



# Commande directe de couple d'un moteur asynchrone à base de techniques intelligentes

Soufien Gdaim

## ► To cite this version:

Soufien Gdaim. Commande directe de couple d'un moteur asynchrone à base de techniques intelligentes. Intelligence artificielle [cs.AI]. Ecole Nationale d'Ingénieurs de Monastir. Tunisia, 2013. Français. <NNT : >. <tel-01456394v2>

**HAL Id: tel-01456394**

**<https://theses.hal.science/tel-01456394v2>**

Submitted on 8 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*



Université de Monastir

\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*

Ecole Nationale d'Ingénieurs de Monastir

\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*\_\*

Année universitaire 2012/2013

# THESE

*Présentée à l'*

Ecole Nationale d'Ingénieurs de Monastir

Pour obtenir le grade de

## DOCTEUR

Dans la discipline

### Génie Electrique

Présenté par

**Soufien GDAIM**

*(Ingénieur Génie Electrique)*

*(Mastère en Electronique et Informatique temps réel)*

---

---

## Commande directe de couple d'un moteur asynchrone à base de techniques intelligentes

---

---

Soutenue le 26/12/2013 devant le jury composé de :

M. Ali DOUIK

Président de Jury

Maitre de conférences à l'ENIM

M. Lotfi KRICHEN

Rapporteur

Maitre de conférences à l'ENIS

M. Anis SELLAMI

Rapporteur

Maitre de conférences à l'ESSTT

M Abdellatif MTIBAA

Examineur

Professeur à L'ENIM

M. Med Faouzi MIMOUNI

Directeur de thèse

Professeur à l'ENIM



*Je dédie cette thèse à la mémoire de mon père*

# Avant-propos

*Le travail présenté dans ce mémoire a abouti grâce à une collaboration entre l'équipe CSR du laboratoire Electronique et Micro-électronique de la Faculté des sciences de Monastir et l'unité de recherche ESIER (Etude des Systèmes Industriels et Energies Renouvelables) à l'ENIM.*

*Mes remerciements ainsi que le témoignage de mon profond respect s'adressent tout d'abord à mon encadreur Monsieur Abdellatif MTIBAA professeur à l'Ecole Nationale des Ingénieurs de Monastir, pour m'avoir accueilli dans son équipe de recherche CSR. Son encadrement et surtout les conseils avisés qu'il m'a prodigué, témoignent de ses qualités scientifiques et humaines*

*Je tiens à exprimer ma profonde gratitude à Monsieur Mohamed Faouzi MIMOUNI, professeur à l'Ecole Nationale des Ingénieurs de Monastir, pour avoir dirigé ce travail et avec qui j'ai beaucoup appris dans le domaine théorique et pratique. Je le remercie de tout mon cœur pour le temps qu'il m'a consacré, pour ces nombreux conseils et son soutien régulier.*

*J'exprime ma reconnaissance à Monsieur Ali DOUIK, maître de conférences à l'Ecole Nationale des Ingénieurs de Monastir, pour l'honneur qu'il m'a fait en acceptant de juger mon travail et de présider mon jury de thèse.*

*Je remercie les deux rapporteurs de ce travail, Monsieur Lotfi Krichen maître de conférences à l'ENIS et Monsieur Anis SELLAMI maître de conférences à l'ESSTT d'avoir accepté de rapporter cette thèse. Je les remercie en particulier pour leurs remarques judicieuses.*

*Que tous les membres de l'équipe CSR se trouvent ici remerciés. La très bonne humeur et l'ambiance au sein du groupe m'ont permis de mener mes travaux de recherche de façon agréable. Les discussions, les remarques et les commentaires de mes collègues ont été sources d'idées et ont contribué au développement et à l'amélioration de cette étude.*

*Pour finir, je ne saurais oublier le soutien compréhensif de ma famille, particulièrement de :*

- *Papa Stoufa pour ses encouragements,*
- *Ma mère Saida pour son amour, sa profonde tendresse, ses prières et son encouragement,*
- *Ma femme Beya, la seule personne qui a partagé avec moi les moments d'inquiétude avant ceux de joie,*
- *Celles qui illuminent mes journées de leurs présences, mes princesses Farah, Feriel et Elaa.*
- *Mes Frères Chahir, Nizar, Maher, Nabil et Afef.*

*Je dédie donc ce travail à toute ma famille GDAIM et MEJRI.*

## Table des matières

<b>Introduction générale .....</b>	<b>1</b>
<b>1 Chapitre 1 : Etat de L'art de la Commande DTC de la Machine Asynchrone .....</b>	<b>6</b>
1.1 Introduction .....	7
1.2 Commande Directe de Couple .....	7
1.2.1 Avantages de la Commande DTC .....	8
1.2.2 Inconvénients de la Commande Directe de Couple .....	8
1.3 Etat de l'art sur la Commande DTC de la machine asynchrone .....	8
1.3.1 Les stratégies DTC à fréquence de commutation contrôlée.....	9
1.3.2 Les stratégies DTC à fréquence de commutation non contrôlée.....	12
1.4 Implémentation matérielle des algorithmes de commande sur FPGA.....	15
1.4.1 Etat de l'art sur l'implémentation sur les FPGAs .....	15
1.4.2 Environnement de description.....	16
1.4.3 Configuration d'un FPGA .....	18
1.4.4 Démarche pour la description et la simulation.....	18
1.5 Conclusion.....	19
<b>2 Chapitre 2 : Modélisation et commande DTC de la Machine asynchrone .....</b>	<b>21</b>
2.1 Introduction .....	22
2.2 Modélisation de la machine asynchrone .....	22
2.2.1 Présentation de la machine .....	22
2.2.2 Hypothèses simplificatrices.....	22
2.2.3 Modèle diphasée de la machine asynchrone .....	23
2.3 Alimentation de la MAS par un onduleur de tension .....	24
2.3.1 Introduction .....	24
2.3.2 Structure de l'onduleur .....	24
2.3.3 Détermination des tensions de sortie de l'onduleur : .....	25
2.3.4 Commande de l'onduleur triphasé .....	26
2.4 Commande directe de couple de la machine asynchrone.....	26
2.4.1 Introduction .....	26
2.4.2 Structure de commande des moteurs asynchrones par DTC.....	27
2.4.3 Stratégie de commande directe de couple et de flux .....	27
2.4.4 Elaboration de la table de commutation .....	42
2.5 Résultats de simulation.....	43
2.6 Conclusion.....	51
<b>3 Chapitre 3 : Commande DTC d'une MAS via l'utilisation des techniques intelligentes.....</b>	<b>52</b>
3.1 Introduction .....	53
3.2 Commande directe floue de couple (DTFC) d'une MAS .....	53
3.2.1 Introduction .....	53
3.2.2 Principe de la logique floue.....	55
3.2.3 Commande par logique floue .....	57

3.2.4	Commande directe floue de couple de la machine asynchrone .....	59
3.2.5	Résultats de simulation.....	63
3.2.6	Conclusion.....	66
3.3	Commande Directe Neuronale de Couple (DTNC) d'une MAS .....	66
3.3.1	Introduction .....	66
3.3.2	Présentation générale.....	67
3.3.3	Architectures d'un réseau de neurones.....	68
3.3.4	Notion d'apprentissage.....	69
3.3.5	Conception d'un réseau de neurones.....	70
3.3.6	L'algorithme d'apprentissage de RNA .....	71
3.3.7	Application des RNA pour la commande DTC d'une MAS.....	73
3.3.8	Résultat de simulation .....	74
3.3.9	Conclusion.....	76
3.4	Etude comparative.....	76
3.5	Conclusion.....	80
<b>4</b>	<b>Chapitre 4 : Implémentation de la commande DTC intelligente de la MAS sur FPGA .....</b>	<b>81</b>
4.1	Introduction .....	82
4.2	Étude et implémentation de l'algorithme de commande directe de couple de la machine asynchrone.....	82
4.2.1	Introduction .....	82
4.2.2	Flot et environnement de conception adoptés .....	82
4.2.3	Modélisation et Spécification de la chaîne de commande .....	83
4.2.4	Spécification de la commande directe de couple de la MA .....	85
4.2.5	Transformations de haut niveau .....	87
4.2.6	Description RTL.....	87
4.2.7	Résultat de simulation .....	91
4.2.8	La synthèse.....	93
4.2.9	Simulation logique .....	95
4.2.10	Conclusion.....	97
4.3	Implémentation de la Commande Directe Floue de Couple .....	97
4.3.1	Description du module Fuzzification .....	97
4.3.2	Description du module moteur d'inférence.....	100
4.3.3	Amélioration du module moteur d'inférence .....	101
4.3.4	Description du module defuzzification .....	102
4.3.5	Architecture de la Commande DTC floue.....	102
4.4	Implémentation de la Commande Directe Neuronale de Couple.....	105
4.4.1	Description de l'unité neurone : .....	106
4.4.2	Description d'une couche.....	108
4.4.3	Description de la propagation .....	110
4.4.4	Description de l'algorithme de rétropropagation .....	112
4.4.5	Réalisation de l'apprentissage.....	114
4.4.6	La commande DTC neuronale .....	117
4.5	Conclusion.....	120

<b>5</b>	<b>Chapitre 5 : Résultats expérimentaux de la commande DTC intelligente.....</b>	<b>121</b>
5.1	Introduction .....	122
5.2	Présentation du banc d'essai expérimental .....	122
5.2.1	Le circuit de puissance .....	124
5.2.2	Système de mesure et d'acquisition.....	124
5.2.3	Partie commande .....	125
5.2.4	Interface de commande .....	127
5.3	Etude expérimentale .....	129
5.3.1	Vérification des signaux de commande .....	129
5.3.2	Implémentation de la DTC .....	130
5.3.3	Implémentation de la DTFC.....	134
5.4	Conclusion.....	137
	<b>Conclusion générale .....</b>	<b>138</b>
	<b>Bibliographie.....</b>	<b>142</b>
<b>6</b>	<b>Annexes .....</b>	<b>154</b>
6.1	Annexe A : Bibliothèque de virgule fixe .....	155
6.2	Annexe B : Acquisition des grandeurs électriques.....	157
6.3	Annexe C : Réalisation de la carte de commande de l'onduleur .....	164



## *Liste des Tableaux*

Tableau 2.1. Tableau de la DTC conventionnelle définissant les séquences d'un onduleur. ..	43
Tableau 2.2. Paramètre de la machine asynchrone .....	43
Tableau 3.1. Différentes T-normes et S-normes .....	57
Tableau 3.2. Base de règles adoptées .....	62
Tableau 3.3. Résultats de comparaison des écarts des ondulations .....	78
Tableau 4.1. Ressources utilisées sur le FPGA par l'algorithme de commande.....	95
Tableau 4.2 : Ressources utilisées sur le FPGA par l'algorithme de commande DTFC .....	105
Tableau 4.3 : Fonctionnement de la machine à états finis .....	111
Tableau 4.4 : Fonctionnement de la fsm liée à l'apprentissage .....	115
Tableau 4.5 : Ressources utilisées sur le FPGA par l'algorithme de commande Neuronale. ....	119
Tableau 5.1. Performances temporelles de l'architecture DTC .....	131
Tableau 5.2. Performances temporelles de l'architecture DTFC.....	135
Tableau 6.1. Tableau des entrées et des sorties de la bibliothèque en virgule fixe.....	156

## *Liste des figures*

### **Chapitre 1**

Figure 1.1. Stratégie de Commande Directe de Couple .....	7
Figure 1.2. Schéma détaillé du bloc estimateur du flux statorique sous XSG .....	17
Figure 1.3. Configuration d'un FPGA. ....	18
Figure 1.4. Environnement de description et de simulation.....	19

### **Chapitre 2**

Figure 2.1. Structure diphasée de la machine asynchrone dans le repère d-q.....	23
Figure 2.2. Structure d'un onduleur triphasé de tension à deux niveaux.....	25
Figure 2.3. Schéma structurel de la commande DTC. ....	27
Figure 2.4. Vecteurs actifs des tensions statoriques et secteurs .....	29
Figure 2.5. Schéma de principe d'un estimateur d'état.....	30
Figure 2.6. Schéma de principe d'un observateur d'état.....	32
Figure 2.7. Fonction signe (commande de type relais) .....	37
Figure 2.8. Fonction saturation (commande adoucie).....	38
Figure 2.9. Schéma structurel de la commande DTC a base d'un observateur de flux statorique. ....	41
Figure 2.10: a) Choix des tensions $V_i$ correspondant au contrôle de l'amplitude du $\phi_s$ .....	41
Figure 2.11. Comparateur à hystérésis de couple à trois niveaux. ....	42
Figure 2.12 Résultats de simulation de la commande DTC de Takahashi.....	44
Figure 2.13. Résultats de la commande DTC de Takahashi à base d'un estimateur de flux et sans variation de la résistance statorique.....	46
Figure 2.14. Résultats de la commande DTC de Takahashi à base d'un estimateur de flux et avec variation de la résistance statorique. ....	47
Figure 2.15. Résultats de la commande DTC de Takahashi à base d'un observateur par mode glissant et avec variation de la résistance statorique. ....	49
Figure 2.16. Résultats de la commande DTC de Takahashi à base d'un observateur à grand gain et avec variation de la résistance statorique. ....	50
Figure 2.17. Résultats de la commande DTC de Takahashi à base d'un observateur par mode glissant équipé d'un mécanisme d'adaptation de la résistance statorique. ....	51

