseau reflète trop les particularités du problème au détriment de la tâche réelle.
- **Mauvais échantillonnage:** la matrice d'apprentissage ne reflète pas toujours adéquatement la tâche. Il en résulte que le réseau généralise mal.

## 4.6 Algorithme de rétropropagation

L'algorithme qu'on désigne souvent par "Back-propagation" est le plus populaire parmi les deux techniques d'apprentissage des réseaux multicouches. La deuxième technique est l'apprentissage de la machine de Boltzman. L'algorithme de rétropropagation est basé sur la généralisation de la règle de Widrow-Hoff [11] en utilisant une fonction d'activation sigmoïde. Le réseau utilisé est un réseau à couches tel que défini à la section 2.3 où chaque neurone est connecté à l'ensemble des neurones de la couche suivante. Le principe de cet algorithme est la propagation d'un signal provenant des noeuds d'entrées vers la sortie et ensuite on propage l'erreur commise de la sortie vers les couches internes jusqu'à l'entrée [11].

#### 4.6.1 Formulation

Pour un exemple de données à apprendre, on note  $x$  le vecteur d'entrées et  $y_d$  le vecteur de sortie désirée. Supposant que notre réseau a  $n$  noeuds d'entrée et  $m$  neurones de sortie, on a donc :

$$x = (x_1, x_2, \dots, x_n) \text{ et } y_d = (y_{d1}, y_{d2}, \dots, y_{dm}) \quad (4-9)$$

On note  $y = (y_1, y_2, \dots, y_m)$  le vecteur des sorties obtenues effectivement à l'issue de la propagation avant du vecteur  $x$ . On cherche à minimiser l'erreur quadratique entre les sorties désirées et les sorties obtenues, cette erreur étant considérée comme une fonction des poids des connexions:

$$E(W) = \sum_{i=1, m} (y_{di} - y_i)^2 \quad (4-10)$$

La règle de modification des poids à la présentation numéro  $k$  de l'exemple  $x$  est:

$$W_{ij}(k) = W_{ij}(k-1) - e(k) \cdot d_i \cdot o_j \quad (4-11)$$

où  $d_i$  est calculé de proche en proche de la couche de sortie à la couche d'entrée :

$$d_i = 2 \cdot (y_i - y_{di}) \cdot f'(l_i) \text{ pour la couche de sortie} \quad (4-12)$$

$$d_i = \sum_h y_h \cdot W_{hi} \cdot f'(l_i) \text{ pour les couches cachées} \quad (4-13)$$

où  $h$  parcourt les neurones vers lesquels le neurone  $i$  envoie une connexion,  $f$  : est la fonction sigmoïde d'un neurone,  $f'$  est sa dérivée,

$o_j$  : est la sortie du neurone  $j$ ,

$$l_i : \text{ est l'entrée du neurone } i, \quad l_i = \sum_j W_{ij} \cdot o_j \quad (4-14)$$

$e(k)$ : est le pas du gradient à l'étape  $k$ .

#### 4.6.2 Algorithme de Levenberg - Marquardt

Nous retrouvons trois variantes de l'algorithme de rétropropagation disponibles dans le "toolbox" des réseaux de neurones de MATLAB®: "standard backpropagation" et "faster backpropagation" sont deux algorithmes basés sur le gradient descendant et l'algorithme de Levenberg - Marquardt basé sur l'approximation de Newton. La troisième technique est plus puissante que celle du gradient descendant.

L'algorithme de Levenberg - Marquardt converge plus rapidement que les autres algorithmes cités, mais il demande plus de mémoire lorsque le réseau devient grand.

La règle de correction des poids est la suivante [14].

$$\Delta W = (J^T J + \mu I)^{-1} J^T e \quad (4-15)$$

$J$  : la matrice jacobienne des dérivées de l'erreur,

$\mu$  : un scalaire,

$e$  : le vecteur d'erreur.

Lorsque  $\mu$  est grand, l'algorithme se rapproche de la méthode de descente du gradient. Par contre, si  $\mu$  est petit l'algorithme se rapproche de la méthode de Gauss-Newton. La procédure de l'algorithme est la suivante:

- 1- Présentation du vecteur d'entrée en propageant celui-ci jusqu'à la sortie et calcul de l'erreur quadratique,
- 2- Calcul de la matrice jacobienne,
- 3- Calcul de  $\Delta W$  pour corriger les poids du réseau,
- 4- Calcul des nouveaux poids, ajustement du paramètre  $g$  et vérification de la convergence vers l'erreur demandée.

## 4.7 Différentes classes des réseaux de neurones

Chaque architecture définit la forme externe du réseau et contient plusieurs classes spécifiques à une application donnée.

### 4.7.1 Perceptron

Le perceptron est la forme la plus simple d'un réseau de neurones, il modélise la perception visuelle. Il comprend trois principaux éléments: la rétine, les cellules d'association et les cellules de décision. La fonction d'activation utilisée dans ce réseau est de type tout ou rien (0 ou 1). L'apprentissage du perceptron va se faire suivant la règle de Hebb, il n'y a qu'une seule couche de poids modifiables entre les cellules d'association et les cellules de décision. Le perceptron est limité dans ses applications; premièrement, il ne peut être applicable que dans la classification dont les variables sont linéairement séparables et deuxièmement la sortie ne peut être que 0 ou 1 [11].

### 4.7.2 Réseau linéaire

Cette classe de réseaux diffère du perceptron car elle possède un neurone dont la fonction d'activation est linéaire. La règle d'apprentissage utilisée est nommée règle de Widrow-hoff ou delta qui permet d'effectuer une descente de gradient de l'erreur sur une mesure d'erreur quadratique.

Ces techniques d'estimation sont dites des moindres carrés ("Least Mean Squares"). Ce type de réseau est appelé Adaline ("ADaptive LINear Element"). Les domaines d'application comprennent le traitement du signal,

le contrôle et le radar. L'avantage de ce réseau est qu'il converge sur un seul minimum si la solution existe, sinon l'ajout de couches n'a aucun effet. Parmi ces inconvénients, il est limité à une couche de sortie et ne peut résoudre que les problèmes dont la relation entrées/sorties est linéaire [9-15].

#### **4.7.3 Perceptron multicouche**

Cette classe est la plus importante des réseaux de neurones car elle représente la généralisation du perceptron monocouche avec une fonction d'activation de type sigmoïde et une ou plusieurs couches cachées. Le vecteur d'entrée se propage dans le réseau de couche en couche jusqu'à la sortie, l'entraînement de celui-ci se fait avec l'algorithme le plus populaire, soit l'algorithme de rétro-propagation de l'erreur ("error back-propagation algorithme"). Ce réseau est caractérisé par :

- 1- Le modèle du neurone incluant la non linéarité à la sortie (fonction sigmoïde) ;
- 2- Le réseau comporte une ou plusieurs couches cachées ;
- 3- Le plus grand nombre de connexions permettant de résoudre la majorité des problèmes.

Même avec les avantages des couches cachées et la performance de l'algorithme d'apprentissage, il reste plusieurs problèmes non réglés comme le choix du nombre de couches, le nombre de neurones par couche et le problème des minimums locaux où le réseau peut converger [15].

#### 4.7.4 Réseau RBF

Les réseaux RBF ("Radial Basis Function") sont des réseaux à couches qui ont comme origine une technique d'interpolation nommée la méthode d'interpolation RBF. Ce réseau comporte une seule couche cachée dont la fonction d'activation est appelée fonction-noyau ou gaussienne et une couche de sortie avec une fonction d'activation linéaire. La méthode RBF est particulière par ses réponses utiles pour un domaine de valeurs restreint. La réponse de la fonction-noyau est maximale au noyau et décroît généralement de façon monotone avec la distance qui existe entre le vecteur d'entrée et le centre de la fonction-noyau. Afin d'approximer un comportement donné, les fonctions-noyau sont assemblées pour couvrir par leurs champs récepteurs l'ensemble des données d'entrée. Ces fonctions sont ensuite pondérées et la somme de leurs valeurs est calculée pour produire la valeur de sortie. Les réseaux RBF sont capables de calculs puissants. L'apprentissage est plus rapide et plus simple mais demande beaucoup de neurones par rapport aux réseaux multicouches. De plus, ils s'avèrent davantage insensibles à la destruction de leurs poids. Leur domaine d'application est vaste : le traitement d'image, la reconnaissance de forme, les radars et surtout dans les problèmes de classification [12-13-15].

#### 4.7.5 Réseaux récurrents

Cette classe de réseaux se caractérise de trois façons : la non linéarité des unités élémentaires (neurones) du réseau, la symétrie synoptique des

connexions et enfin l'utilisation abondante des boucles dans le réseau. Il y a deux réseaux récurrents importants:

- 1- Réseau de Hopfield: ce réseau est basé sur les principes de la physique statistique qui est fondamentalement une mémoire adressable par son contenu. Les neurones, basés sur le modèle de McCulloch et Pitts, sont tous interconnectés et la règle d'apprentissage utilisée est celle de Hebb. Plusieurs domaines d'application sont possibles, en particulier: les mémoires associatives et l'économie. La principale limitation est qu'il n'y a pas de couches cachées.
- 2- Réseau d' Elman: ce réseau a deux couches dont la première est bouclée (la sortie est reliée à l'entrée à travers un délai). La fonction d'activation est de type tangente-sigmoïde pour la première et linéaire pour la couche de sortie. Ce réseau peut résoudre la plupart des phénomènes temporaires. Son domaine d'application est la détection et l'identification [15].

#### **4.7.6 Réseaux compétitifs**

Ces réseaux utilisent la compétition comme mode d'interaction prédominant; rappelons nous que le phénomène de compétition est le contraire de la coopération. Les neurones sont reliés par des liens inhibiteurs, c'est-à-dire que leur activité est antagoniste, ceci permet notamment de filtrer l'entrée d'informations parasites et facilite ainsi leur traitement. Il y a trois types importants de réseaux compétitifs: le LVQ ("Learning Vector Quantization"), le SOM de Kohonen ("Self-Organizing Map") et Le réseau ART ("Adaptive Resonance Theory"). Leur domaine d'application est la classification [15].

## 4.8 Application des réseaux de neurones dans l'électronique de puissance

Dans le domaine de l'électronique de puissance, les montages génèrent des formes d'ondes de tensions et courants complexes à cause de plusieurs facteurs tels que le mode de fonctionnement, la fréquence du réseau, la fréquence de commutation de la commande, les types de charges à alimenter, etc. Ces montages sont fortement non linéaires. Ceci poussa plusieurs chercheurs à trouver quelques applications des réseaux de neurones pour contrôler et estimer les paramètres électriques des convertisseurs et les paramètres mécaniques des machines électriques de l'électronique de puissance.

Les premières applications se trouvaient dans la commande des machines électriques, comme le réglage de la vitesse, du couple et la commande vectorielle [16]. Ensuite, quelques applications ont vu le jour dans le contrôle des formes d'ondes comme le courant de source, la tension de sortie des onduleurs et la correction du facteur de puissance, ainsi que l'estimation des formes d'ondes pour le diagnostic de ces montages [9].

### 4.8.1 Régulateur statique

Le régulateur statique est un modèle inverse qui ne prend pas en considération la dynamique du système. Le réseau de neurones établira une relation entre les paramètres d'entrées et de sorties pour simuler l'inverse du système réel et permettre de le linéariser. Ainsi, on obtiendra l'identité formée par les systèmes inverses et réel à contrôler. La banque d'apprentissage doit comprendre toute la plage de fonctionnement du système à commander afin

permettre une bonne généralisation et un bon fonctionnement du réglage du système.

#### **4.8.2 Régulateur dynamique**

Ce type de régulateur est plus performant que le modèle inverse statique car il permet une amélioration de la réponse dynamique du système. Deux méthodes sont utilisées pour concevoir un régulateur à réseau de neurones dynamiques, la méthode directe ou la méthode indirecte, cette dernière est la plus utilisée, elle comprend deux étapes: la première est de concevoir un modèle à l'aide d'un réseau de neurones du système réel et la seconde est d'entraîner le régulateur à l'aide de celui-ci. Pour l'apprentissage de la dynamique, il n'y a pas de méthode systématique qui permet de déterminer le nombre de retards ou les paramètres à boucles, que ce soit pour le modèle ou pour le régulateur.

#### **4.8.3 Commande des machines électriques**

A cause de la non linéarité des machines électriques et de leurs constantes de temps qui sont relativement plus grandes que celles des convertisseurs, les premières applications des réseaux de neurones ont vu le jour dans ce domaine [6]. Pour que le fonctionnement d'un système d'entraînement à vitesse variable soit robuste et stable (stabilité mécanique et électrique), la machine doit fonctionner dans le régime linéaire (sans saturation de flux). L'avantage des réseaux de neurones est de permettre le réglage de la vitesse

même s'il y a une variation de paramètres non accessibles de la machine comme la variation de la résistance interne. En outre, des applications de type commande vectorielle des machines à induction sont sur le point de se développer.

#### 4.9 Conclusion

Ce chapitre a porté sur les fondements des réseaux de neurones artificiels en commençant par la formulation mathématique d'un neurone. Ensuite, on a mentionné le principe biologique des réseaux de neurones, leur architecture, leurs classes, de même l'apprentissage et les problèmes liés à leur utilisation ont été soulignés. Enfin quelques applications qui ont vu le jour en électronique de puissance. Il paraît, que jusqu'à aujourd'hui, l'entraînement des réseaux de neurones demeure encore plutôt un art plus qu'une science. Ceci constituera la base de nos applications des RNA, parmi une vaste gamme désormais possible, dans l'électronique de puissance.

# *Chapitre 5*

## *Adaptation de la constante de temps rotorique dans la commande vectorielle*

### **5.1 Introduction**

Le contrôle vectoriel à orientation du flux rotorique reste le plus répandu vu sa simplicité d'implantation. Mais ce contrôle est basé sur le modèle inverse du moteur et il est très sensible aux variations des paramètres. Le découplage est maintenu uniquement si les paramètres réels de la machine sont égaux à ceux programmés dans le contrôleur. La constante de temps rotorique joue un rôle dominant dans le réglage du régulateur. Ce paramètre varie en fonction de la température et du niveau de saturation dans la machine ce qui peut entraîner une perte de découplage entre le flux et le couple. Il s'avère nécessaire d'estimer la constante de temps rotorique réelle du moteur et d'annuler l'erreur entre la valeur estimée et la valeur nominale programmée dans le contrôleur.

## 5.2 Détermination de la constante de temps rotorique réelle du moteur

L'étude précédente nous a permis de conclure que la variation de la constante de temps rotorique est exclusivement due à la résistance rotorique. Comme la machine fonctionne à flux constant, on considère que les inductances sont constantes. Les variables notées par une étoile représentent les commandes et  $\tau_r^*$  la valeur de commande (nominale) de la constante de temps rotorique.

Dans la commande à flux orienté les conditions de découplage sont données par [16]:

$$i_{ds}^* = \frac{1 + s\tau_r^*}{M} \cdot \Psi_r^* = I_{mr} \cdot [1 + s\tau_r^*] \quad (5-1)$$

$$i_{qs}^* = \frac{2 \cdot L_r}{3 \cdot pp \cdot M \cdot \Psi_r^*} \cdot T_{em}^* \quad (5-2)$$

$$\omega_{sl}^* = \frac{M \cdot i_{qs}^*}{\tau_r^* \cdot \Psi_r^*} \quad (5-3)$$

La fréquence angulaire du stator est donnée par le contrôleur vectoriel en régime permanent comme suit :

$$\omega_e^* = pp \cdot \omega_{mec} + \frac{i_{qs}^* \cdot M}{\tau_r^* \cdot \Psi_r^*} = pp \cdot \omega_{mec} + \frac{i_{qs} \cdot M}{\tau_r \cdot \Psi_r} \quad (5-4)$$

où :

$i_{qs}$ ,  $\tau_r$  et  $\Psi_r$  sont les valeurs réelles dans le moteur,  $i_{qs}^*$ ,  $\tau_r^*$  et  $\Psi_r^*$  sont les valeurs de commande calculées par le contrôleur et  $\omega_{mec}$  est la vitesse du moteur mesurée directement.

À partir de l'équation (5-4), il est évident que :

$$\frac{i_{qs}^* \cdot M}{\tau_r^* \cdot \Psi_r^*} = \frac{i_{qs} \cdot M}{\tau_r \cdot \Psi_r} \quad \text{ou} \quad \frac{i_{qs}^*}{\tau_r^* \cdot i_{mr}^*} = \frac{i_{qs}}{\tau_r \cdot i_{mr}} \quad (5-5)$$

et que :

$$\tau_r = \frac{i_{qs}}{i_{mr}} \left[ \frac{\tau_r^* \cdot i_{mr}^*}{i_{qs}^*} \right] \quad (5-6)$$

on peut aussi montrer que :

$$i_s = i_{ds}^2 + i_{qs}^2 = (i_{ds}^*)^2 + (i_{qs}^*)^2 \quad (5-7)$$

$$i_{mr} = \frac{\Psi_r}{M} \quad (5-8)$$

$$i_{mr}^* = i_{ds}^* \cdot \frac{1}{1 + s \tau_r^*} \quad (5-9)$$

Pour un flux de commande  $\Psi_r^*$  constant on a :

$$i_{ds}^* = \frac{\Psi_r^*}{M} \quad (5-10)$$

$$i_{qs} = \sqrt{i_s^2 - \left( \frac{\Psi_r}{M} \right)^2} \quad (5-11)$$

En remplaçant (5-8), (5-9), (5-10) et (5-11) dans (5-6) on aura :

$$\tau_r = \left[ \frac{\tau_r^* \cdot \Psi_r^*}{M \cdot i_{qs}^*} \right] \cdot \frac{1}{1 + s \tau_r^*} \cdot \frac{\sqrt{(Mi_s)^2 - \Psi_r^2}}{\Psi_r} \quad (5-12)$$

à partir de l'équation (5-12) on peut voir que l'information sur la constante de temps réelle du moteur est donnée par l'amplitude du flux rotorique.

### 5.3 Adaptation de la constante de temps rotorique par les réseaux de neurones

Pour garder les hautes performances du contrôle vectoriel, la constante de temps rotorique programmée dans le contrôleur doit rester égale à constante de temps rotorique réelle dans le moteur (estimée). Un réseau de neurones assure en permanence l'estimation de la constante de temps réelle dans la machine et injecte cette valeur dans le contrôleur pour permettre le calcul des consignes. Le principe de la régulation est donné par la figure 5.1.

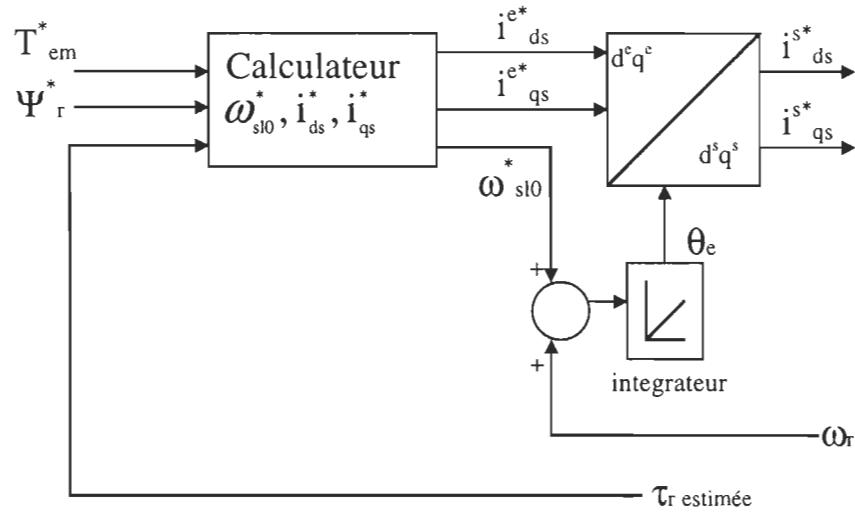


Figure 5.1. Principe d'adaptation de la constante de temps rotorique.

Ce schéma de contrôle est utilisé pour adapter la constante de temps rotorique dans le contrôle vectoriel de la machine à induction. Un réseau de neurones est entraîné pour estimer la constante de temps rotorique en utilisant les tensions, les courants et la fréquence angulaire statoriques ( $v_{as}, v_{bs}, i_{as}, i_{qs}, \omega_e$ ), ce qui nous permet d'avoir un estimateur indépendant des paramètres de la machine [17].