### **Chapitre 3**

Figure 3.1. Comparaison d'un ensemble classique et d'un ensemble flou .....	55
Figure 3.2. Classification des personnes suivant leurs âges.....	56
Figure 3.3. Fonctions de forme trapézoïdale.....	56
Figure 3.4. Principe de la commande par logique floue.....	58
Figure 3.5. Schéma de principe de la commande directe floue de la MA.....	59
Figure 3.6. Les fonctions d'appartenances de l'erreur de flux $e_\phi$ .....	60
Figure 3.7. Les fonctions d'appartenances de l'erreur de couple $e_c$ .....	60
Figure 3.8. Les fonctions d'appartenances de l'angle $\theta_s$ .....	60
Figure 3.9. Vecteur flux statorique dans le plan $(\alpha, \beta)$ . FRa/FRd: flux rapide augmentation /diminution. FFa/FFd: flux faible augmentation /diminution. CRa/CRd: Couple rapide augmentation /diminution. CFa/CFd: Couple faible augmentation /diminution.....	61
Figure 3.10. Les fonctions d'appartenances de la sortie .....	61
Figure 3.11. Diagramme de système d'inférence flou .....	63

Figure 3.12. Couple électromagnétique. (a). DTC conventionnelle. (b). DTC flou. (c). comparaison DTC conventionnelle et DTC flou.....	64
Figure 3.13 .Amplitude du flux statorique. (a). DTC conventionnelle. (b). DTC flou. (c). comparaison DTC conventionnelle et DTC flou.....	64
Figure 3.14. Le courant statorique. (a).DTC conventionnelle. (b). DTC flou.....	65
Figure 3.15. Amplitude du flux statorique dans le plan ( $\alpha,\beta$ ). (a).DTC conventionnelle. (b). DTC flou.....	65
Figure 3.16. Commutation de l'interrupteur Sa. (a).DTC Floue. (b). DTC Floue (zoom). ....	65
Figure 3.17. Unité de base : Le neurone .....	67
Figure 3.18: Modèle perceptron .....	68
Figure 3.19. Architecture d'un réseau de neurones.....	69
Figure 3.20. Apprentissage supervisé. ....	70
Figure 3.21. Apprentissage non supervisé. ....	70
Figure 3.22. Algorithme d'apprentissage par la méthode de rétropropagation de l'erreur. ....	72
Figure 3.23. Schéma de principe de la commande DTNC de la MAS. ....	73
Figure 3.24. Structure du réseau neuronal utilisé pour la commande .....	74
Figure 3.25. Couple électromagnétique. (a). DTC conventionnelle. (b). DTC Neuronale. ....	75
Figure 3.26. Amplitude de flux statorique. (a). DTC conventionnelle. (b). DTC Neuronale. .	75
Figure 3.27. Amplitude du courant statorique. (a).DTC conventionnelle. (b). DTC Neuronale. ....	75
Figure 3.28. Amplitude du flux statorique dans le plan ( $\alpha,\beta$ ). (a).DTC conventionnelle. (b). DTC Neuronale. ....	76
Figure 3.29. Commutation de l'interrupteur Sa. (a).DTC Neuronale. (b). DTC Neuronale (zoom). ....	76
Figure 3.30. Résultats de simulations de la commande DTC et la DTC intelligente.....	78
Figure 3.31. Spectre de fréquence. ....	79

## Chapitre 4

Figure 4.1. Flot de conception adopté .....	83
Figure 4.2. Modèle de la structure générale de la commande.....	84
Figure 4.3. Décompositions algorithmiques .....	85
Figure 4.4. La méthodologie de décomposition et spécification de l'algorithme de commande .....	86
Figure 4.5. Module VHDL de l'onduleur de tension. (a). Entité de module de l'onduleur de tension. (b). Architecture de module de l'onduleur de tension .....	88
Figure 4.6. Modèle VHDL de la Machine asynchrone. (a). Entité de module de MAS. (b). Architecture de module de MAS.....	89
Figure 4.7. Structure algorithmique de commande DTC d'une MAS .....	89
Figure 4.8. Modèle d'architecture de l'Algorithme DTC de la MAS .....	90
Figure 4.9. Architecture module Estimation. (a) Séquenceur de l'estimation. (b) Chemin de données.....	91
Figure 4.10. Simulation de la commande DTC de la machine asynchrone .....	92
Figure 4.11. Schéma RTL de l'algorithme de commande .....	94
Figure 4.12. Le schéma RTL du module table de sélection.....	94
Figure 4.13 Résultat de routage de la commande DTC de la MA .....	95
Figure 4.14. Le couple électromagnétique (N.m). (a). Le modèle de simulation non synthétisable. (b).Le modèle de simulation synthétisable utilisant la bibliothèque de virgule fixe.....	96
Figure 4.15. La vitesse rotorique (rad/sec). (a). Le modèle de simulation non synthétisable. (b).Le modèle de simulation synthétisable utilisant la bibliothèque de virgule fixe.....	96

Figure 4.16. Le circuit combinatoire de $e_\phi$ .....	98
Figure 4.17. Le circuit combinatoire de $e_c$ .....	99
Figure 4.18. Le circuit combinatoire de l'angle du vecteur flux statorique $\theta$ .....	99
Figure 4.19. Architecture de la fuzzification.....	100
Figure 4.20. Architecture du module moteur d'inférence .....	101
Figure 4.21. Architecture du module moteur d'inférence avec sélecteur de règles actives... ..	102
Figure 4.22. Architecture de la defuzzification.....	102
Figure 4.23. Architecture de la commande DTC floue .....	103
Figure 4.24: Couple électromagnétique obtenu par simulation VHDL. (a). Commande DTC conventionnelle. (b). Commande DTC Floue .....	103
Figure 4.25. Schéma RTL de la DTC floue (Top Level) .....	104
Figure 4.26. Portion du schéma RTL du module Couple-Flux .....	104
Figure 4.27. Résultat de routage de l'architecture de la commande floue .....	105
Figure 4.28. Implémentation matérielle d'un neurone .....	106
Figure 4.29: Modèle VHDL de l'entité neurone .....	107
Figure 4.30. Simulation de la séquence d'exécution du module neurone.....	107
Figure 4.31. Schéma haut niveau du module Neurone. ....	108
Figure 4.32 : Structure d'une couche .....	108
Figure 4.33: Modèle VHDL de la couche1 .....	109
Figure 4.34 : Résultat de simulation d'une couche .....	109
Figure 4.35 : Structure du réseau de neurones. ....	110
Figure 4.36 : Modèle VHDL de la propagation .....	111
Figure 4.37. Les états de fsm_prop .....	112
Figure 4.38 : Résultat de simulation de l'entité prop .....	112
Figure 4.39 : Modèle VHDL de l'entité retro .....	113
Figure 4.40 : Mise à jour de poids d'un neurone .....	113
Figure 4.41 : Structure du module apprentissage.....	114
Figure 4.42 : Modèle VHDL de l'entité apprentissage .....	115
Figure 4.43. Les différents états de la machine d'états finis de l'apprentissage .....	116
Figure 4.44. Résultat de simulation du réseau de neurones adapté à la commande.....	117
Figure 4.45 : Structure générale de la commande neuronale .....	117
Figure 4.46. Couple électromagnétique obtenu par simulation VHDL. (a) Commande classique. (b).Commande neuronale. ....	118
Figure 4.47 : Résultat de synthèse de la DTC neuronale (Top Level) .....	118
Figure 4.48: Résultat de synthèse de la DTC neuronale du module réseau de neurones.....	119
Figure 4.49. Le résultat de routage de l'architecture de la commande Neuronale.....	120

## Chapitre 5

Figure 5.1 Photo du banc d'essai expérimental .....	123
Figure 5.2. Schéma synoptique du banc d'essai expérimental .....	123
Figure 5.3. Chaîne de mesure et d'acquisition.....	124
Figure 5.4. La carte de commande à base de FPGA : plateforme ML507.....	126
Figure 5.5. Contrôle et acquisition de données du convertisseur.....	126
Figure 5.6. Interface graphique de visualisation de données .....	127
Figure 5.7 : Schéma synoptique de la carte de commande rapprochée.....	128
Figure 5.8. Schéma de principe de l'optocoupleur .....	129
Figure 5.9. Les signaux de commande d'un même bras d'onduleur.....	129
Figure 5.10. Les signaux de commande de deux bras d'onduleur .....	130
Figure 5.11. Diagramme temporel de l'architecture de commande DTC.....	131
Figure 5.12. Les courants statoriques de la MAS .....	132

Figure 5.13 Résultats expérimentaux de la commande DTC à base d'un estimateur.....	133
Figure 5.14 Résultats expérimentaux de la commande DTC à base d'un observateur de flux .....	134
Figure 5.15. Diagramme temporel de l'architecture de commande DTFC .....	135
Figure 5.16. Résultats expérimentaux de la commande DTFC.....	136

## Annexes

Figure 6.1. Opération de multiplication sur 16 bits.....	156
Figure 6.2. Capteur de courant .....	157
Figure 6.3. Schéma de montage de la chaîne de mesure et d'adaptation du signal du signal	158
Figure 6.4. Echange des données entre le convertisseur ADS8509 et l'FPGA. ....	160
Figure 6.5. Schéma de câblage de l'ADS8509.....	160
Figure 6.6: Organigramme de déroulement d'une conversion.....	161
Figure 6.7: Schéma de montage de la carte 1 d'acquisition.....	162
Figure 6.8: Visualisation en 3D de la carte d'acquisition .....	162
Figure 6.9: Schématique de la carte des capteurs.....	163
Figure 6.10: Visualisation en 3D de la carte des capteurs .....	163
Figure 6.11 : Schéma synoptique de la carte de commande de l'onduleur.....	164
Figure 6.12: Schéma bloc de la carte de commande rapprochée .....	164
Figure 6.13 : Schéma de câblage de l'optocoupleur 6N136 .....	165
Figure 6.14: principe de création de temps mort.....	166
Figure 6.15: Schéma de câblage du circuit intégré CD4098.....	166
Figure 6.16. Montage de création de temps mort.....	167
Figure 6.17. Visualisation en 3D de la carte de commande de l'onduleur .....	167
Figure 6.18: Vue réelle de la carte de la commande de l'onduleur. ....	168
Figure 6.19: Schéma de circuit de protection de la carte de génération de temps mort.....	168
Figure 6.20: Schéma bloc des cartes d'isolations des signaux de commande.....	169
Figure 6.21: visualisation en 3D de carte d'isolation de signaux.....	170
Figure 6.22: vue réelle de carte d'isolation de signaux.....	170

## Notations et abréviations utilisées

Les notations et les abréviations utilisées lors de la rédaction de ce mémoire, sont indiquées dans les tableaux suivants.