### 5.3.1 Choix du réseau de neurones

La notion des réseaux à couches s'applique à la majorité des types de réseaux connus aujourd'hui. La plupart des réseaux proposés plus récemment sont structurés en couches comme le perceptron, l'Adaline, etc. Le développement du perceptron multicouches marqua un grand tournant dans la recherche connexioniste. C'est un réseau qui contient au moins une couche cachée et il est capable de faire l'approximation de toute fonction continue. L'estimateur de la constante de temps rotorique n'a pas besoin de comportement dans le temps car l'apprentissage se fait par des données en régime permanent [10,13]. Tous ces facteurs nous amènent à choisir le réseau perceptron multicouche. La figure 5.2 montre le schéma d'un réseau de neurones en blocs Simulink.

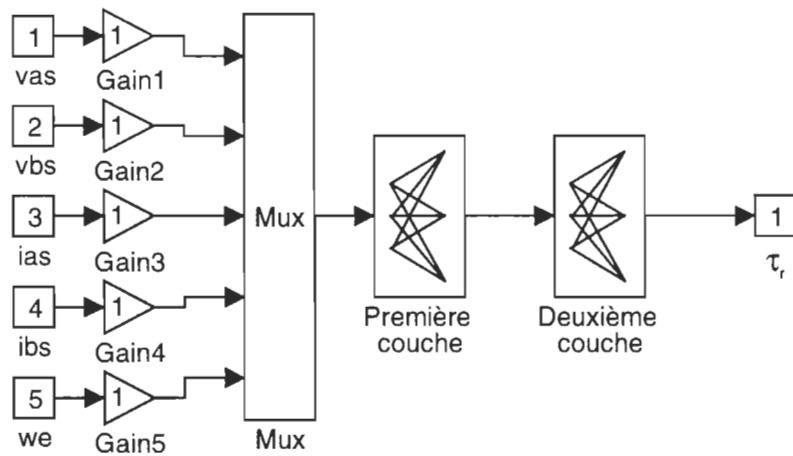


Figure 5.2. Réseau de neurones en blocs Simulink

### 5.3.2 Apprentissage du réseau

L'apprentissage utilisé dans cette application est appelé apprentissage supervisé. On présente au réseau un vecteur d'entrée et un autre de sortie désiré vers lequel l'apprentissage tente de faire converger le réseau.

réseau comporte quelques dizaines de poids mais il occupe une grande zone mémoire. Il y a huit paramètres à choisir pour cet algorithme. En général, on ne choisit que les trois premiers qui sont le pas d'affichage, le nombre d'itérations et l'erreur quadratique globale et on prend le reste des paramètres par défaut [15].

#### **5.3.2.4 Procédure d'apprentissage**

Dans ce paragraphe, on donne la structure d'un programme d'apprentissage supervisé disponible dans l'environnement MATLAB® :

- chargement des données d'apprentissage,
- normalisation des données par rapport à la valeur maximale,
- choix du nombre de neurones,
- initialisation des poids de façon aléatoire,
- choix des paramètres de l'algorithme d'apprentissage Trainlm
- appel de l'algorithme Trainlm [15],
- sauvegarde de la matrice des poids.

Si l'erreur globale est inférieure à un seuil, on aura l'arrêt de l'apprentissage sinon on continue l'apprentissage en prenant les poids de la dernière itération. L'initialisation des poids est effectuée une seule fois au début de l'apprentissage.

#### **5.3.3 Réseau estimateur de la constante de temps rotorique**

Un réseau de neurones est entraîné pour estimer la constante de temps rotorique en utilisant les tensions, les courants et la fréquence angulaire

statoriques ( $v_{as}, v_{bs}, i_{as}, i_{bs}, \omega_e$ ). Les entrées/sorties du réseau sont générées par le modèle de la machine à induction contrôlée selon la méthode du flux orienté dans l'environnement Simulink. Dans ce modèle, on programme un changement de la résistance rotorique par pas successifs.

Donc, le réseau possédera 5 entrées et une seule sortie ( $\tau_{re}$ ). La préparation des données est faite suivant la méthodologie décrite précédemment.

La phase d'apprentissage est faite en utilisant le "Neural Network Toolbox" de Matlab®. L'erreur quadratique globale est fixée à  $10^{-2}$ . Une couche cachée de 8 neurones était suffisante pour que l'erreur globale converge vers l'erreur désirée après 160 000 itérations. La constante de temps rotorique programmée dans le moteur est variable par échelon.

La constante de temps rotorique estimée se superpose sur celle programmée dans le moteur (figure 5.4) avec une certaine erreur qu'on a fixée au début de l'apprentissage.

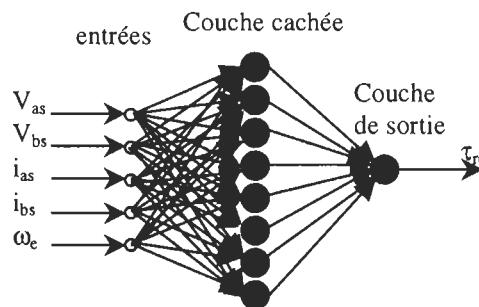


Figure 5.3. Réseau de neurones estimateur de la constante de temps rotorique

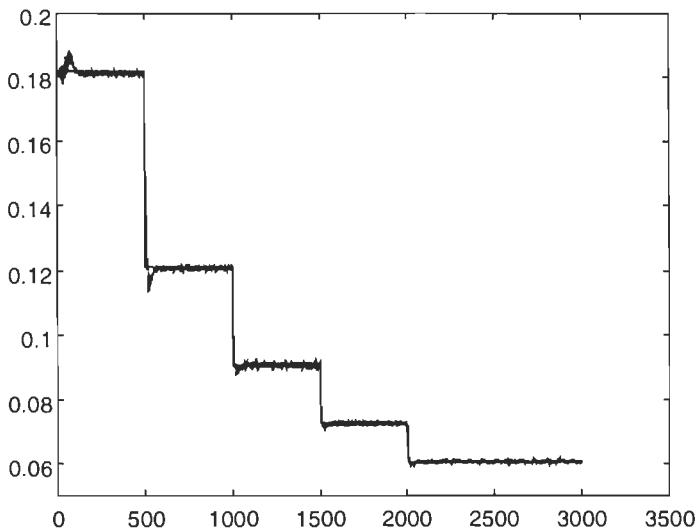


Figure 5.4 Résultat d'apprentissage

## 5.4 Résultats de simulations

### 5.4.1 Variation de $\tau_r$ par échelon

Les résultats de simulation ont été effectués pour une commande de couple de valeur 10 N.m et une commande de flux de 0.55 Wb. A l'instant  $t = 0s$ , la résistance rotorique programmée dans le modèle du moteur est égale à sa valeur nominale. Un pas de variation de 150% est appliqué à  $t = 4s$  et à  $t = 8s$  comme le montre la figure 5.5.

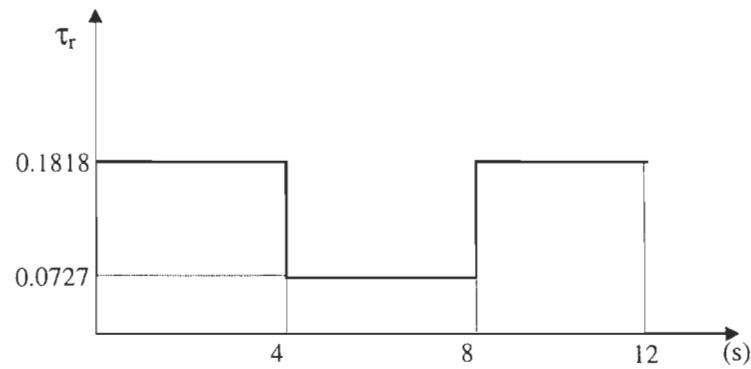


Figure 5.5. Variation programmée de la constante de temps rotorique.

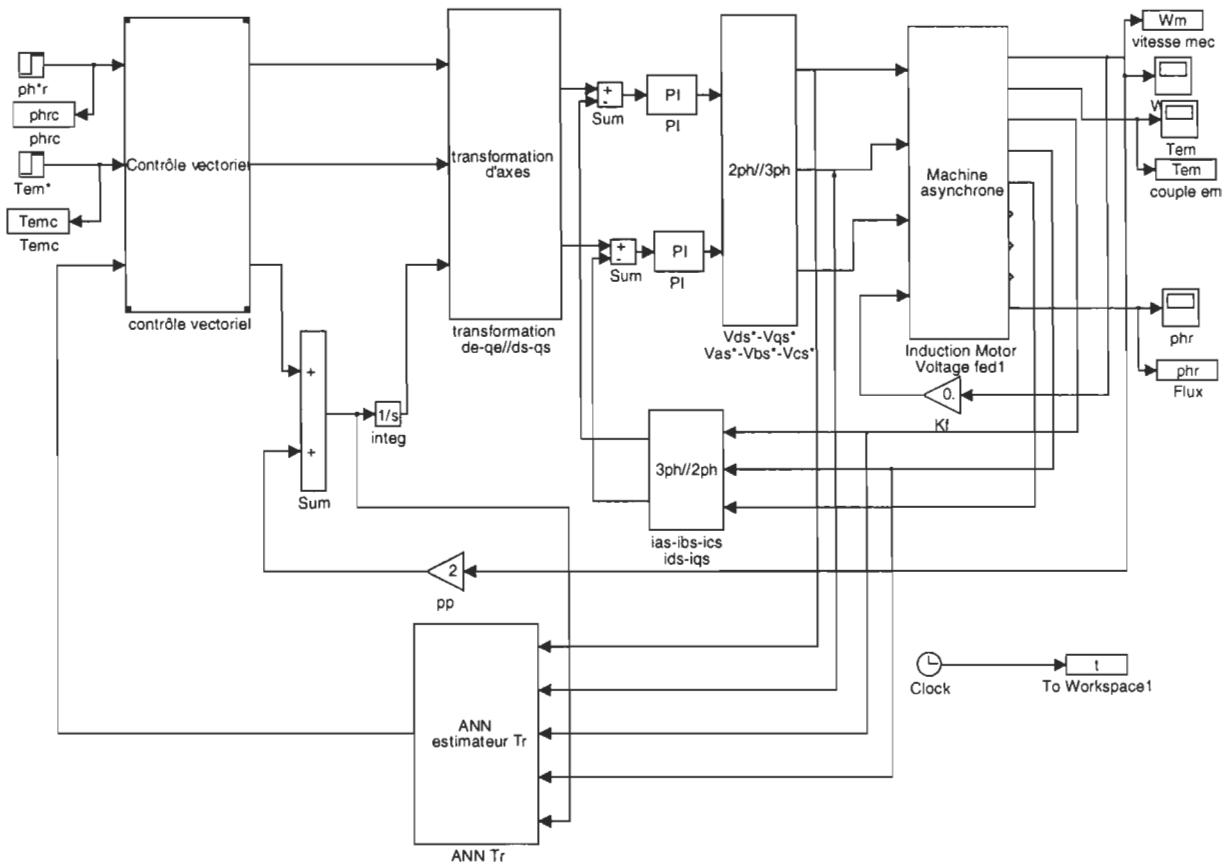


Figure 5.6. Modèle Simulink du contrôle vectoriel avec adaptation de  $\tau_r$

La figure 5.6 montre le modèle Simulink du contrôle vectoriel du moteur avec adaptation de  $\tau_r$ . Le bloc "ANN Tr" représente le réseau de neurones qui

va estimer la constante de temps rotorique. On y a programmé les poids et les seuils obtenus après la phase d'apprentissage, on peut remarquer aussi le régulateur PI utilisé pour corriger l'erreur entre la constante de temps rotorique estimée et sa valeur nominale.

On peut remarquer que la constante de temps rotrorique dans le contrôleur est maintenue égale à la constante de temps réelle dans la machine via l'estimateur à réseau de neurone. Les variations du flux (figure 5.7) et du couple (figure 5.8) sont compensés grâce à l'adaptation.

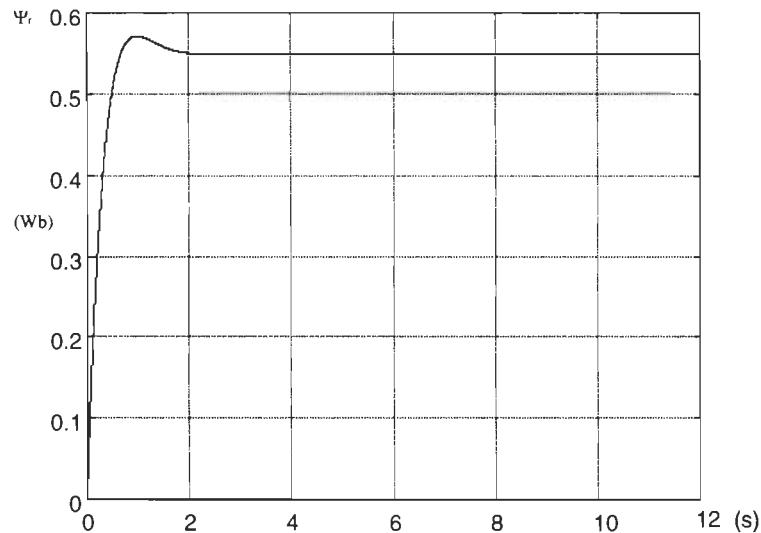


Figure 5.7. Réponse du flux avec adaptation

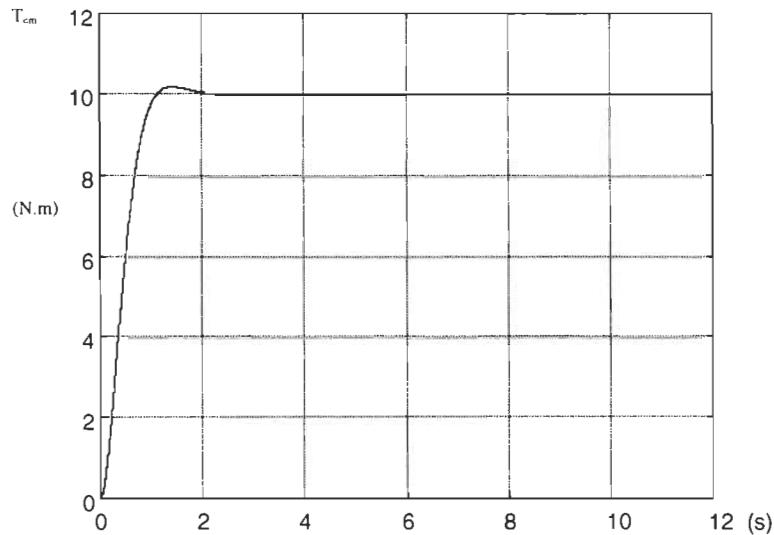


Figure 5.8. Réponse du couple avec adaptation

#### 5.4.2 Variation exponentielle de $\tau_r$

Maintenant, on va simuler un cas proche de la réalité où  $\tau_r$  diminue exponentiellement. On suppose toujours que la variation de  $\tau_r$  est due à la variation de  $R_r$  et que celle-ci suit la loi [19]:

$$R_r = R_r^* (2.1 - 1.5 e^{-0.1t})$$

où  $R_r^*$  est la résistance rotorique nominale de la machine.

Les figures 5.10 et 5.11 indiquent que dans le contrôle vectoriel conventionnel, lorsque  $\tau_r$  change, le niveau du flux et du couple changent aussi par rapport à leurs valeurs désirées. Mais lorsqu'il y a adaptation de la constante de temps rotorique, le flux et le couple restent pratiquement constants et ne s'éloignent pas de leurs consignes.

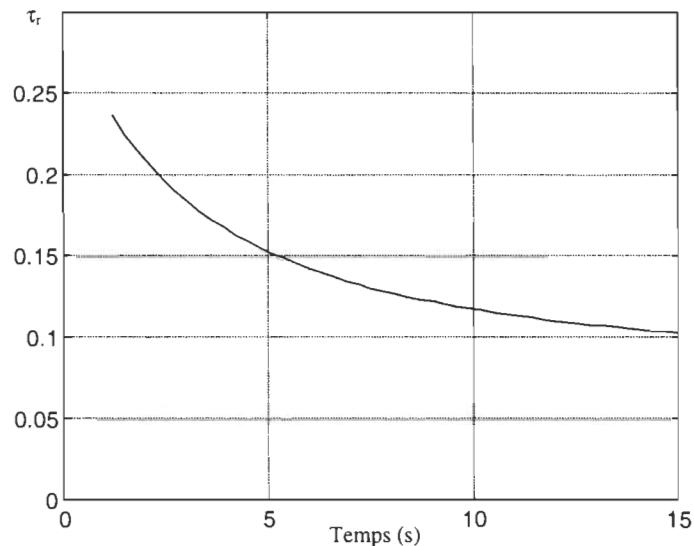


Figure 5.9 Variation exponentielle de la constante de temps rotorique

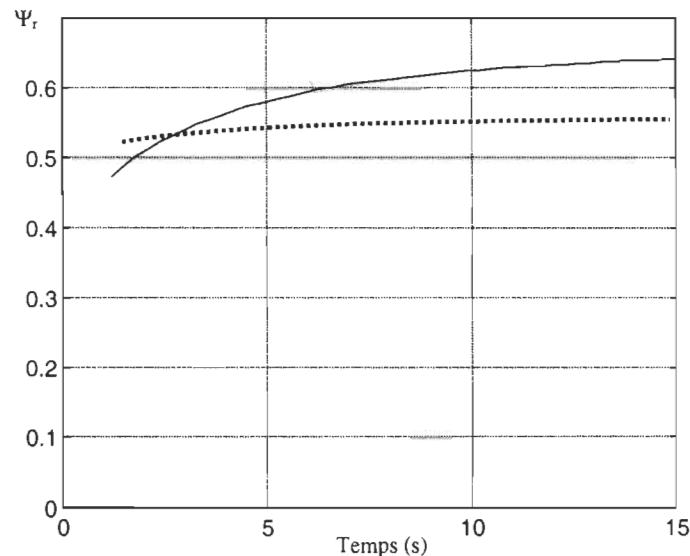


Figure 5.10 — flux rotorique sans adaptation  
 --- flux rotorique avec adaptation

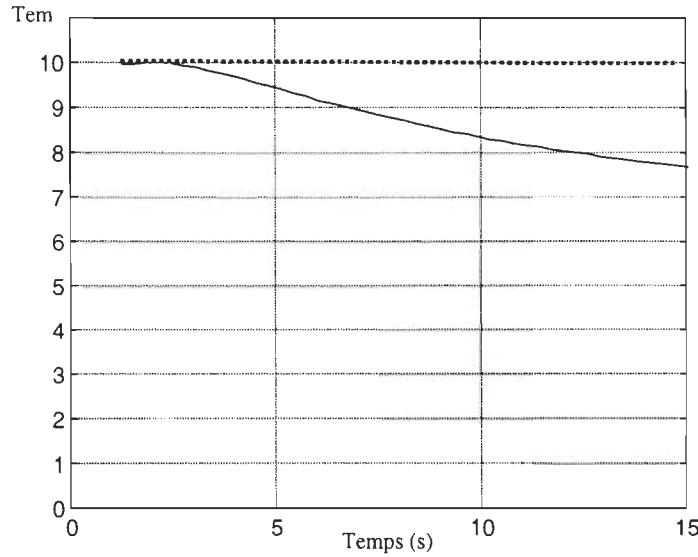


Figure 5.11 — Couple sans adaptation  
---- Couple avec adaptation

## 5.5 Estimateur basé sur le filtre de Kalman

Le filtre de Kalman étendu est très utilisé pour identifier les paramètres qui varient dans les commandes adaptatives[19]. Le suivi de l'évolution de la constante de temps rotorique constitue une application importante.

L'article Zai et Lipo [18] présente une estimation de la constante de temps rotorique implantée dans une commande vectorielle indirecte d'un moteur à induction où le flux rotorique est maintenu constant sur l'axe d du repère dq synchrone lié au champ tournant.