### 1. Notations utilisées :

Symbole	Signification
$R_s$	Résistance d'une phase du stator
$R_r$	Résistance d'une phase de rotor
$T_s$	Constante de temps statorique
$T_r$	Constante de temps rotorique
$L_s$	Inductance propre de stator
$L_r$	Inductance propre de rotor
$M_s$	Inductance mutuelle entre deux phases statoriques
$M_r$	Inductance mutuelle entre deux phases rotoriques
$M_{sr}$	Inductance mutuelle cyclique entre le stator et le rotor
$M$	La valeur maximale de l'inductance mutuelle cyclique
$l_s$	Inductance cyclique propre de stator
$l_{s0}$	Inductance cyclique homopolaire de stator
$l_r$	Inductance cyclique propre de rotor
$l_{r0}$	Inductance cyclique homopolaire de rotor
$p$	Nombre de paires de pôles
$\theta$	Position de rotor par rapport au stator
$\theta_s$	Position de stator par rapport au repère (d,q)
$\theta_r$	Position de rotor par rapport au repère (d,q)
$\omega_s$	Pulsation des courants statoriques
$\omega_r$	vitesse électrique du rotor
$\Omega$	vitesse mécanique du rotor
$J$	Moment d'inertie
$f$	Coefficient de frottement
$C_{em}$	Couple électromagnétique
$C_r$	Couple résistant
$\sigma$	Coefficient de dispersion
$T_e$	Période d'échantillonnage
$(v_A, v_B, v_C)$	Tensions statoriques triphasées
$(i_A, i_B, i_C)$	Courants statoriques

$(i_a, i_b, i_c)$	Courants rotoriques
$\varphi_s = (\varphi_A, \varphi_B, \varphi_C)$	Flux statorique
$\varphi_r = (\varphi_a, \varphi_b, \varphi_c)$	Flux rotorique
$v_{sa\beta} = (v_{sa}, v_{s\beta})^T$	Composantes directe et quadrature de la tension statorique dans le repère $(\alpha, \beta)$
$i_{sa\beta} = (i_{sa}, i_{s\beta})^T$	Composantes directe et quadrature de courant statorique dans le repère $(\alpha, \beta)$
$i_{ra\beta} = (i_{ra}, i_{r\beta})^T$	Composantes directe et quadrature de courant rotorique dans le repère $(\alpha, \beta)$
$\varphi_{sa\beta} = (\varphi_{sa}, \varphi_{s\beta})^T$	Composantes directe et quadrature de flux statorique dans le repère $(\alpha, \beta)$
$\varphi_{ra\beta} = (\varphi_{ra}, \varphi_{r\beta})^T$	Composantes directe et quadrature de flux rotorique dans le repère $(\alpha, \beta)$
$v_{sdq} = (v_{sd}, v_{sq})^T$	Composantes directe et quadrature de la tension statorique dans le repère $(d, q)$
$i_{sdq} = (i_{sd}, i_{sq})^T$	Composantes directe et quadrature de courant statorique dans le repère $(d, q)$
$i_{rdq} = (i_{rd}, i_{rq})^T$	Composantes directe et quadrature de courant rotorique dans le repère $(d, q)$
$\varphi_{sdq} = (\varphi_{sd}, \varphi_{sq})^T$	Composantes directe et quadrature de flux statorique dans le repère $(d, q)$
$\varphi_{rdq} = (\varphi_{rd}, \varphi_{rq})^T$	Composantes directe et quadrature de flux rotorique dans le repère $(d, q)$
$[\cdot]^T$	Désigner la transposée d'une matrice ou d'un vecteur
$[\cdot]^{-1}$	Désigner l'inverse d'une matrice
$\overline{X}$	désigne le vecteur espace de $X$
$\hat{X}$	Grandeur estimée
$\dot{X} = \frac{dX}{dt}$	désigne le dérivé de $X$ par rapport au temps

## 2. Abréviations utilisées :

Acronyme	Signification
AAA ou A <sup>3</sup>	Adéquation Algorithme Architecture
ASIC	Application Specific Integrated Circuit
CAN	Convertisseur Analogique Numérique
DTC	Commande Directe de Couple ( <b>D</b> irect <b>T</b> orque <b>C</b> ontrol)
DTFC	Commande Directe Floue de Couple ( <b>D</b> irect <b>T</b> orque <b>F</b> uzzy <b>C</b> ontrol)
DTNC	Commande Directe Neuronale de Couple ( <b>D</b> irect <b>T</b> orque <b>N</b> eural <b>C</b> ontrol)
FPGA	<b>F</b> ield <b>P</b> rogrammable <b>G</b> ate <b>A</b> rray
FSM	<b>F</b> inite <b>S</b> tate <b>M</b> achine
MAS	<b>M</b> achine <b>A</b> synchrone
SVM	<b>S</b> pace <b>V</b> ector <b>M</b> odulation
VHDL	<b>V</b> ery <b>H</b> igh <b>S</b> peed <b>I</b> ntegrated <b>C</b> ircuit <b>H</b> ardware <b>D</b> escription <b>L</b> angage

# **Introduction générale**



## **Introduction générale**

La commande des systèmes automatisés de production à base de techniques intelligentes, tels que la logique floue, les réseaux de neurones artificiels ou les algorithmes génétiques, est de plus en plus utilisée dans plusieurs domaines d'application (Reconnaissance de la parole ou d'images, la robotique, l'automatisme etc. D'une façon générale, les algorithmes basés sur l'intelligence artificielle sont considérés comme une solution très intéressante pour la commande et le réglage de systèmes non linéaires ou les systèmes décrits par des modèles complexes incertains [AbbA09] [BoA12] [AMes11].

Sur le plan conduite en temps réel des applications gérées à base de techniques intelligentes, on retrouve qu'avec l'avancement technologique dans le domaine de la microélectronique, de nouvelles solutions de conception matérielles telles que les FPGAs (Field Programmable Gate Array) ou les ASICs (Application Specific Integrated Circuit) sont disponibles et peuvent être utilisées comme cibles numériques pour l'implémentation des algorithmes de commande dans un seul et unique composant [ErMo07] [MoNa07b]. Les avantages d'une telle implémentation sont multiples : la réduction du temps d'exécution en adoptant le traitement parallèle, le prototypage rapide de la commande numérique sur FPGA, la confidentialité de l'architecture, la possibilité d'application des commandes intelligentes font recours à des techniques plus lourdes en termes de temps de calcul et l'amélioration de la qualité de la commande des machines électriques en exploitant les nouvelles technologies de systèmes numériques. De nos jours, l'ensemble de ces avantages forment un besoin et une nécessité pour la commande des machines électriques caractérisées par de hautes performances.

En général, les actionneurs électriques les plus utilisés dans la majorité des applications industrielles sont construits autour de la machine asynchrone à cage. Cette dernière se distingue en particulier par sa robustesse, sa fiabilité, son faible coût et elle ne nécessite pas un entretien régulier. Cependant, son comportement dynamique est souvent très complexe, vu que sa modélisation aboutit à un système d'équations non linéaire, fortement couplé et multi variables. De plus, certaines de ses variables d'état, comme les flux ne sont pas mesurables. Ces contraintes exigent des algorithmes de commande plus avancés pour contrôler en temps réel le couple et le flux de ces machines. Plusieurs stratégies de commande ont été proposées dans la littérature pour atteindre cet objectif. Vers le milieu des années 80, une stratégie de contrôle de la machine asynchrone, connue sous le nom de Commande Directe de Couple ou DTC (Direct Torque Control) est apparue pour concurrencer les commandes conventionnelles. Cette commande a été introduite par TAKAHASHI [TakI89] et M. DEPENBROCK [DepM88]. Son principe est basé sur une détermination directe des impulsions de commandes appliquées aux interrupteurs de l'onduleur de tension, afin de maintenir le couple électromagnétique et le flux statorique à l'intérieur de deux bandes à hystérésis prédéfinies. Une telle application de cette technique permet d'assurer un découplage entre le contrôle du couple et le flux sans la nécessité d'utiliser une modulation de largeur d'impulsions (MLI), ni une transformation de coordonnées.

En outre, les principaux avantages de la DTC sont la rapidité de la réponse dynamique du couple, la dépendance vis-à-vis des paramètres du rotor et l'absence des transformations de coordonnées [AnMa04], [BeSe11a]. Cependant, cette stratégie présente deux inconvénients majeurs : d'une part, la fréquence de commutation est fortement variable et d'autre part, l'amplitude des ondulations du couple et du flux statorique restent mal maîtrisée dans toute la gamme de vitesse du fonctionnement envisagée. Il faut souligner que les ondulations du couple engendrent des bruits et des vibrations supplémentaires et par conséquent causent la fatigue au niveau de l'arbre en rotation. Pour réduire d'avantage la répercussion de ces phénomènes sur la durée de vie des actionneurs électriques, on estime que les techniques intelligentes permettent d'apporter une amélioration.

Dans ce contexte, le thème de recherche développé dans cette thèse concerne principalement l'exploitation des nouvelles solutions technologiques pour implémenter des commandes intelligentes à base de la Commande Directe de Couple d'une machine asynchrone autour d'un environnement matériel basé sur un FPGA. Cette implémentation vise principalement la réduction des ondulations au niveau du couple électromagnétique et du flux statorique. Dans cette partie, les deux régulateurs à hystérésis et la table de commutation de Takahashi seront remplacé par un contrôleur intelligent. Le principe de commande DTC de la machine asynchrone est basé sur l'utilisation de son modèle établi dans le référentiel lié au stator faisant intervenir la chute de tension introduite par la résistance statorique. Pour améliorer la qualité de la reconstitution du flux statorique, on a développé deux observateurs non linéaires de flux : un observateur à mode glissant et un observateur à grand gain. Chaque observateur sera intégré dans la chaîne de commande DTC à la place de l'estimateur. Les différents travaux concernant la commande DTC intelligente font l'objet de cinq chapitres.

Le premier chapitre résume en premier lieu l'essentiel de l'état de l'art relatif aux travaux de recherche menés autour de la commande DTC. La variété et la richesse des sujets traités justifient l'intérêt de cette stratégie de commande dans divers champs d'application. Ensuite, on présente en second lieu, un état de l'art sur l'implémentation matérielle des algorithmes de commande des machines asynchrones. En plus on donne l'environnement et la démarche pour la description, simulation et la synthèse afin de réussir l'implémentation de la commande de machines asynchrones sur cible FPGA.

Le deuxième chapitre est scindé en trois parties : on présente dans la première les modèles mathématiques de la machine asynchrone. Ces modèles sont définis généralement, dans un référentiel diphasé, soit tournant, soit fixe lié au stator. Quand à la deuxième partie, on rappelle le principe de la commande directe de couple de Takahashi appliquée à la machine asynchrone. Dans cette partie, deux approches d'observation du flux statorique de la machine asynchrone sont présentées et développées. Ces deux observateurs sont : observateur à grand gain et observateur à mode glissant. Les observateurs permettent de reconstruire les variables d'état qui ne sont pas mesurables. Ces méthodes permettent essentiellement de réduire le coût de fabrication, de maintenance et de proposer une solution dégradée mais fonctionnelle aux applications avec capteurs en cas de panne de ceux-ci. Une étude comparative des résultats de

simulation de la commande DTC avec observateur et estimateur du flux statorique d'une machine asynchrone fait l'objet de la troisième partie.

Le troisième chapitre est consacré à la présentation de l'ensemble des techniques intelligentes utilisées tel que la logique floue et les réseaux de neurones afin d'améliorer certaines performances de la commande DTC de la machine asynchrone. Ce chapitre se décompose en deux parties. Dans la première partie, on s'intéresse à la commande directe de couple à base d'un contrôleur flou. Il paraît nécessaire de commencer par présenter la logique floue et d'explicitier de manière simple sa mise en œuvre. Ensuite on a développé la commande directe floue de couple (DTFC) de la machine asynchrone. Enfin, une étude comparative entre les résultats de simulation de la DTFC et celle de la commande directe conventionnelle est présentée et discutée. Quand à la deuxième partie, elle est consacrée à la commande DTC à base d'un réseau de neurones. On commence cette partie par présenter une description théorique des réseaux de neurones artificiels, ensuite on a développé la commande directe neuronale de couple (DTNC) de la machine asynchrone. Cette partie s'achèvera par une étude comparative entre les résultats de la DTNC et la commande directe conventionnelle. Ce chapitre est clôturé par une étude comparative entre la commande DTFC et la commande DTNC.

La première partie du quatrième chapitre est destinée à l'étude de l'environnement et la méthodologie de conception d'une architecture matérielle pour implémenter la commande directe de couple d'une machine asynchrone sur cible FPGA. La méthodologie de conception s'appuie sur le concept d'adéquation algorithme architecture. On présente dans la première partie le flot de conception adopté, ensuite la modélisation et la spécification fonctionnelle de haut niveau de la chaîne de commande. Enfin, on donne la méthodologie suivie et les outils utilisés lors de la simulation et la synthèse pour aboutir à l'implémentation de la commande DTC d'une machine asynchrone à base d'un module matériel pour cible FPGA. La deuxième partie concerne l'implémentation de la commande directe floue de couple sur FPGA. Une étude détaillée sur la description matérielle sous VHDL des modules qui composent le système d'inférence flou est présentée et discutée. Ensuite, on présente la description sous VHDL et les résultats de synthèse de la Commande Directe Neuronale de Couple de la machine asynchrone afin de l'implémenter sur une plate-forme à base d'un module matériel FPGA.

Le cinquième chapitre est consacré à l'implémentation expérimentale des algorithmes développés dans les chapitres précédents afin de commander la machine asynchrone sans capteur mécanique. Ce chapitre débute par une description de la maquette expérimentale à base de cible FPGA. Cette maquette expérimentale servira par la suite pour l'implémentation expérimentale des différentes stratégies de commande et pour la mise en évidence de l'apport des FPGAs dans ce domaine. On présente ensuite les résultats de l'implémentation de la commande DTC de la machine asynchrone pour deux types de reconstruteur de flux. Le premier type : la commande DTC est basée sur un estimateur du flux statorique. Le deuxième type : la commande DTC est basée sur un observateur par mode glissant. Enfin, on présente les résultats de l'implémentation de la Commande Directe Floue de Couple de la machine

asynchrone. Une étude comparative entre les différentes stratégies de commande de la machine asynchrone clôturera ce chapitre.