Ils ont choisi, dans le repère dq lié au stator, les données suivantes :

$$\text{le vecteur d'état } X = [i_{qs} \ i_{ds} \ v_{qr} \ v_{ds} \ \frac{R_r}{L_r}]^T, \quad (5-13)$$

$$\text{l'entrée } U = \begin{bmatrix} v_{qs} \\ v_{ds} \end{bmatrix} \quad (5-14)$$

$$\text{et la sortie } Y = \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} \quad (5-15)$$

Le système, en temps continu, est donné par les équations non linéaires suivantes:

$$\begin{cases} \frac{dX}{dt} = f(X, U, t) + G(t) W_X(t) \\ Y = h(X, t) + V \end{cases} \quad (5-16)$$

avec :

$$f(X, U, t) = \sigma \cdot \begin{pmatrix} -\frac{R_s}{L_s}x_1 - \frac{L_m^2}{L_s L_r} \omega_e x_2 + \frac{L_m}{L_s} x_3 x_5 - \frac{L_m}{L_s} \omega_e x_4 - \frac{1}{L_s} u_1 \\ \frac{L_m^2}{L_s L_r} \omega_e x_1 - \frac{R_s}{L_s} x_2 + \frac{L_m}{L_s} \omega_e x_3 + \frac{L_m}{L_s} x_4 x_5 + \frac{1}{L_s} u_2 \\ \frac{L_m}{L_r} \omega_e x_1 + \frac{L_m}{L_r} \omega_e x_2 - x_3 x_5 + \omega_e x_4 - \frac{L_m}{L_s L_r} u_1 \\ -\frac{L_m}{L_r} \omega_e + \frac{L_m}{L_r} \left( \frac{R_s}{L_s} \right) x_2 - \omega_e x_3 - x_4 x_5 - \frac{L_m}{L_s L_r} u_2 \\ 0 \end{pmatrix} \quad (5-17)$$

$$\text{où : } X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T \quad (5-18)$$

$$\text{et } G(t) = \sigma \cdot \begin{pmatrix} \frac{1}{L_s} & 0 & 0 \\ 0 & \frac{1}{L_s} & 0 \\ -\frac{1}{L_s} \frac{L_m}{L_r} & 0 & 0 \\ 0 & -\frac{1}{L_s} \frac{L_m}{L_r} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5-19)$$

Le bruit  $W_X$  représente les harmoniques de tension et l'incertitude paramétrique. Le paramètre  $V$  modélise les harmoniques de courant. Les matrices  $F$  et  $H$  obtenues sont :

$$F = \sigma \begin{pmatrix} -\frac{R_s}{L_s} & -\frac{L_m^2}{L_s L_r} \omega_e & \frac{L_m}{L_r} \left( \frac{R_r}{L_r} \right) & -\frac{L_m}{L_s} \omega_e & -\frac{L_m}{L_s} i_{qr} \\ -\frac{L_m^2}{L_s L_r} \omega_e & -\frac{R_s}{L_s} & \frac{L_m}{L_s} \omega_m & \frac{L_m}{L_r} \left( \frac{R_r}{L_r} \right) & \frac{L_m}{L_s} i_{dr} \\ \frac{L_m}{L_r} \left( \frac{R_s}{L_s} \right) & \frac{L_m}{L_r} \omega_e & -\frac{R_r}{L_r} & \omega_m & -i_{qr} \\ -\frac{L_m}{L_r} \omega_e & \frac{L_m}{L_r} \left( \frac{R_s}{L_s} \right) & -\omega_e & -\frac{R_r}{L_r} & -i_{dr} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5-20)$$

$$\text{et } H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (5-21)$$

Bien qu'il soit l'algorithme le plus utilisé, sa critique est qu'il nécessite l'évaluation en ligne de l'inductance magnétisante  $L_m$ , la matrice  $F$  dépend sensiblement de ce paramètre ainsi que de  $R_s$ .

Si les paramètres du moteur utilisés dans la matrice  $F$  ne sont pas les mêmes que ceux du moteur, il y aura une dégradation du comportement de l'estimateur. La robustesse de l'estimateur basé sur le filtre de Kalman est examinée en faisant varier  $L_m$  et  $R_s$  (figure 5.12).

Les résultats présentés dans l'article (figure 5.12) montrent que l'estimateur basé sur le filtre de Kalman devient instable lorsque  $L_m$  varie de plus de 10% de sa valeur nominale.

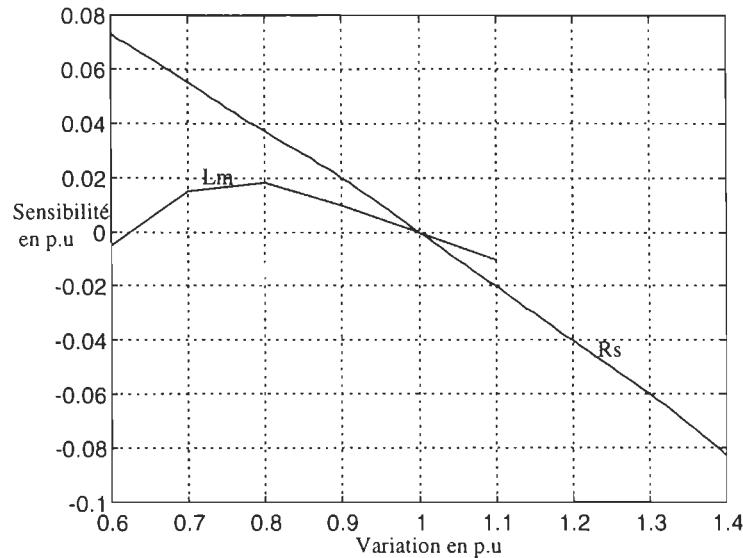


Figure 5.12 Sensibilité de l'estimateur basé sur le filtre de Kalman [18].

## 5.6 Conclusion

En comparant les deux méthodes présentées dans ce chapitre on peut tirer les remarques suivantes :

- La méthode d'estimation basée sur le filtre de Kalman dépend d'autres paramètres ce qui nécessite une autre adaptation pour que l'estimateur soit stable.
- La méthode d'estimation basée sur les réseaux de neurones ne dépend d'aucun paramètre de la machine. Une fois la phase d'apprentissage est terminée, le réseau n'effectue que des calculs simples (produit, somme et la fonction d'activation) pour estimer la constante de temps rotorique.

# *Chapitre 6*

## *Implantation pratique de la méthode sur DSP TMS320C31*

### **6.1. Introduction**

Plusieurs concepteurs utilisent le processeur numérique de signaux ou DSP pour résoudre les problèmes survenus dans des applications de contrôle. Aujourd’hui le DSP met en évidence pratique plusieurs applications qui étaient difficiles à implanter ou coûtaient trop chère.

Comme le prix du DSP a diminué, ces processeurs ont rapidement remplacé les microcontrôleurs et les composants analogiques dans plusieurs applications de contrôle.

A titre d’illustration citons quelques domaines où le DSP a réduit les prix des applications : le servo-contrôle, les alimentations à découpage [24], le contrôle numérique des machines et la robotique.

Dans les entraînements à vitesse variable, le DSP est plus souvent utilisé dans le contrôle vectoriel. La facilité de la maintenance des machines à induction leur a permis de remplacer les machines à courant continu, cependant elles sont très complexes dans leurs structures de contrôle à cause du couplage entre les trois phases.

## 6.2 Considérations pratiques

Plusieurs configurations matérielles peuvent être utilisées pour l'implantation des méthodes de contrôle utilisant les réseaux de neurones. Les plus rapides sont réalisées par des circuits intégrés spécialisés et dédiés aux réseaux de neurones. Une autre alternative peut être réalisée par un système de développement à base de DSP très rapide.

Une solution qui coûte moins chère est basée sur Le DSP TMS320C31 de Texas Instrument. Ce dernier inclus toutes les fonctions pour l'implantation en temps réel comme les cartes d'acquisition, une sortie logique PWM et une entrée pour encodeur.

## 6.3 Implantation sur DSP TMS320C31

Un banc d'essai pour entraînement à vitesse variable, disponible au laboratoire de recherche en électronique de puissance, nous a facilité la mise en œuvre pratique de l'adaptation de la constante de temps rotorique à l'aide de réseaux de neurones .

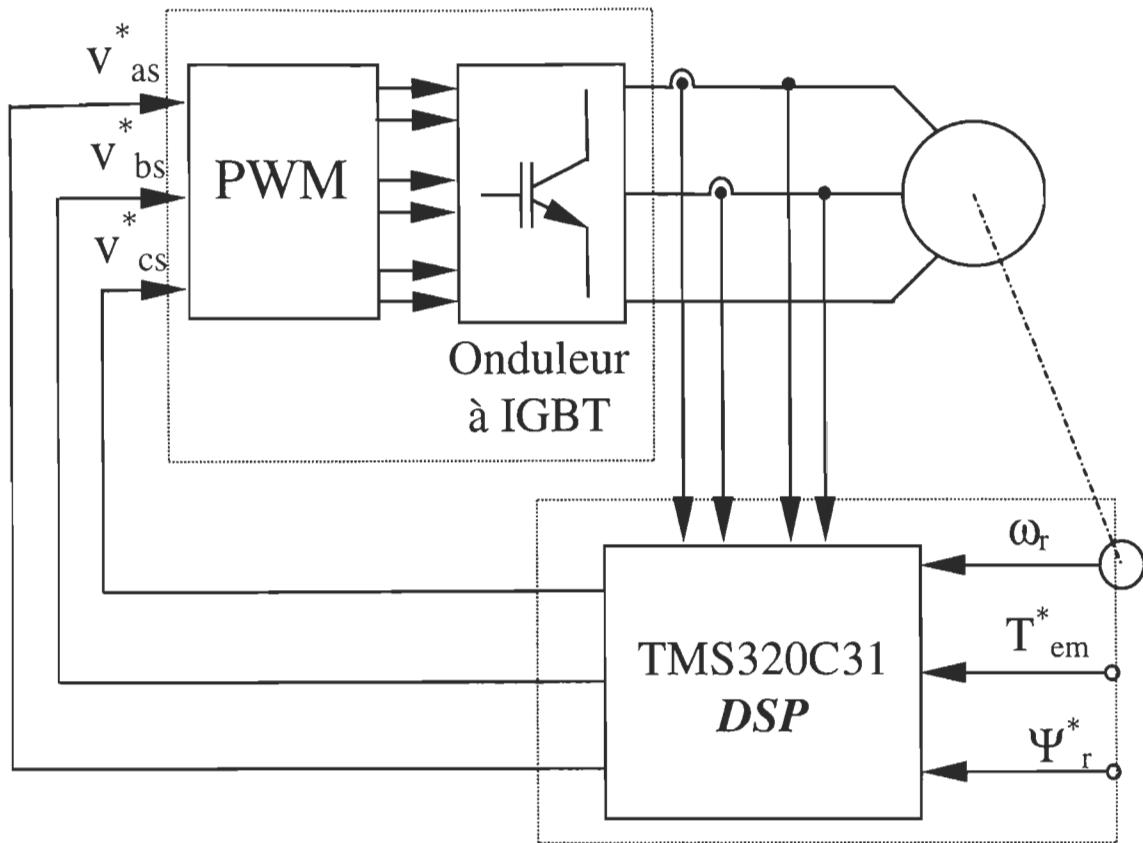


Figure 6.1. Montage pratique.

Le montage pratique comporte un onduleur à transistors à grille isolée (IGBT) fonctionnant à 10 kHz, contrôlé par la technique PWM, la machine à induction à contrôler et le DSP TMS320C31.

La configuration présentée par la figure 6.2 utilise six convertisseurs analogique/numérique pour les entrées et trois convertisseurs numérique/analogique pour les sorties. Les six entrées sont les tensions (a,b), les courants (a,b), la vitesse mécanique et la consigne du couple. Dans ce modèle, le flux a été fixé à sa valeur nominale (0.55Wb).

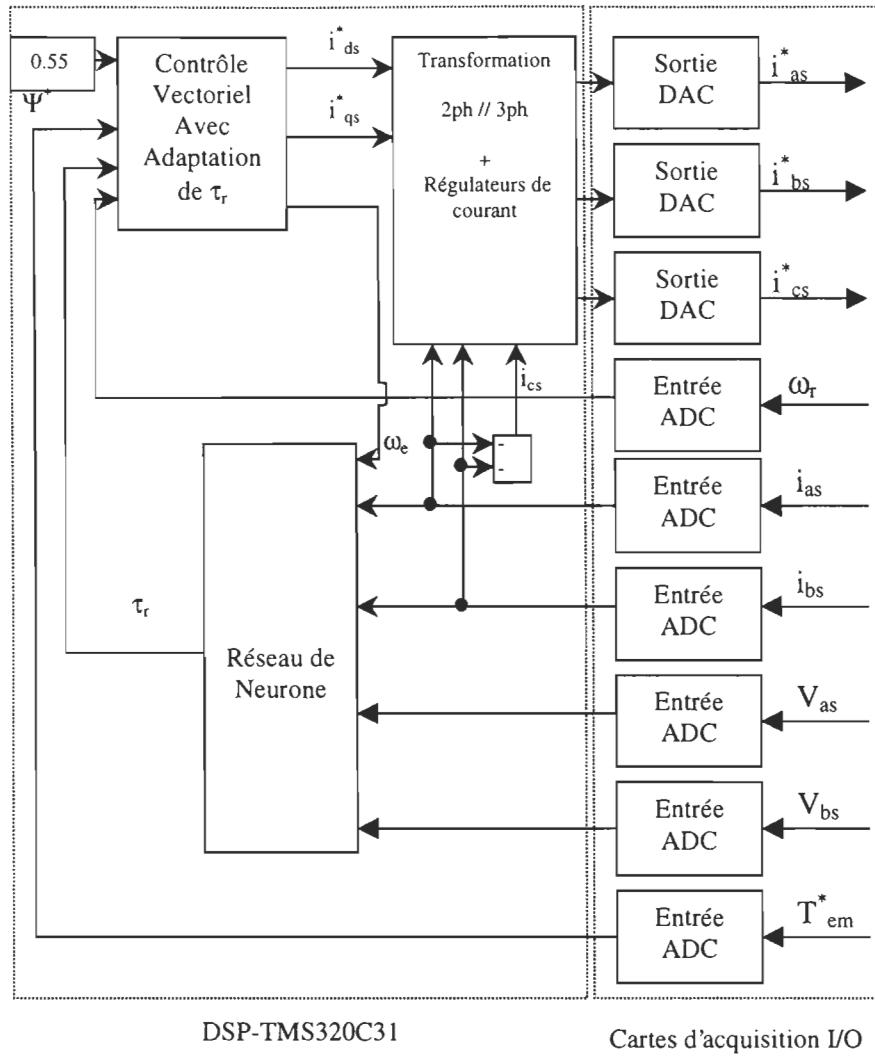


Figure 6.2. Schéma de la réalisation sur DSP

La programmation se fait en utilisant le language C plus des instructions spéciales. Le logiciel Winboss [23] permet de gérer les communications entre le PC et le DSP TMS320C31. Une fois le programme compilé, le fichier (.out) sera chargé dans le TMS320C31 pour nous générer les consignes à envoyer au circuit PWM.

## 6.4 Résultats pratiques

La mise en évidence de la méthode proposée sur DSP TSM320C31 nous permet de comprendre de plus près le problème de la constante de temps rotorique dans le contrôle vectoriel indirect. Comme celle-ci varie très lentement, donc elle prend du temps pour varier considérablement. Pour cela on variera la constante de temps rotorique programmée dans le modèle dq inverse (contrôleur) puisqu'elle est accessible par le code de la programmation du DSP TMS320C31.

### 6.4.1 Effet pratique de $\tau_r$

Sans l'adaptation et au début de l'entraînement, la constante de temps rotorique dans le contrôleur est programmée à la même valeur que de celle du moteur. La figure 6.3 représente la réponse du flux pour une consigne de couple de 10 N.m

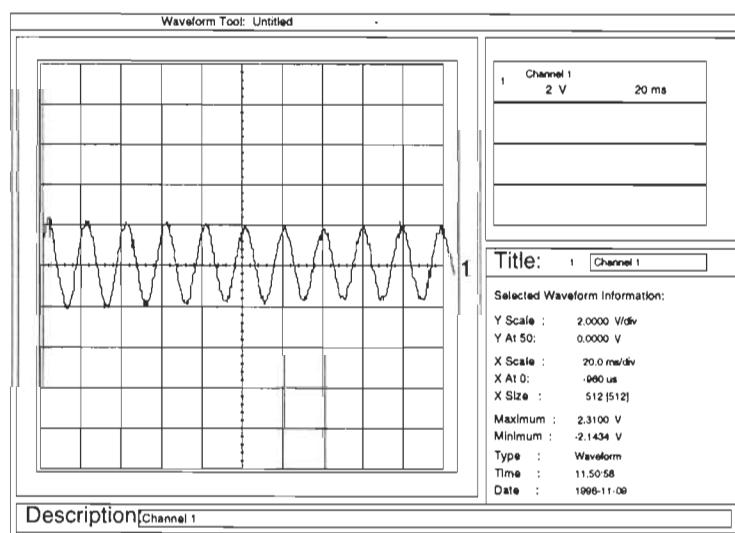


Figure 6.3. Niveau du flux rotorique dans la machine

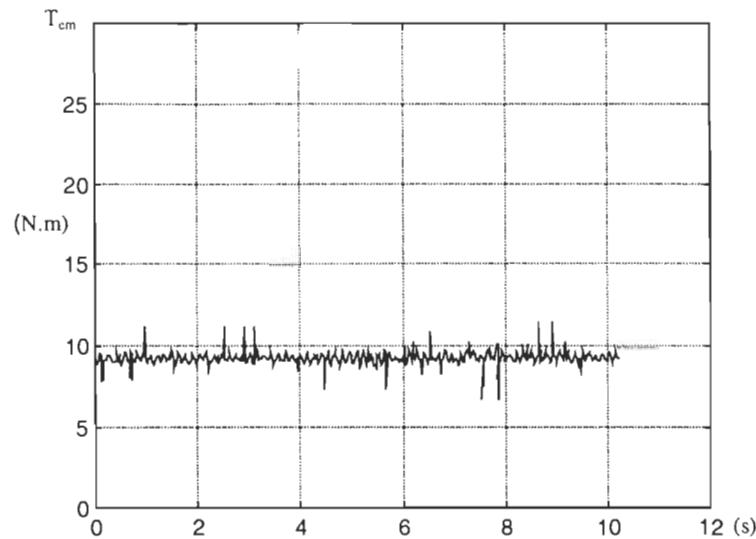


Figure 6.4. Réponse du couple

Maintenant, on réduit la constante de temps rotorique programmée dans le modèle dq inverse à 30%, ce qui donne une augmentation de 100% du niveau du flux dans la machine et une diminution de 17% du couple. Comme il est montré par la figure 6.5, le niveau du flux a augmenté jusqu'à la saturation de la machine.

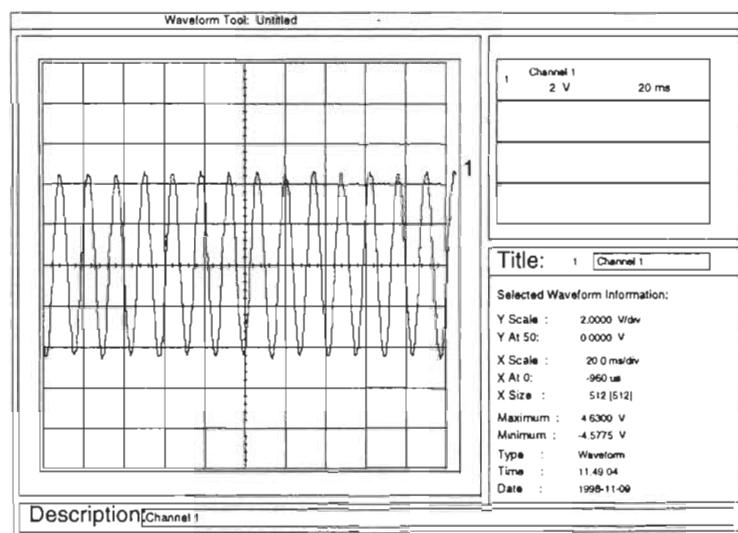


Figure 6.5. Niveau du flux dans la machine pour une réduction de 25% de  $\tau_r$

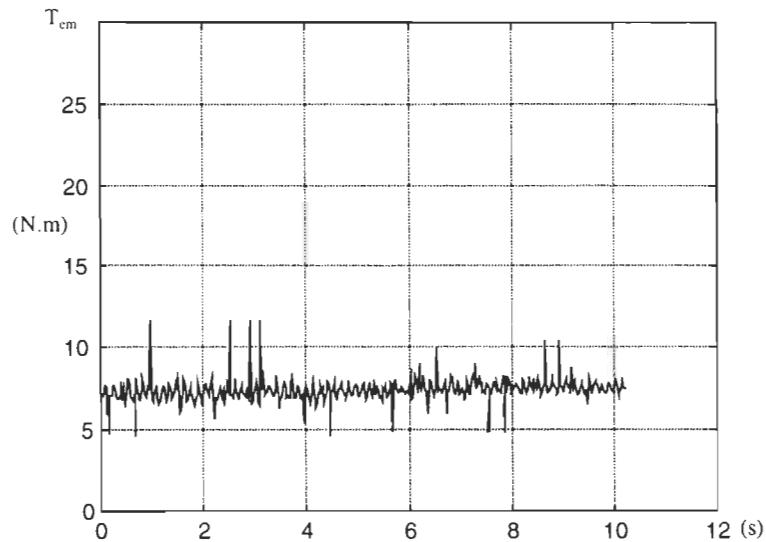


Figure 6.6. Réponse du couple pour une réduction de 25% de  $\tau_r$

Ces résultats sont en bonne concordance avec ceux obtenus en simulation. Avec la même variation (de 25%) de  $\tau_r$ , on obtient presque les mêmes pourcentage de variation pour le flux et le couple.