Ce mémoire de thèse s'achève par une conclusion générale sur l'ensemble de mes travaux de recherches, en proposant des perspectives à court et à moyen terme.

# **Chapitre 1 : Etat de L'art de la Commande DTC de la Machine Asynchrone**

## 1.1 Introduction

Au cours de ces dernières décennies, l'avancement technologique dans le domaine de l'électronique de puissance et les progrès de l'informatique industrielle, permettent de développer des stratégies de commande avancées pour les machines électriques. Afin de positionner notre étude dans le domaine de la commande DTC, un état de l'art de la commande DTC est présenté en regroupant l'ensemble des publications qu'on a choisi pour aborder notre étude. Pour chaque publication, on résume la thématique de recherche et la possibilité d'avoir une validation expérimentale. Enfin, un état de l'art sur l'implémentation matérielle des algorithmes de commande des systèmes électriques sur FPGA est présenté. Dans cette partie, on donne la structure générale des FPGAs et on mettra l'accent sur la démarche pour la description et la simulation.

## 1.2 Commande Directe de Couple

La Commande DTC a été proposée par Takahashi [Tak189] au milieu des années 80. Comparée à la commande vectorielle, la commande DTC est moins sensible aux variations paramétriques de la machine et permet d'obtenir une dynamique précise et rapide du couple. Le principe de cette commande est de contrôler directement le couple et le flux statorique de la machine. Dans ce cadre, deux comparateurs à hystérésis sont utilisés et qui permettent de comparer les valeurs estimées avec celles de références, ensuite on commande directement les états de l'onduleur à travers une table de sélection prédéfinie. La stratégie de la Commande DTC de la machine asynchrone est illustrée par la figure 1.1.

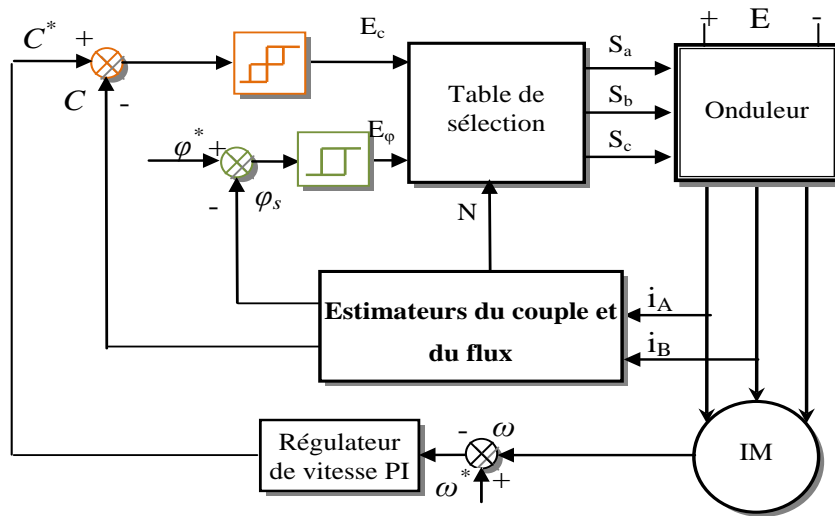


Figure 1.1. Stratégie de Commande Directe de Couple

Les principaux éléments constitutifs de la structure de commande DTC avec asservissement de vitesse sont les suivants :

- Deux estimateurs du couple et du flux statorique basés sur le modèle lié au stator,
- Une table de sélection du vecteur tension désiré du stator, établie en concordance avec les erreurs du flux et du couple engendrées,
- Deux comparateurs à hystérésis l'un à deux niveaux destiné pour le contrôle du flux, l'autre à trois niveaux consacré au contrôle du couple électromagnétique,

- Un régulateur de vitesse.

On rappelle dans la suite de ce paragraphe les avantages et les inconvénients de la stratégie de commande DTC conventionnelle objet des travaux de recherche présentés dans cette thèse [AnMa04], [MoHa09] :

### 1.2.1 Avantages de la Commande DTC

Parmi les avantages de la Commande DTC on peut citer :

- Une excellente dynamique du couple,
- Une bonne robustesse vis-à-vis des variations des paramètres du rotor de la machine,
- Absence d'utilisation des transformations de Park et de son inverse,
- Absence des blocs de calculs de la modulation de tension MLI,
- Absence de découplage des courants par rapport aux tensions de commande,
- Absence de nécessité de connaître l'angle de position rotorique. Seul le secteur dans lequel se trouve le flux statorique est nécessaire,
- Absence de nécessité d'utiliser un capteur de vitesse pour l'implémenter.

### 1.2.2 Inconvénients de la Commande Directe de Couple

Les inconvénients de la Commande DTC peuvent être résumés par les points suivants :

- L'existence des ondulations souvent importantes de couple et de flux,
- La fréquence de commutation n'est pas contrôlée,
- La nécessité d'utiliser des estimateurs de flux statorique et de couple,
- Un fonctionnement mal contrôlé à basse vitesse,
- Les courants statoriques sont mal contrôlés en régimes transitoires.

Récemment, plusieurs études ont été développées pour la commande DTC afin de palier à ses inconvénients. Beaucoup d'entre eux [RiAl02], [FaMe10] et [RaTa11] ont introduits des observateurs d'état pour corriger les insuffisances enregistrés au niveau de l'estimateur. D'autres études [MaZe05] et [RAbd11], ont utilisé la technique de *SVM* (*Space Vector Modulation*) dans la commande DTC. Ces études ont comme objectif de réduire les ondulations de couple, de flux en régime permanent. De même le contrôle de la fréquence de commutation est abordé. On a jugé utile de consacrer le paragraphe suivant à la présentation de l'état de l'art portant sur la commande DTC de la machine asynchrone.

## 1.3 Etat de l'art sur la Commande DTC de la machine asynchrone

Durant les deux dernières décennies, de nouvelles stratégies de commande DTC sont développées. Ces stratégies se basent sur le même principe de contrôle instantané du flux statorique et du couple ainsi que sur la bonne sélection d'un vecteur tension à partir de la table de *Takahashi*. Toutefois, ces nouvelles stratégies nécessitent une implémentation numérique afin de les valider. Cette dernière est effectuée sur une plate forme de commande de hautes performances à base de composant programmable telle que le microprocesseur, le microcontrôleur, le DSP ou le composant reconfigurable telle que le FPGA. Dans la plupart de ces nouvelles stratégies, les comparateurs à hystérésis sont écartés et remplacés par de nouveaux contrôleurs. Ces stratégies ont été proposées dans le but d'améliorer les

performances de la commande conventionnelle et de permettre un contrôle de la fréquence de commutation de l'onduleur.

La majorité des structures de commande DTC se regroupent en deux grandes familles à savoir :

- Les stratégies DTC à fréquence de commutation contrôlée.
- Les stratégies DTC à fréquence de commutation non contrôlée.

### 1.3.1 Les stratégies DTC à fréquence de commutation contrôlée

Plusieurs auteurs [BeSe11b] [MaZe05] [RAbd11] ont utilisé la technique de *SVM* (*space vector modulation*) dans la commande de l'onduleur. Le principe consiste à imposer le vecteur de tension adéquat via une modulation vectorielle d'espace. L'algorithme de commande s'avère plus complexe, mais les oscillations de flux et de couple sont réduites. Les auteurs de [BeSe11b], ont développé un algorithme qui permet d'obtenir une fréquence de modulation constante. Cette stratégie est caractérisée par l'élimination de la table de sélection de Takahashi et des comparateurs à hystérésis. Dans ce cadre, la technique MLI vectorielle est utilisée pour générer le vecteur de sortie de la commande. L'objectif de cette stratégie est de réaliser un contrôle direct du vecteur flux statorique dans un repère  $(\alpha, \beta)$  lié au stator. Les composantes de projection du vecteur tension statorique désiré sur les deux vecteurs de tension adjacents du repère  $(\alpha, \beta)$  permettent le calcul des temps de commutations désirés.

La stratégie de contrôle direct de couple de la machine asynchrone proposée dans [DePi02] et [JeMa98] est basée sur la modulation de largeur d'impulsion (PWM). Elle est développée en temps discret, afin de permettre la mise en œuvre sur une plate forme à base de microcontrôleur ou DSP. Cette méthode de contrôle nécessite la connaissance de la résistance statorique et l'inductance de fuite, ainsi qu'elle exige des mesures de courant et de tension. Des essais expérimentaux ont été réalisés afin de valider la stratégie proposée.

Dans le même cadre, les travaux de [YuPa10] ont utilisé la modulation vectorielle d'espace pour la commande DTC. Les auteurs ont utilisé un régulateur hybride proportionnel-intégral PI/flou pour la boucle de vitesse et deux régulateurs PI pour les boucles du couple et du flux pour construire le vecteur tension de référence utilisé par la technique SVM-DTC. Cependant, les auteurs de [JaPu11], ont développé une stratégie DTC à base d'un estimateur MRAS associé à un régulateur hybride PI/flou. Par comparaison à la stratégie DTC conventionnelle, les performances de la méthode proposée sont meilleures grâce à l'utilisation de la logique floue. Le flux statorique et le couple sont directement contrôlés par le vecteur tension généré par l'onduleur dont le contrôle est assuré par une MLI vectorielle.

Les auteurs de [DoCa00] et [HrKe04] ont montré que le nombre de vecteurs de tension appliqué à la machine pour la DTC conventionnelle est faible. Ces travaux montrent aussi que l'utilisation de la modulation vectorielle d'espace dans la Commande DTC permet d'une part de résoudre en partie le problème des ondulations du flux et du couple, et d'autre part de contrôler la fréquence de commutation des interrupteurs de l'onduleur. De cette manière, la commande DTC à base de SVM et en utilisant un comparateur de couple de cinq niveaux permet de produire un nombre de vecteurs tension plus élevé. Les performances de la stratégie



proposée ont été validées par des simulations et des essais expérimentaux. Ces performances montrent bien une réponse améliorée de flux et de couple avec une fréquence de commutation fixe.

Dans [CaMa02], les auteurs ont proposé la commande DTC à fréquence de commutation imposée. Dans cette stratégie, le vecteur tension choisi se déplace dans le plan de phase de façon adjacente. Ces déplacements assurent la minimisation des dérivés de tension : de cette manière, les ondulations du couple et du flux sont réduites et la fréquence de commutation est convenablement contrôlée. Les résultats de simulation présentés dans ce travail montrent bien des améliorations au niveau des performances de la stratégie de commande proposée comparant à la commande DTC conventionnelle.

Dans [MaHa10], les auteurs ont présenté en premier lieu, une approche basée sur le modèle de la machine asynchrone. Ils ont montré que les insuffisances de cette approche résident au niveau des paramètres du moteur qui doivent être identifiés correctement. En deuxième lieu, une nouvelle approche de la DTC à base d'un modèle de minimisation des pertes combinée à la technique de backstepping est présentée et étudiée. Le contrôleur basé sur la technique de backstepping a comme entrée l'erreur du couple électromagnétique et l'erreur du flux rotorique. Le contrôleur proposé est exprimé dans le référentiel de Concordia. Il a une capacité de poursuite rapide du flux rotorique ainsi que du couple électromagnétique. Les auteurs ont également introduit un observateur mode glissant du flux rotorique. Cet observateur permet de déterminer simultanément la constante du temps rotorique, le flux rotorique et la vitesse de rotation. Le contrôleur proposé est implémenté sur une plate forme à base d'un FPGA. Les résultats de simulation ainsi que les résultats expérimentaux sont présentés afin de vérifier l'efficacité de la méthode proposée.

Dans sa thèse [CaMa00], l'auteur a proposé deux stratégies de contrôle direct du couple, compatibles avec des onduleurs de tension multi-niveaux. Ces deux stratégies sont définies d'une manière généralisé par rapport au nombre de niveau de l'onduleur. L'algorithme de ces stratégies, comporte les modèles prédictifs de la machine afin d'exploiter de manière optimale les nombreux degrés de liberté offerts par les structures multi-niveaux. Dans ces travaux, la fréquence de commutation est bien maîtrisée ainsi que le contenu harmonique. En particulier, la deuxième stratégie, désignée par CoDiFI (Contrôle Direct à Fréquence de commutation Imposée) combine ces deux facteurs, tout en gardant les performances dynamiques attribuées à la stratégie DTC conventionnelle. Cette stratégie CoDiFI est validée expérimentalement sur un onduleur à 4 niveaux, alimentant une machine asynchrone via une architecture hybride DSP/FPGA. Une étude comparative entre la DTC conventionnelle et les stratégies proposées est réalisée, tout en présentant l'intérêt d'une association entre les lois de commande DTC et les onduleurs multi-niveaux.