#### 6.4.1 Efficacité de la méthode d'adaptation

Pour une consigne de 10 N.m et un niveau de flux de 0.55 Wb on lance l'entraînement pendant plus de huit heures avec la méthode d'adaptation de la constante de temps rotorique proposée, on constate que le flux garde le même niveau et le couple ne change pas et ils auront les mêmes allures que les figures 6.3 et 6.4

## 6.5 Conclusion

La méthode d'adaptation de la constante de temps rotorique proposée a été expérimentalement testée. La librairie "Neuralib" nous a facilité l'implantation du réseau de neurones. Les résultats pratiques nous ont rapproché d'avantage du problème de l'effet de la constante de temps rotorique sur le contrôle. De plus on a pu confirmer la fiabilité de la méthode d'adaptation durant plusieurs jours d'entraînement.

# CONCLUSION

Dans ce mémoire, une technique d'orientation du flux rotorique, avec adaptation de la constante de temps rotorique, basée sur les réseaux de neurones a été présentée. Les résultats de simulation et de pratique nous ont confirmé la faisabilité de cette méthode qui peut être facilement appliquée à d'autres structures de la commande vectorielle telles que l'orientation du flux statorique et l'orientation du flux d'entrefer.

Un grand avantage des réseaux de neurones est qu'ils s'implantent facilement en simulation dans l'environnement SIMULINK et à l'aide de la boîte d'outils "Neural Networks". De plus, la mise en oeuvre expérimentale sur analyseur de signaux numériques "DSP" est relativement simple et ne présente aucune difficulté particulière. Les essais expérimentaux ont permis de mettre en évidence la fiabilité et la robustesse des réseaux développés. Toutefois, il faudrait mentionner que l'absence de règles systématiques pour le choix du nombre de neurones et du nombre de couches cachées nous impose une étape de tâtonnements afin d'arriver à une structure fiable du réseau. Actuellement, ceci présente un inconvénient des réseaux neuroniques car la période de développement de la structure peut nécessiter plus ou moins de temps et d'essais.

Les résultats obtenus lors de l'expérimentation sont très prometteurs ceci nous amène à conclure que le recours aux réseaux neuroniques, pour les applications industrielles, connaîtra sûrement une forte croissance dans les prochaines années.

## BIBLIOGRAPHIE

- [1] L.C. Zai & T.A. Lipo, " An Extended Kalman Filter Approach to Rotor Time Constant Measurement in PWM Induction Motor Drives", Proc. IEEE IAS Annual Meeting, 1987, pp 177-183.
- [2] V. Rajagopalan, " Computer Aided Analysis of Power Electronics Systems", Marcel Dekker inc., N.Y, 1987.
- [3] J. Lesenne, F. Notelet & G. Séguier," Introduction à l'électronique approfondie ", Technique et documentation, Paris 1981.
- [4] A. Ba-Razzouk, " Étude et mise au point d'un module de simulation des systèmes d'entraînement à courant alternatif ", Mémoire, Université du Québec à Trois-Rivières , décembre 1992.
- [5] A. Ba-Razzouk, "Note de cours : Entraînement électronique des machines électriques ", Université du Québec à Trois-Rivières, Automne 1997.
- [6] B. K.Bose, " Power Electronics and AC drives", New Jersey : Prentice Hall, ch 7 , pp264-291, 1986.
- [7] Y. El Bittioui, " Rapport de séminaire 2", cours de séminaire, Hiver 1998.
- [8] J. F. Jodouin, " Les Réseaux de Neurones " Principes et définitions" Hennes, Paris, 1994
- [9] A. Meftah, " Applications des Réseaux de Neurones dans le domaine de l'Électronique de puissance", mémoire de maîtrise, Université du Québec à Trois-Rivières, juin 1997.
- [10] E. N. Marieb, " Anatomie et Physique Humaine", éditions du Renouveau Pédagogique Inc, Canada 1993.

- [11] P. Davalo, " Des Réseaux de Neurones ", Editions Eyrolles, Paris 1989.
- [12] M. H. Hassoun, " Fundamentals of Artificial Neural Networks ", MIT, USA ,1995.
- [13] S. Haykin, " Neural Networks A comprehensive foundation ", Mac millan college publishing company, New York, 1992.
- [14] M.T. Hagen & M.B. Menhaj, " Training Feedforward Neural Networks With the Marquardt Algorithm ", IEEE Transactions on Power Electronics, Vol 5, N° 6, November 1994, p. 989-993.
- [15] H. Demuth, M. Beale, " Neural Network Toolbox ", The Mathworks inc, january 1994.
- [16] A. Ba-Razzouk, A.Chériti & G.Olivier, " A Neural Network Based Field Oriented Control Scheme For Induction Motors", IEEE Industry Applications Society, Annual Meeting, New Orléans, Louisiana, October 5-9, 1997.
- [17] A. Ba-Razzouk, A. Chériti & G. Olivier, " Artificial Neural Network Rotor Time Constant Adaptation in Indirect Field Oriented Control Drives", IEEE Power Electronics Specialits Conference (PESC), juin 1996, p.780-786.
- [18] L.C. Zai & T.A. Lipo, " An Extended Kalman Filter Approach to Rotor Time Constant Measurment in PWM Induction Motor Drives", Proc.of IEEE IAS Annual Meeting, 1987, p. 177-183.
- [19] C. C. Chan & H. Wang, " An Effective Method for Rotor Resistance Identification For high Performance Induction Motor Vector Control", IEEE Transactions on Industrial Electronics, Vol.37, No 6, December 1990.
- [20] J.T. Boys & M.J. Miles, " Empirical Thermal Model for Inverter-Driver Cage Induction Machines ", IEE Proc. Electr. Power Appl., vol. 141, No 6 , November 1994.

- [21] G. Grellet & G. Clerc, " Actionneur Électriques : Principes Modèles Commandes", Editions Eyrolles, Paris, 1998.
- [22] D. J. Atkinson, J. W. Finich & P. P. Acarnley, " Estimation of Rotor Resistance in Induction Motors ", IEE Proc. Electr. Power Appl., Vol. 143, No 1, January 1996.
- [23] " MX31 Modular Embedded System, Developer's Guide", Version2, integrated motions inc. Berkeley, California, 1993.
- [24] Irfan Ahmed, " Digital control application with the TMS320 family", Texas Instrument, USA, May 1995.

## ANNEXE A

### Programme d'apprentissage du réseau de neurones

```
clear;
```

```
% chargement de données d'apprentissage  
load mat.mat
```

```
% Coefficients de normalisation  
maxP = 105;  
maxT = 1;  
P1=P/maxP;  
T1=T/maxT;
```

```
% Structure du RNA  
R=5; % 5 entrées  
S1=20; % 20 neurones sur la couche cachée  
S2=1; % 1 neurone de sortie
```

```
F='tansig'; % Définition des fonctions non linearaires
```

```
% Initialisation du RNA  
[W1,B1]=nwtan(S1,R);  
[W2,B2]=nwtan(S2,S1);
```

```
% Paramètres d'apprentissage  
%*** pour trainbp ***  
TP(1)=20; % Epochs between updating display, default = 25.  
TP(2)=10000; % Maximum number of epochs to train, default = 1000.  
TP(3)=0.0001; % Sum-squared error goal, default = 0.02.  
TP(4)=0.01; % Learning rate, 0.01.  
TP(5)=10; % Learning rate increase, default = 1.05.  
TP(6)=0.1; % Learning rate decrease, default = 0.7.  
TP(7)=0.9; % Momentum constant, default = 0.9.  
TP(8)=1.04; % Maximum error ratio, default = 1.04.
```

```
%*** pour trainlm ***
%TP(1)= 20;      % Affichage
%TP(2)= 20000;   % Itérations maximales
%TP(3)= 0.0001;% Erreur désirée
%TP(4)= 0.0001; % Gradient de trainlm
%TP(5)= 0.0001; % Mu de trainlm
%TP(6)= 10;      % Multiplicateur pour incrementer de Mu de trainlm
%TP(7)= 0.1;     % Multiplicateur pour décrementer de Mu de trainlm
%TP(8)= 1e10;    % Mu limite
```

```
[%W1,B1,W2,B2,TE,TR] = trainlm(W1,B1,F,W2,B2,F,P1,T1,TP);
[W1,B1,W2,B2,TE,TR] = trainbpx(W1,B1,F,W2,B2,F,P1,T1,TP);
```

```
save matannf1.mat W1 B1 W2 B2 P T
[A1,A2] = simuff(P1,W1,B1,F,W2,B2,F);
k=1:length(P);
plot(k,A2,k,T1)
```