Dans la thèse de [BeSe11a], l'objectif était d'améliorer les performances de la commande DTC d'une machine asynchrone. Cette amélioration concerne la réduction des oscillations de couple et de flux, la commande à une fréquence de commutation constante, ainsi que la robustesse de la DTC vis-à-vis la variation de la résistance statorique. Dans cette thèse, l'auteur a remplacé la table de sélection de *takahashi* et les deux comparateurs à hystérésis par

un contrôleur PI prédictif. Afin d'avoir une commande à fréquence de commutation constante, l'auteur a ajouté un module MLI vectorielle à la commande DTC. Les propriétés de robustesse de la commande obtenue ne sont pas garanties face aux incertitudes paramétriques. Pour remédier à ce problème, l'auteur a intégré la commande adaptative comme amélioration pour la stratégie DTC-SVM.

Dans [VAmb04], l'auteur a proposé une nouvelle stratégie de Commande DTC de la machine asynchrone afin de réduire les ondulations importantes au niveau de couple et de flux. Dans cette nouvelle stratégie, le vecteur tension de l'onduleur sélectionné de la table de *Takahashi*, est appliqué pendant un intervalle de temps  $t$ . Cet intervalle du temps est défini comme étant le temps nécessaire pour que l'erreur couple atteigne la limite inférieure ou la limite supérieure de la bande d'hystérésis. Cette technique permet au système de commande de reproduire le fonctionnement de l'hystérésis pendant l'intervalle du temps  $t$  et pas durant toute la période. Les résultats de simulation et les résultats expérimentaux sur une carte à base d'un DSP, ont illustré une réduction considérable des ondulations du couple.

En outre, plusieurs chercheurs proposent de nouvelles stratégies de la DTC. Ces stratégies permettent de fixer la fréquence de commutation, afin de réduire les ondulations au niveau du couple et du flux. *Idris et al*, ont proposé dans [NrId03], de remplacer les comparateurs à hystérésis par deux régulateurs de flux et de couple. Le régulateur de flux est synthétisé à base d'un régulateur linéaire type PI. Un comparateur entre la sortie du régulateur et un signal triangulaire de haute fréquence et une fonction seuil. Par contre, le régulateur de couple est conçu aussi autour d'un régulateur PI, deux comparateurs, deux signaux triangulaire de haute fréquence et deux fonctions seuil. La sortie de la fonction seuil du régulateur du flux et la somme des deux sorties des fonctions seuil du régulateur du couple fournissent les paramètres de sélection de la table de *Takahashi*. L'implémentation matérielle de ces régulateurs sur FPGA est très convenable puisque seulement des comparaisons et des sommes sont utilisées dans cette approche. Les résultats expérimentaux illustrent la réduction des ondulations tout en gardant la fréquence de commutation fixe. Dans le même contexte, les auteurs de [SaZa09] ont proposé une nouvelle approche permettant d'obtenir une réponse rapide en couple obtenu pour une fréquence de commutation fixe. Cette approche se base sur l'optimisation de la sélection des vecteurs tensions pour augmenter le couple au maximum. Les résultats de simulation et l'implémentation matérielle montrent bien que la réponse de couple et/ou de vitesse est très rapide comparant à la commande conventionnelle.

*Bojoi et al*, ont présenté dans [RBoj05] l'étude de la commande DTC d'une machine asynchrone triphasée à double stator. Le stator de la machine asynchrone est composé par deux enroulements triphasés décalés d'un angle électrique égal à  $\pi/6$ . La commande DTC est basée sur un algorithme prédictif et implémentée dans le repère  $\alpha$ - $\beta$  lié au stator. Les avantages de cette stratégie de commande avec une machine asynchrone à double stator sont : fréquence de commutation fixe, bonne performance du couple et du flux en régimes transitoire et permanent, faible distorsion du courant statorique. Les résultats expérimentaux ont été menés sur une machine asynchrone à double stator de 10 kW.

### 1.3.2 Les stratégies DTC à fréquence de commutation non contrôlée

Afin de palier à limitations enregistrées lors de l'utilisation des onduleurs à 2 niveaux, les auteurs de [FaKh11] ont présenté une étude comparative des onduleurs multi niveaux pour la commande DTC d'une machine asynchrone. Cette étude concerne les onduleurs à cinq et sept niveaux symétriques et asymétriques. L'objectif de cette étude est de chercher la tension de sortie la plus adéquate pour minimiser les pertes par commutation, ce qui réduit les ondulations au niveau du couple et du flux. L'étude expérimentale a montré que la configuration asymétrique fournit une tension de sortie presque sinusoïdale avec une faible distorsion, tout en réduisant les pertes de commutation ainsi que la réduction des ondulations de couple

Dans son mémoire d'habilitation [AbYa12], l'auteur a présenté en premier lieu une analyse des performances de trois stratégies DTC, dédiées au contrôle de vitesse en considérant l'association machine asynchrone-onduleur de tension à deux niveaux et à trois bras. Il s'agit de : (i) la stratégie conventionnelle de *Takahashi*, (ii) la stratégie basée sur la variation du couple électromagnétique par rapport au temps en considérant six vecteurs actifs et (iii) la stratégie ayant le même fondement mathématique que la seconde stratégie mais en considérant douze vecteurs actifs. L'étude comparative entre ces trois stratégies DTC à fréquence de commutation non contrôlée a montré que le phénomène de démagnétisation à basse vitesse apparait dans la stratégie conventionnelle. Cependant, ce phénomène est écarté dans les deux autres stratégies. En deuxième lieu, l'auteur a présenté une étude sur l'amélioration des performances des stratégies DTC pour le contrôle de position. Cette amélioration est effectuée en subdivisant le plan de Concordia en vingt secteurs en faisant intervenir un régulateur à hystérésis de cinq niveaux pour le contrôle du couple et avec un usage adéquat des vecteurs tensions pleines. Ce mémoire est clôturé par une étude expérimentale des stratégies DTC dédiées au contrôle de vitesse et de position.

Récemment, une autre catégorie de commande basée sur l'intelligence artificielle est présentée dans la littérature. Ces commandes proposent d'améliorer les performances dynamiques de la commande DTC ; soit en adaptant la bande d'hystérésis [IbOk10], [NaUd12], soit en remplaçant la table de sélection de *Takahashi* ainsi que les hystérésis par des régulateurs intelligents [DeJi06], [RiTo08], [SmGa09], [HaHa12], [HaLi10]. L'intelligence artificielle comme la logique floue, le réseau de neurones et l'algorithme génétique, a connu un grand succès non seulement dans la modélisation mais aussi dans la commande des systèmes électriques, en particulier la commande des machines asynchrone. Ce succès est dû à ce que l'intelligence artificielle peut se rapprocher facilement du comportement de contrôle de l'expert humain qui travaille dans des environnements souvent mal définis.

Dans l'article [SmGa09], une étude comparative entre quatre régulateurs de vitesse intelligents pour la commande DTC est présentée et discutée. Ces quatre régulateurs sont : un contrôleur PI à base de la logique floue, un contrôleur PI à base d'algorithme génétique, un contrôleur flou et un contrôleur mode glissant flou. Un indice de performance basé sur l'erreur de la vitesse est choisi comme critère afin de comparer tous ces régulateurs. Chaque stratégie de contrôle a été testée pour évaluer sa robustesse vis-à-vis des paramètres de la

machine et sa capacité du rejet des ondulations et des distorsions. Les résultats de simulations menés dans cet article ont montré que les contrôleurs intelligents sont plus robustes contre les variations paramétriques de la machine comparés aux contrôleurs à base de PI. Le contrôleur mode glissant flou paraît le meilleur choix pour la régulation de vitesse, mais nécessite des ajustements pour réduire le phénomène de chattering dans l'état permanent.

La variation de la fréquence de commutation des interrupteurs d'un onduleur est due essentiellement à l'utilisation des hystérésis de couple et de flux dans la commande DTC. L'auteur *Ibrahim Okumus* a proposé dans l'article [IbOk10] une stratégie de contrôle adaptative en temps réel de la bande de l'hystérésis. De cette façon les ondulations de couple se réduisent et la fréquence de commutation tend vers une fréquence fixe. La stratégie de contrôle de la bande de l'hystérésis adaptative proposée est vérifiée par les simulations sous Matlab. Les résultats expérimentaux réalisés sur une plate-forme à base d'un processeur DSP TMS320C6711, prouvent la faisabilité de la stratégie proposée comparée avec les méthodes conventionnelles. Dans ce même contexte, *Nasir Uddin* dans [NaUd12] a proposé un contrôleur flou pour ajuster la bande d'hystérésis en temps réel. Cet ajustement est basé sur les pentes de la variation du couple estimé et du courant statorique, ou le contrôleur flou sélectionne la bande passante optimum de l'hystérésis du couple. Les résultats de simulation sur un modèle de simulation développé sous MATLAB/Simulink, ainsi que les résultats expérimentaux réalisés avec un DSP prouvent les performances obtenues par ce contrôleur flou. Une étude comparative entre la DTC basée sur le contrôleur flou proposé et la DTC conventionnelle montre que l'ondulation de couple de la commande proposée a été réduite considérablement.

Dans l'article [DeJi06], les auteurs ont proposé une stratégie de commande DTC basée sur la logique floue. L'objectif des travaux présentés est d'améliorer les performances de la commande DTC tout en minimisant les ondulations de couple à basses vitesses. Ils ont intégré un régulateur flou de vitesse qui permet d'ajuster dynamiquement le coefficient de proportionnalité  $k_p$  et le coefficient d'intégration  $k_i$  en fonction de l'erreur et des variations de la vitesse. De plus, les hystérésis de couple et de flux sont remplacés par un régulateur flou afin d'optimiser la sélection du vecteur tension. Les résultats expérimentaux montrent que le système de contrôle flou proposé peut assurer une réponse rapide et une haute précision de la vitesse en régime permanent, ainsi que la réduction remarquable des ondulations du couple même à basses vitesses. En outre, l'introduction de la logique floue améliore la précision de l'observation du flux statorique qui augmente les performances du système entier.

Dans les travaux de thèse de [HaHa12], l'auteur a mis l'accent sur les performances de la stratégie DTC de *Takahashi* et aussi sur ses manquements. Il a commencé par développer trois estimateurs de vitesse, à savoir : (i) un estimateur basé sur le flux orienté, (ii) un estimateur basé sur l'approche du système adaptatif à modèle de référence (MRAS), et (iii) un estimateur basé sur un réseau de neurones artificiels. Ensuite, il a enchaîné par une étude comparative de leurs performances à travers des travaux de simulation. Lors du fonctionnement à grandes vitesses et sous le couple nominal, ces trois estimateurs ont montré une bonne convergence. Tandis que, à basse vitesse la réponse de l'estimateur basé sur le flux orienté présente des erreurs d'estimation. Cependant, l'estimateur basé sur un réseau de

neurones présente la meilleure précision, couvrant une large plage de vitesse. L'estimateur basé sur l'approche MRAS se classe en deuxième position avec une précision relativement affectée à basse vitesse. Enfin, l'auteur a présenté sa contribution pour améliorer les performances de la stratégie DTC, afin de pallier à ses manquements. Il a remplacé l'estimateur du flux statorique, vu qu'il est sensible à la variation de la résistance statorique, par un observateur à mode glissant.

L'objectif principal de cette thèse [RiTo08] est d'introduire des stratégies d'amélioration des performances de la DTC, capable de fournir une bonne dynamique du couple électromagnétique avec une robustesse contre les variations paramétriques sans aucun capteur. L'auteur de cette thèse a commencé par étudier le régime transitoire magnétique de la commande et il a élaboré une correction des phénomènes d'ondulations qui apparaissent à basses vitesses. Ensuite, il a abordé l'analyse de l'influence du terme résistif sur le comportement de la commande et il a élaboré la méthode de décalage des zones pour améliorer l'établissement du flux statorique. En outre, l'auteur a étudié l'effet de la variation de la résistance statorique sur les performances de la commande DTC. Il a proposé d'autres stratégies d'amélioration basées sur la commande DTC à savoir la DTC avec un onduleur 3 Niveaux, la DTC à 12 secteurs et la DTC basée sur la MLI vectorielle. En fin, l'auteur a intégré les techniques intelligentes dans la commande DTC. Il a remplacé la table de *Takahashi* et les deux hystérésis par un contrôleur flou, neuronal et neuro-flou. La validité de ces techniques proposées a été prouvée par les résultats de simulation. Le contrôleur flou et le contrôleur neuro-flou ont donné des meilleures performances que celles obtenues par les autres stratégies proposées.

Les auteurs de [HaLi10], ont proposé une nouvelle stratégie de commande DTC, en utilisant l'algorithme génétique afin d'optimiser le régulateur PI-Flou. Dans cette stratégie, en fonction de l'erreur vitesse et de sa dérivée par rapport au temps, l'ajustement du coefficient intégral  $k_i$  et proportionnelle  $k_p$  est réalisé en temps réel par un régulateur adaptatif PI-flou de vitesse. Les paramètres flous sont raffinés par l'algorithme génétique afin d'améliorer l'auto-adaptation de la vitesse. De plus, les régulateurs d'hystérésis ont été remplacés par un autre régulateur flou afin d'améliorer le choix du vecteur de tension. Enfin l'auteur a présenté une étude comparative entre la DTC conventionnelle de *Takahashi*, le régulateur PI-Flou et la stratégie proposée. Cette étude a prouvé la diminution importante des ondulations au niveau du couple, flux, et courant. Ainsi que l'amélioration de la précision et le suivi de vitesse.

A l'issu de cette synthèse bibliographique sur la stratégie de commande DTC et son implémentation sur un système de développement à base d'FPGA, disons en général que, les problématiques liées en particulier d'une part à la présence des ondulations de couple et de flux statorique et d'autre part à la fréquence de commutation qui reste assez élevée sont loin d'être complètement résolus. Dans ce cadre, les travaux proposés dans cette thèse sont directement orientés vers l'application des techniques intelligentes afin de pallier à certains aspects de la commande DTC. Le premier aspect, auquel on a investigué consiste à contourner en partie la forte ondulation au niveau des grandeurs couple électromagnétique et flux statorique. Pour ce faire, on propose de remplacer les comparateurs à hystérésis et la table de commutation par des contrôleurs intelligents. Les performances de ces contrôleurs ont été

testées par simulation, le contrôleur flou a permis de réduire ces ondulations ainsi que la réduction de la fréquence de commutation. Le deuxième aspect concerne la sensibilité de l'estimation du flux statorique vis-à-vis des variations de la résistance statorique. Ainsi, on propose de développer deux approches d'observation du flux statorique. Ces deux observateurs non linéaires sont : à grand gain et par mode glissant avec adaptation de la résistance statorique. On a montré l'aptitude de l'observateur à garantir une bonne observation du flux statorique. Le dernier aspect de cette thèse est orienté vers l'implémentation de stratégies sur une plate forme à base d'un FPGA. Le paragraphe suivant est consacré à présenter un état de l'art sur l'implémentation matérielle sur les FPGAs.

## **1.4 Implémentation matérielle des algorithmes de commande sur FPGA**

### **1.4.1 Etat de l'art sur l'implémentation sur les FPGAs**

L'évolution récente des composants numériques de types FPGA et ASIC a permis d'envisager l'implémentation complète de commandes numériques de dispositifs d'Electronique de Puissance dans un seul et unique composant. Les avantages d'une telle implémentation sont multiples : la réduction du temps d'exécution en adoptant un traitement parallèle, le prototypage rapide de la commande numérique sur FPGA, possibilité d'application des commandes intelligentes font recours à des techniques plus lourdes en terme de temps de calcul et amélioration de la qualité de la commande des machines électriques en exploitant les nouvelles technologies de systèmes numériques.

Cette évolution a poussé plusieurs concepteurs d'architectures dédiées à la commande des systèmes électriques à exploiter la richesse des composants FPGA telle que la rapidité et la souplesse, afin d'améliorer les performances des algorithmes de commande. En effet, ils ont été utilisés pour le contrôle des machines asynchrones [ToLi05a] [JaLi08] [ToSu11] [ErMo11], la commande des robots [XiSh07] [FaPi11]. Dans [ErMo11], E. Monmasson a présenté un état de l'art de l'utilisation des FPGAs dans les systèmes industriels, ainsi que les outils de développement adéquat, afin d'obtenir les performances souhaitées. Dans la thèse [LoCh06], l'auteur propose une démarche méthodologique en vue de l'implémentation des algorithmes de commande de systèmes électriques sur des cibles d'implémentation matérielles de type FPGA. La première phase de la méthodologie est le raffinement algorithmique comportant les différentes étapes depuis le cahier des charges jusqu'à la spécification de l'algorithme en format virgule fixe. La deuxième phase est le développement architectural, basée sur l'emploi de différents concepts méthodologiques conduisant à la mise au point d'une bibliothèque des modules IP. Le flot de conception proposé est illustré par des exemples du domaine.

L'auteur des travaux [MoNa07a] et [MoNa07b] présente l'intérêt de l'exploitation des solutions matérielles à base des FPGAs comme support pour l'implémentation des algorithmes de commande de systèmes électriques. Une méthodologie de développement d'implémentation sur FPGA est présentée et discutée dans ces travaux. Cette méthodologie a permis le développement d'une bibliothèque de composants matériels réutilisables, dédiée à la commande de machines électriques. Elle a aussi permis l'implémentation sur FPGA de

plusieurs algorithmes de contrôle du courant d'une machine synchrone, telle que le contrôle par mode glissant, le contrôle par régulateur PI et le contrôle prédictif. Tout au long de ces travaux, la contribution à l'utilisation des FPGAs pour la réalisation de chaque technique de contrôle de courant sont étudiés, analysés et discutés.

Les papiers [RRaj10] et [RRaj12] présentent une nouvelle approche pour la conception et l'implémentation de la commande DTC avec SVM d'une machine asynchrone sur FPGA. L'auteur cite en premier lieu les inconvénients de la commande DTC conventionnelle qui sont : les ondulations importantes au niveau du couple et du flux statorique ainsi la variation de la fréquence de commutation. Afin de remédier à ces problèmes l'auteur propose dans ces papiers le concept de la DTC-SVM. Les résultats de simulation et les essais expérimentaux montrent que la méthode proposée a amélioré considérablement les performances de la commande au niveau du couple et de flux statorique tout en conservant les performances dynamiques de la DTC conventionnelle.

La thèse de BAHRI I. [ImBa11], s'intéresse à l'utilisation des plateformes FPGA pour l'implémentation des algorithmes d'entraînement électrique pour des applications avioniques. Ce domaine d'application est caractérisé par des difficultés techniques comme leur environnement de travail (température élevée, pression) et les exigences de performances (la flexibilité, le haut degré d'intégration). Durant cette thèse, l'auteur propose un contrôleur de vitesse qui doit fonctionner à 200°C de température ambiante. Ce contrôleur est basé sur une commande par flux orienté pour une Machine Synchrone. Une approche de développement et de validation a été proposée et testée sur une plate forme à base d'FPGA. D'un autre côté et pour profiter des avantages offerts par les systèmes sur puce (SoC), l'auteur propose une méthodologie de Co-conception matériel-logiciel pour le contrôle d'entraînement électrique. La validation expérimentale de ce contrôleur de courant a été mise en œuvre et les résultats obtenus prouvent l'intérêt de l'approche proposée.

### 1.4.2 Environnement de description

L'implémentation des algorithmes de commande de machines électriques sur des nouvelles solutions matérielles tels que les FPGAs est une démarche qui nécessite une parfaite maîtrise des processus de conception et un savoir faire méthodologique des concepteurs, afin de satisfaire les contraintes inhérentes de l'implémentation. Cependant, les applications de commande de machines électriques constituent un domaine multidisciplinaire. En effet, le concepteur doit maîtriser le domaine de commande des machines électriques, de l'électronique de puissance et de la microélectronique. D'un autre côté, le langage de description matérielle telle que VHDL utilisé pour le développement de l'FPGA, nécessite une bonne connaissance non seulement de l'algorithme à implémenter mais aussi de l'architecture de l'FPGA et du compilateur utilisé. Pour la plupart des concepteurs qui sont souvent des chercheurs ou des ingénieurs de développement en logiciel, ce langage matériel n'est pas familier et d'une utilisation parfois délicate. Afin d'apporter des solutions à ce problème, de nouveaux langages de conception de circuits de haut niveau basés sur le langage C, ont été développés. On peut citer comme exemple le Handel-C [Celo05] de Celoxica ou le langage System-C de Synopsys [Synopsys].

D'autres solutions de conception basées sur des environnements graphiques ont été proposées, à titre d'exemple, Labview [Labview] avec son langage de programmation graphique propose des outils de haut niveau, pour créer des applications souples, évolutives et sophistiquées, en particulier sur des circuits FPGA [TeOr11]. On peut citer aussi que l'environnement Matlab-Simulink [Matlab] plus XSG (Xilinx System Generator) [XSG11] offre un environnement complet pour le prototypage rapide des algorithmes à implémenter sur FPGA, en particulier les algorithmes de commande des machines électriques. XSG est une boîte à outils (toolbox) développé par Xilinx qui communique avec l'environnement Matlab-Simulink et qui laisse l'utilisateur créer des architectures sous forme de blocs graphiques de Xilinx interconnectés correspondant aux tâches de traitement. La figure 1.2 illustre l'architecture du bloc estimateur du flux statorique développé sous XSG. Une fois le système testé et validé, les blocs matériels peuvent être synthétisés et implémentés sur FPGA.

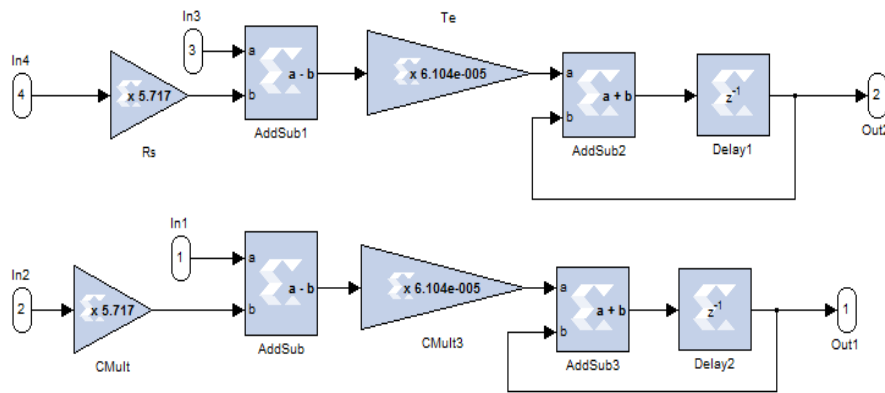


Figure 1.2. Schéma détaillé du bloc estimateur du flux statorique sous XSG

Dans ce contexte, [JeMa08] [OzAk10] [ErMo11] [RRaj12] et [BoAl12] ont proposé un flot de conception dans le but de développer des architectures de commande de machines électriques. Les étapes de développement de ces architectures à implémenter sont principalement effectuées via l'environnement Matlab-Simulink plus XSG ainsi que les outils CAO des solutions matérielles. L'avantage de la méthodologie de développement utilisée dans ces travaux est qu'elle est facile à appliquer par un concepteur Electrotechnicien sans qu'il soit spécialiste en microélectronique avec temps de développement réduit. En revanche, l'inconvénient majeur de cette méthodologie, basée sur l'environnement Matlab-Simulink plus XSG réside dans le fait que sans connaissance en architecture, le développeur ne peut pas atteindre les hautes performances matérielles en exploitant pleinement les ressources matérielles disponibles sur les FPGAs pour réaliser des implémentations performantes et optimales. En fait, il existe un compromis entre l'efficacité matérielle et le temps de développement [DeSa11]. Une durée de développement relativement importante et les connaissances spécifiques des experts développeurs sont nécessaires pour réussir à obtenir de hautes performances matérielles.

Compte tenu de ce qui précède et sur la base des méthodologies de conception récemment publiées dans la littérature [KhGh05] [LoCh06] [MoNa07b] [ErMo07] [ImBh11], on a choisi l'environnement VHDL comme étant le langage de la modélisation comportementale et structurelle de haut niveau, de simulation et de synthèse pour la commande DTC de la



machine asynchrone à implémenter, ainsi que pour les commandes intelligentes. Le langage VHDL est basé sur la notion de modèles temporels discrets et pourrait ainsi être utilisé pour la modélisation comportementale des éléments analogiques et de puissance, après avoir discrétisé les grandeurs et les modèles continus à mettre en œuvre [GirC99].

### 1.4.3 Configuration d'un FPGA

La figure 1.3 résume les différentes étapes de configuration d'un FPGA [MoNa07b]. Dans un premier temps, le synthétiseur génère une *Netlist* qui décrit la connectivité de l'architecture. Ensuite l'outil de placement et routage place tous les composants et exécute le routage entre les différentes cellules logiques. Un fichier de configuration appelé *bitstream* sera généré après ces deux étapes. Ce fichier est enfin implémenté sur FPGA précisément au niveau de la mémoire de configuration.