## ANNEXE B

### Programme du contrôle vectoriel sans adaptation

```
/*=====
*Ce fichier permet de sortir les trois consignes de courant pour une      *
* commande vectorielle                                              *
* Lecture de la vitesse par impulsion et interruption 1                *
* Consigne du couple lire sur B3AD1                                     *
* Sens de rotation lu sur B3AD0                                     *
*=====
#include "/winboss/include/math.h"
#include "/winboss/include/timers.h"
#include "/winboss/include/mx31_dev.h"
#include "/winboss/include/parall2.h"
#include "/winboss/include/genutil2.h"
#include "/winboss/include/uart2.h"
#include "/winboss/include/c30io.h"
#include "/winboss/include/bossutil.h"
#include "/winboss/include/devutil2.h"
#include "/winboss/include/aio122.h"
#define Ts 0.000400          /*Timer Period Secondes*/
#define T (Ts*4166666.6666667)
#define pi 3.14159265359
#define C1 14.3266          /* Inverse of mutual inductance */
#define C2 0.3472          /* 2.Lr/(3.pp.M) */
#define C3 0.3840          /* M/Tr*/
#define K1 0.8165          /* sqrt(2/3) */
#define K2 -0.4082          /* -1/sqrt(6) */
#define K3 0.7071          /* sqrt(2)/2 */
#define Psic 0.55          /* Flux command */
#define pp 2.0              /* Pairs of poles */
double idse, iqse;
double idss, iqss;
double ias, ibs, ics;
void c_int01(void);
void c_int08(void);
```

```

int P, P1, wait, Kg;
double wslc, wmecl=0.0, teta, teta0=0.0, f, sens, Temc, x;
int V=8333333;
int *TCR1=(int *)0x808030; /*Timer1 control register */
int *TPR1=(int *)0x808038; /*Timer1 period register */
int *TCA1=(int *)0x808034; /*Timer1 count register */
int *TCR0=(int *)0x808020; /*Timer0 control register */
int *TPR0=(int *)0x808028; /*Timer0 period register */
int *TCA0=(int *)0x808024; /*Timer0 count register */
int *B3AD0=(int*)0x780F05;
int *B3AD1=(int*)0x780F06;
int *B3DA0=(int*)0x780F02;
int *B3DA1=(int*)0x780F03;
int *B3ADT=(int*)0x780F01;
int *B2DA0=(int*)0x780E02;
int *B2DA1=(int*)0x780E03;
main ()
{
    Display("FOC-RUNS!");
    IntVectWrite(8, c_int08);
    IntVectWrite(1, c_int01);
    asm(" AND 0000h,IF"); /* Clear all possible pending Interrupts */
    asm(" OR 2000h,ST"); /* Set Global Interrupt Enable */
    asm(" OR 0002h,IE"); /* Set EINT1 Interrupt Enable */
    *TCR1=0x301; /* Timer1 configuration */
    *TPR1=4294967295; /* Timer1 period to max*/
    *TCA1=0; /* Reset Timer1*/
    *TCR1=0x3C1; /* Start Timer1*/
    asm(" OR 0100h,IE"); /*Set Timer0 Interrupt Enable*/
    *TCR0=0x301; /*Configuration du Timer0*/
    *TPR0=T; /*Fixer la periode du Timer0*/
    *TCA0=0; /*Reset Timer0*/
    *TCR0=0x3C1; /*Start Timer0*/
    while (1)
    {

```

```

/*===== Lecture de la vitesse =====*/
f= (1953125.000015625/(double) V);
/*===== Lecture du sens de rotation =====*/
P= *B3AD0&0xFFFF;
if(P & 0x800)
    P |= 0xffff000;
x = 0.0048828125*(P+2048)-11.6;
if (x > 0 & f >= 1)
    sens=-1.0;
    Kg=1;
if (x < 0 & f >= 1)
    sens=1.0;
    Kg=1;
if (f < 1)
    sens=0;
    Kg=1;
wmec=sens*0.10471975511966*f; /* RPM to rad.s-1 */
/*===== Lecture de la consigne du couple =====*/
P1= *B3AD1&0xFFFF;
if(P1 & 0x800)
    P1 |= 0xffff000;
Temc=0.0048828125*(P1+2048)-10;
/*===== Calcul des consignes synchrones =====*/
idse=C1*Psic;
iqse=C2*Temc/Psic;
wslc=C3*iqse/Psic;
/*===== Integration de we =====*/
teta=Ts*wslc*Kg-Ts*pp*wmec+teta0;
if (teta >= pi)
    teta0=-pi;
else if (teta <= -pi)
    teta0=pi;
else
    teta0=teta;
/*===== Calcul des consignes stationnaires =====*/
idss=idse*cos(teta)-sin(teta)*iqse;
iqss=idse*sin(teta)+cos(teta)*iqse;

```

```

/*===== Calcul des consignes de phase =====*/
    ias=K1*idss;
    ibs=K2*idss-K3*iqss;
    ics=K2*idss+K3*iqss;
    *B3DA0 = (int)(204.8*ias+ 2048);
    *B3DA1 = (int)(204.8*ibs+ 2048);
    *B2DA0 = (int)(204.8*ics+ 2048);
    *B2DA1 = (int)(204.8*(-wmec/50)+2048);
    wait=1;
    while(wait);
}
void c_int01(void)
{
    V= *TCA1&0xFFFFFFFF; /* Read Timer1 */
    asm(" AND 7Fdh,IF"); /* Clear INT1 */
    (*TCA1)=0; /* Reset Timer1 */
    (*TCR1)=0x3C1; /* Start Timer1 */
}
void c_int08(void)
{
    *B3ADT=1;
    wait=0;
    /* asm(" AND 6FFh,IF"); */ /* Clear TINT0 */
}

```

## ANNEXE C

### Programme du contrôle vectoriel avec adaptation

```
/*=====
 * Ce fichier réalise une commande vectorielle dans le cas          *
 * d'une alimentation en Tension          *
 * Lecture de la vitesse par impulsion et interruption 1          *
 * Consigne du couple lire sur B3AD1          *
 * Sens de rotation lu sur B3AD0          *
 */=====

#include "/winboss/include/math.h"
#include "/winboss/include/timers.h"
#include "/winboss/include/mx31_dev.h"
#include "/winboss/include/parall2.h"
#include "/winboss/include/genutil2.h"
#include "/winboss/include/uart2.h"
#include "/winboss/include/c30io.h"
#include "/winboss/include/bossutil.h"
#include "/winboss/include/devutil2.h"
#include "/winboss/include/aio122.h"
#define Ts 0.000400      /*Timer Period Secondes*/
#define T (Ts*4166666.6666667)
#define pi 3.14159265359
#define C1 14.3266      /* Inverse of mutual inductance */
#define C2 0.3472      /* 2.Lr/(3.pp.M) */
#define C3 0.1920      /* M/0.5*Tr/0.5*0.3840 */
#define K1 0.8165      /* sqrt(2/3) */
#define K2 -0.4082      /* -1/sqrt(6) */
#define K3 0.7071      /* sqrt(2)/2 */
#define Psic 0.55      /* Flux command */
#define pp 2.0          /* Pairs of poles */
double idse, iqse;
double idss, iqss;
double ias, ibs, ics;
void c_int01(void);
void c_int08(void);
int P, P1, wait, Kg;
```

```

double wslc, wmec=0.0, teta, teta0=0.0, f, sens, Temc, x;
int V=8333333;
int *TCR1=(int *)0x808030; /*Timer1 control register */
int *TPR1=(int *)0x808038; /*Timer1 period register */
int *TCA1=(int *)0x808034; /*Timer1 count register */
int *TCR0=(int *)0x808020; /*Timer0 control register */
int *TPR0=(int *)0x808028; /*Timer0 period register */
int *TCA0=(int *)0x808024; /*Timer0 count register */
int *B3AD0=(int*)0x780F05;
int *B3AD1=(int*)0x780F06;
int *B3DA0=(int*)0x780F02;
int *B3DA1=(int*)0x780F03;
int *B3ADT=(int*)0x780F01;
int *B2DA0=(int*)0x780E02;
int *B2DA1=(int*)0x780E03;
main ()
{
    Display("FOC-RUNS!");
    IntVectWrite(8, c_int08);
    IntVectWrite(1, c_int01);
    asm(" AND 0000h,IF"); /* Clear all possible pending Interrupts */
    asm(" OR 2000h,ST"); /* Set Global Interrupt Enable */
    asm(" OR 0002h,IE"); /* Set EINT1 Interrupt Enable */
    *TCR1=0x301; /* Timer1 configuration */
    *TPR1=4294967295; /* Timer1 period to max*/
    *TCA1=0; /* Reset Timer1*/
    *TCR1=0x3C1; /* Start Timer1*/
    asm(" OR 0100h,IE"); /*Set Timer0 Interrupt Enable*/
    *TCR0=0x301; /*Configuration du Timer0*/
    *TPR0=T; /*Fixer la periode du Timer0*/
    *TCA0=0; /*Reset Timer0*/
    *TCR0=0x3C1; /*Start Timer0*/
    while (1)
    {
/*===== Lecture de la vitesse =====*/
        f= (1953125.000015625/(double) V);
/*===== Lecture du sens de rotation =====*/
    }
}

```

```

P= *B3AD0&0xFFF;
if(P & 0x800)
    P |= 0xfffff000;
x = 0.0048828125*(P+2048)-11.6;
if (x > 0 & f >= 1)
    sens=-1.0;
    Kg=1;
if (x < 0 & f >= 1)
    sens=1.0;
    Kg=1;
if (f < 1)
    sens=0;
    Kg=1;
wmech=sens*0.10471975511966*f; /* RPM to rad.s-1 */
/*===== Lecture de la consigne du couple =====*/
P1= *B3AD1&0xFFF;
if(P1 & 0x800)
    P1 |= 0xfffff000;
Tmec=0.0048828125*(P1+2048)-10;
/*===== Calcul des consignes synchrones =====*/
idse=C1*Psic;
iqse=C2*Tmec/Psic;
wslc=C3*iqse/Psic;
/*===== Integration de we =====*/
teta=Ts*wslc*Kg-Ts*pp*wmech+teta0;
if (teta >= pi)
    teta0=-pi;
else if (teta <= -pi)
    teta0=pi;
else
    teta0=teta;
/*===== Calcul des consignes stationnaires =====*/
idss=idse*cos(teta)-sin(teta)*iqse;
iqss=idse*sin(teta)+cos(teta)*iqse;
/*===== Calcul des consignes de phase =====*/
ias=K1*idss;

```

```

    ibs=K2*idss-K3*iqss;
    ics=K2*idss+K3*iqss;
    *B3DA0 = (int)(204.8*ias+ 2048);
    *B3DA1 = (int)(204.8*ibs+ 2048);
    *B2DA0 = (int)(204.8*ics+ 2048);
    *B2DA1 = (int)(204.8*(-wmeec/50)+2048);
    wait=1;
    while(wait);
}
void c_int01(void)
{
    V= *TCA1&0xFFFFFFFF; /* Read Timer1 */
    asm(" AND 7Fdh,IF"); /* Clear INT1 */
    (*TCA1)=0; /* Reset Timer1 */
    (*TCR1)=0x3C1; /* Start Timer1 */
}
void c_int08(void)
{
    *B3ADT=1;
    wait=0;
    /* asm(" AND 6FFh,IF"); */ /* Clear TINT0 */
}

```