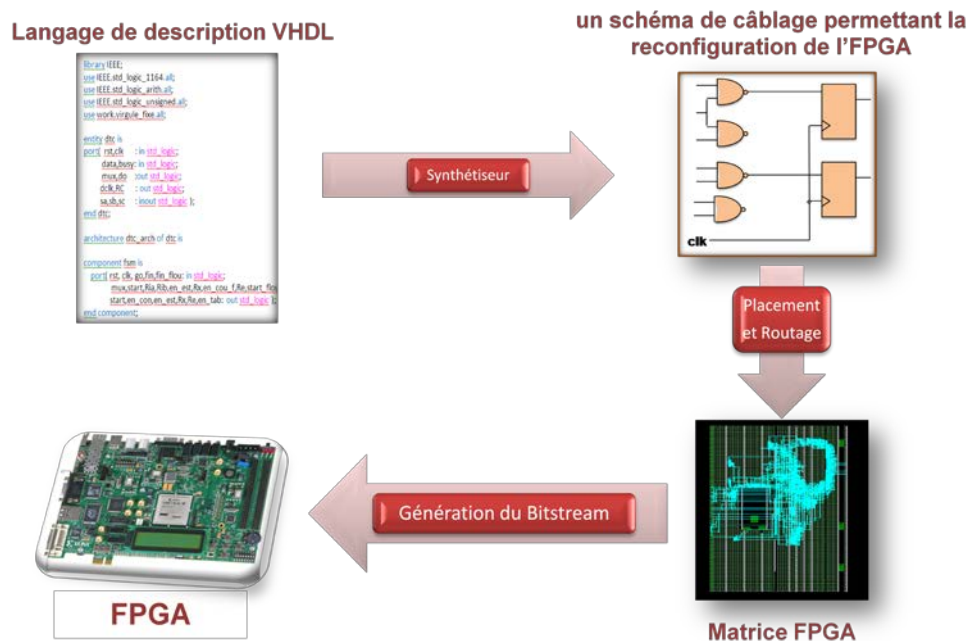


Figure 1.3. Configuration d'un FPGA.

Dans le paragraphe suivant, on va entamer l'implémentation sur FPGA des lois de commande DTC de la machine asynchrone. Cette implémentation sera effectuée tout en adoptant une méthodologie de conception.

### 1.4.4 Démarche pour la description et la simulation

Les environnements de développement (de simulation et de synthèse) permettent la conception du circuit en tenant compte de l'environnement de l'application, en particulier, le comportement des entrées/sorties du circuit. En effet, Il est inconcevable d'envisager la simulation d'une partie d'un système en ignorant l'influence des autres aspects sur son comportement global.

La structure adoptée, illustrée par la figure 1.4, est basée sur les différentes unités de conception. On utilise pour chaque description trois groupes de fichiers ayant chacun une structure bien définie [MtAb00] :

Le premier groupe est constitué d'un fichier principal dans lequel on élabore la spécification de l'entité et le corps de l'architecture qui contient le ou (les) processus actif(s) assurant le fonctionnement du circuit à concevoir G1, comme le montre la figure 1.4.a;

Le deuxième groupe est réalisé à partir d'un fichier appelé souvent "Test\_Bench". La spécification d'entité qui y est contenue est généralement décrite par des ports déclarés en sortie. Ces ports coïncident en partie avec ceux déclarés comme entrées dans le fichier principal. Dans le corps d'architecture de ce fichier, on envoie le stimulus pour le fichier principal G2, comme le montre la figure 1.4.b;

Le troisième groupe est constitué par un fichier, appelé souvent fichier de test. Il assure le lien entre les deux fichiers déclarés ci-dessus et les composants non synthétisables ou les modèles abstraits des composants externes (les capteurs de courant, de vitesse, etc.). C'est dans ce fichier qu'on spécifie la configuration de l'ensemble de l'application. Cette configuration fait appel aux différentes entités spécifiées précédemment.

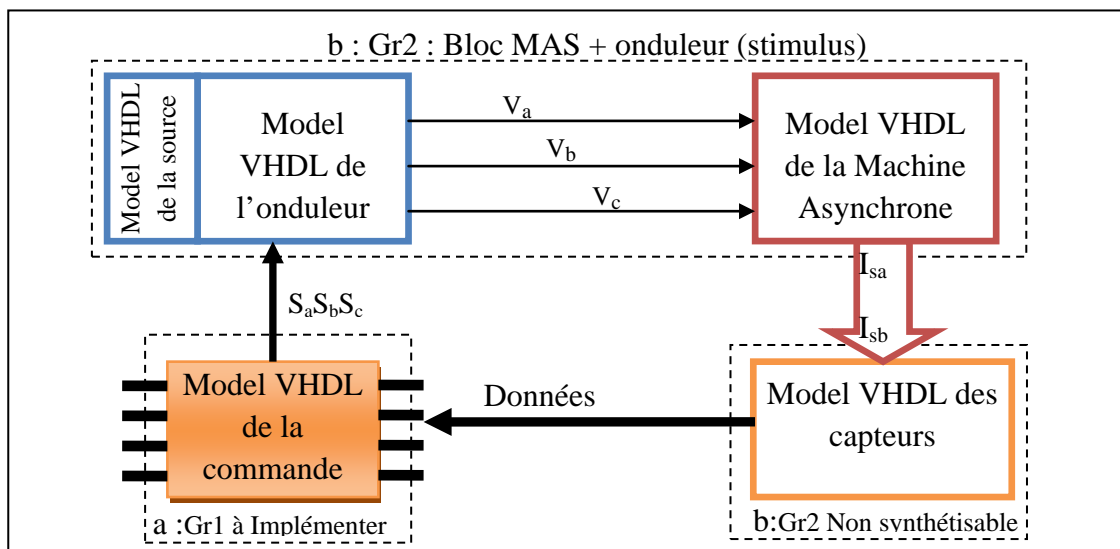


Figure 1.4. Environnement de description et de simulation

L'avantage d'une telle structure de travail est qu'elle nous permet d'éviter l'extraction ou l'insertion des composants synthétisables ou non synthétisables lors du passage d'un niveau d'abstraction à un autre niveau plus bas. En plus, le fichier "Test\_Bench" est utilisé, sans modifications, durant toutes les étapes de la simulation.

Ainsi, le VHDL permet de modéliser et de simuler le système de puissance complet. Ensuite, la commande DTC, décrite en VHDL peut alors être implantée indépendamment de la cible physique choisie, FPGA ou ASIC.

## 1.5 Conclusion

Dans ce chapitre, on a présenté en premier lieu la stratégie de commande DTC, qui a une excellente dynamique du couple et une bonne robustesse vis-à-vis des variations paramétriques de la machine. Ensuite, on a présenté une étude bibliographique se rapportant à

l'évolution dans le temps des stratégies DTC dédiées à la MAS. Cette étude a été menée en distinguant deux majeures classes : les stratégies DTC à fréquence de commutation contrôlée et les stratégies DTC à fréquence de commutation non contrôlée. La variété et la richesse des sujets traités justifient l'intérêt de cette stratégie de commande dans divers champs d'application.

En deuxième lieu, on a présenté un état de l'art sur l'implémentation matérielle des algorithmes de commande des machines asynchrones. Ensuite on a discuté et présenté l'environnement et la démarche pour la description, simulation et la synthèse afin de réussir l'implémentation de la commande de machines asynchrones sur cible FPGA.

Le prochain chapitre, est consacré à la présentation du principe de la commande DTC conventionnelle et à la mise en place de deux observateurs non linéaires de flux statorique.

## **Chapitre 2 : Modélisation et commande DTC de la Machine asynchrone**

## 2.1 Introduction

La commande à vitesse variable des entraînements électriques a bénéficié des évolutions méthodologiques et technologiques. Ainsi, la plupart des algorithmes de commande utilisés pour le contrôle de vitesse de la machine asynchrone reposent sur une bonne modélisation faisant intervenir ses paramètres principaux.

Dans ce cadre la première partie de chapitre est consacré à présenter de manière brève les modèles de la machine asynchrone couramment utilisés dans la littérature. Généralement, ces modèles sont définis dans des référentiels diphasés tournant et fixe lié au stator. Ensuite, on donne le principe et la structure de la stratégie de commande DTC de *Takahashi* de la MAS. Cette dernière, la commande DTC est basée sur un estimateur pour reconstruire le flux statorique. Dans la deuxième partie, deux types d'observateurs de flux statorique de la machine asynchrone ont été développés. Ces derniers sont de type non linéaires à grand gain et par mode glissant, conçus autour du modèle diphasé lié au stator. Il faut signaler que l'observation des grandeurs non mesurables permet en particulier de réduire le coût de fabrication, de maintenance et de proposer éventuellement une commande tolérante aux défauts dédiée aux applications avec ou sans capteurs [MaGh05]. Une étude comparative des résultats de simulation de la commande DTC avec observateur et estimateur de flux statorique de la machine asynchrone est explorée dans la troisième partie.

## 2.2 Modélisation de la machine asynchrone

### 2.2.1 Présentation de la machine

La majorité des applications industrielles ont été conçues autour de la machine asynchrone à rotor à cage. Ceci est du principalement à sa simplicité de construction, son faible coût, son entretien facile et sa robustesse. Cependant, elle est caractérisée par un modèle fortement non linéaire. Dans ce cadre, la mise en œuvre des stratégies de commande performantes pour cette machine nécessite la mise en place d'un modèle fiable et capable d'étudier les différents régimes à observer. Dans la littérature [CCar00] [Allm04] [LoBa04] [MiFa05] [MfMi07], les modèles dynamiques qui sont couramment utilisés dans l'élaboration des lois de commande et/ou dans la mise en place des systèmes d'observation sont établis respectivement dans le référentiel de Park tournant à la vitesse de synchronisme du champ tournant et de Concordia fixe lié au stator. Dans le cadre de cette thèse, on résume l'essentiel de ces deux modèles et on insiste sur les hypothèses simplificatrices adoptées.

### 2.2.2 Hypothèses simplificatrices

Les hypothèses simplificatrices couramment utilisées dans la mise en place des différents modèles de la MAS largement citées dans la littérature [JpCa95], [AdMe07], [Allm04]. Le modèle de la machine asynchrone sera établi en tenant compte des considérations de base suivantes :

- Le circuit magnétique n'est pas saturé et a une perméabilité constante,
- La répartition dans l'entrefer de la force magnétomotrice et celle du flux sont sinusoïdales.

- Les pertes fer par hystérésis et courants de FOUCAULT, l'effet de peau et les effets de dentures sont négligés,
- Les effets des encoches sont négligés,

Dans le but de simplifier la modélisation de la machine asynchrone, l'étude théorique est menée sur une machine bipolaire. Les résultats sont transposables pour une machine multipolaire à condition de multiplier le couple et de diviser la vitesse par p.

### 2.2.3 Modèle diphasée de la machine asynchrone

Au sens du modèle de Park, la structure de la MAS diphasée montrée par la figure 2.1.

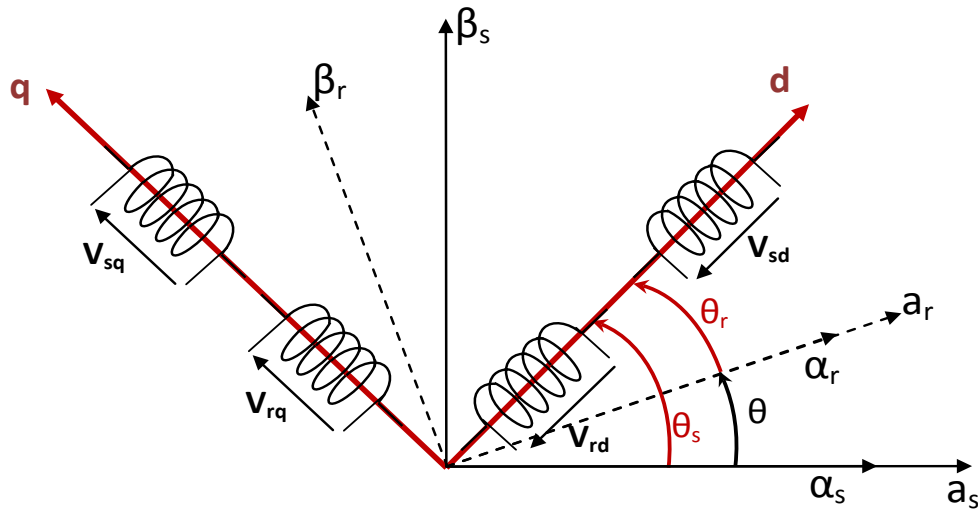


Figure 2.1. Structure diphasée de la machine asynchrone dans le repère d-q

Dans cette figure, on a repéré les différents référentiels communs au stator et au rotor jugé utile dans la mise en place des modèles dynamiques de la machine asynchrone.

#### 2.2.3.1 Modèle de Concordia de la MAS

Afin de développer les lois de commande basées sur le contrôle et le réglage des grandeurs statoriques de la machine, il est plus judicieux de choisir le repère diphasé  $(\alpha, \beta)$  lié au stator. Les variables de commande sont les tensions statoriques alors que les courants statoriques et les flux statoriques représentent les variables d'état. Le modèle d'état pseudo stationnaire de la machine asynchrone dans ce référentiel qui fait apparaître les paramètres principaux est exprimé par [BiBo06] [MfMi08] :

$$\frac{d}{dt} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \varphi_{s\alpha} \\ \varphi_{s\beta} \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) & -\omega_r & \frac{R_r}{\sigma L_r L_s} & \frac{\omega_r}{\sigma L_s} \\ \omega_r & -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) & -\frac{\omega_r}{\sigma L_s} & \frac{R_r}{\sigma L_r L_s} \\ -R_s & 0 & 0 & 0 \\ 0 & -R_s & 0 & 0 \end{bmatrix} \times \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \varphi_{s\alpha} \\ \varphi_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (2.1)$$

avec  $\sigma = 1 - \frac{M^2}{L_s * L_r}$  représente le coefficient de dispersion de Blondel.

Ce modèle constitue la base servant à l'élaboration des lois de commande DTC pour la MAS.

### 2.2.3.2 Modèle de Park de la MAS

Dans le repère de Park (d,q) tournant à la vitesse de synchronisme  $\omega_s$ , le modèle dynamique le plus adopté à l'élaboration des lois de commande vectorielle de la machine asynchrone à flux rotorique orienté est régit par :

$$\frac{d}{dt} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \varphi_{rd} \\ \varphi_{rq} \end{bmatrix} = \begin{bmatrix} -(\frac{1}{T_s \cdot \sigma} + \frac{1-\sigma}{T_r \cdot \sigma}) & \omega_s & \frac{1-\sigma}{T_r M \sigma} & \frac{1-\sigma}{M \sigma} \omega \\ -\omega_s & -(\frac{1}{T_s \cdot \sigma} + \frac{1-\sigma}{T_r \cdot \sigma}) & \frac{1-\sigma}{M \sigma} \omega & \frac{1-\sigma}{T_r M \sigma} \\ \frac{M}{T_r} & 0 & -\frac{1}{T_r} & (\omega_s - \omega) \\ 0 & \frac{M}{T_r} & -(\omega_s - \omega) & -\frac{1}{T_r} \end{bmatrix} \times \begin{bmatrix} i_{sd} \\ i_{sq} \\ \varphi_{rd} \\ \varphi_{rq} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix} \quad (2.2)$$

On retient que la principale caractéristique de ce modèle réside au niveau de ses variables d'état qui sont constants en régime permanent. Il est souvent adopté à la mise en place d'une commande DTC via l'exploitation de la technique de modulation vectorielle.

## 2.3 Alimentation de la MAS par un onduleur de tension

### 2.3.1 Introduction

Le réglage de la vitesse de la MAS se réalise par action simultanée sur la fréquence et la tension (ou le courant) statorique. Par conséquent, pour se donner les moyens de cette action, il faut disposer d'une source d'alimentation capable de délivrer une tension d'amplitude et de fréquence réglable en valeurs instantanées, selon des critères liés aux performances dynamiques souhaitées. Cette source n'est autre qu'un onduleur alimenté par une tension continue constante.

### 2.3.2 Structure de l'onduleur

Pour modéliser l'onduleur de tension présenté par la figure 2.2, on considère son alimentation comme une source de tension parfaite supposée être constituée de deux générateurs de force électromotrice égale à  $E/2$ .

Il est à noter que les interrupteurs d'un même bras ne doivent jamais être fermés simultanément. Chaque interrupteur n'est donc enclenché qu'une seule fois par période, c'est à dire que si :  $S_j=1$ , alors  $S_j'=0$  avec  $j = a, b, c$ .

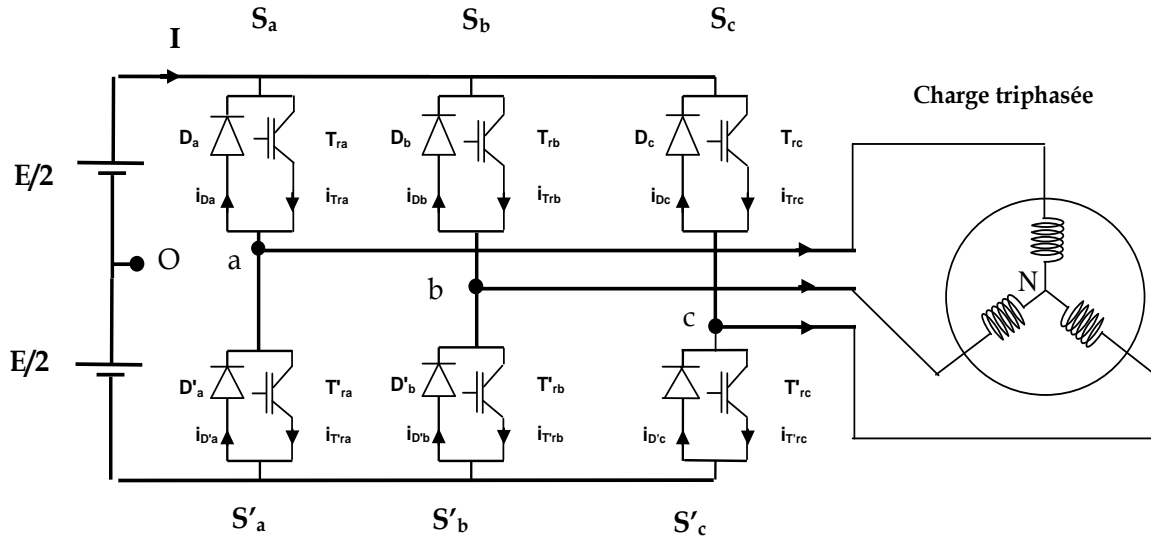


Figure 2.2. Structure d'un onduleur triphasé de tension à deux niveaux

La commutation entre les interrupteurs est supposée instantanée et les chutes de tension à leurs bornes sont considérées négligeables.

### 2.3.3 Détermination des tensions de sortie de l'onduleur :

Pour simplifier l'étude de ce type d'onduleur, les différentes tensions côté continu sont référencées par rapport au milieu O de la source continue à l'entrée de l'onduleur. Les tensions de référence pour les trois phases sont les tensions simples référencées par rapport au point neutre N.

Les tensions composées  $v_{ab}$ ,  $v_{bc}$  et  $v_{ca}$  peuvent être exprimées d'une part en fonction des tensions d'entrée de l'onduleur  $v_{ao}$ ,  $v_{bo}$  et  $v_{co}$  et d'autre part en fonction des tensions de phases  $v_{an}$ ,  $v_{bn}$  et  $v_{cn}$  :

$$\begin{cases} v_{ab} = v_{ao} + v_{ob} = v_{ao} - v_{bo} \\ v_{bc} = v_{bo} + v_{oc} = v_{bo} - v_{co} \\ v_{ca} = v_{co} + v_{oa} = v_{co} - v_{ao} \end{cases} \quad \text{et/ou} \quad \begin{cases} v_{ab} = v_{an} + v_{nb} = v_{an} - v_{bn} \\ v_{bc} = v_{bn} + v_{nc} = v_{bn} - v_{cn} \\ v_{ca} = v_{cn} + v_{na} = v_{cn} - v_{an} \end{cases} \quad (2.3)$$

De plus, la charge connectée à l'onduleur est triphasée équilibrée, on a donc la relation :

$$v_{an} + v_{bn} + v_{cn} = 0 \quad (2.4)$$

Les relations précédentes permettent de déduire le système d'équations suivant reliant les tensions de phase aux bornes de la charge aux tensions d'entrée de l'onduleur :

$$\begin{cases} v_{an} = \frac{1}{3} \times (2.v_{ao} - v_{bo} - v_{co}) \\ v_{bn} = \frac{1}{3} \times (-v_{ao} + 2.v_{bo} - v_{co}) \\ v_{cn} = \frac{1}{3} \times (-v_{ao} - v_{bo} + 2.v_{co}) \end{cases} \quad (2.5)$$



Les tensions simples peuvent s'exprimer en fonction de l'état des interrupteurs. En effet, les tensions  $v_{ao}$ ,  $v_{bo}$  et  $v_{co}$  peuvent s'écrire :

$$\begin{cases} v_{ao} = \frac{E}{2} \times (S_a - S'_a) = \frac{E}{2} \cdot (2.S_a - 1) \\ v_{bo} = \frac{E}{2} \times (S_b - S'_b) = \frac{E}{2} \cdot (2.S_b - 1) \\ v_{co} = \frac{E}{2} \times (S_c - S'_c) = \frac{E}{2} \cdot (2.S_c - 1) \end{cases} \quad (2.6)$$

Ainsi l'onduleur peut être modélisé par une matrice symétrique assurant le passage continu-alternatif :

$$\begin{pmatrix} V_{aN} \\ V_{bN} \\ V_{cN} \end{pmatrix} = \frac{E}{3} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} S_a \\ S_b \\ S_c \end{pmatrix} \quad (2.7)$$

La commande de l'onduleur de tension triphasé est effectuée en utilisant huit possibilités pour le vecteur tension statorique et chaque possibilité correspond à un état de fonctionnement de l'onduleur de tension [TakI89].

### 2.3.4 Commande de l'onduleur triphasé

Il existe de nombreuses façons pour commander l'onduleur triphasé pour obtenir une tension alternative aux bornes de la charge. A titre d'exemple, on cite les techniques de commande les plus considérées :

- Par modulation à hystérésis.
- Par modulation naturelle ou modulation sinus-triangle (intersective) effectuant la comparaison d'un signal de référence sinusoïdale à une porteuse en général triangulaire.
- Par modulation vectorielle "SVM" (Space Vector Modulation) appelée encore MLI régulière symétrique.

Dans cette thèse, on synthétise des lois de commande DTC de la MAS via l'exploitation du principe de la modulation à hystérésis.

## 2.4 Commande directe de couple de la machine asynchrone

### 2.4.1 Introduction

La commande DTC a été introduite en 1986 par Takahashi [TakI89]. La DTC est une technique de commande exploitant la possibilité d'imposer un couple et un flux aux machines asynchrones d'une manière découplée. La commande DTC conventionnelle a été largement utilisée dans les applications industrielles [LinL06] [RaIK06] [ToLi05b]. Pour étudier cette stratégie, on commence par présenter et mettre en œuvre la structure DTC de Takahashi.

### 2.4.2 Structure de commande des moteurs asynchrones par DTC

Le principe de la commande DTC est basé sur une détermination directe des impulsions de commande appliquées aux interrupteurs de l'onduleur de tension. Ceci afin de maintenir le couple électromagnétique et le flux statorique à l'intérieur de deux bandes à hystérésis prédéfinies. Une telle application de cette technique permet d'assurer un découplage entre le contrôle du couple et le flux. L'onduleur de tension permet d'atteindre sept positions dans le plan de phase, correspondant aux huit séquences du vecteur de tension à la sortie de l'onduleur [TakI89].

Le schéma bloc de la figure 2.3, représente le synoptique d'une commande DTC en vitesse appliquée à la machine asynchrone.

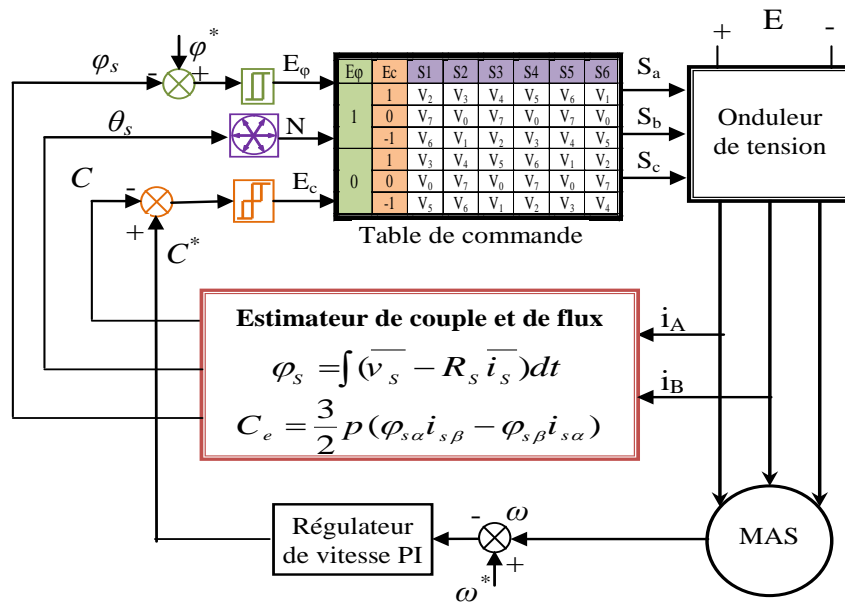


Figure 2.3. Schéma structurel de la commande DTC.

Les principaux éléments constitutifs de la commande DTC sont énumérés ainsi :

- Des estimateurs de flux statorique et de couple électromagnétique,
- La table de commande de Takahashi,
- Les comparateurs à hystérésis,
- Un régulateur de vitesse.

L'objectif cible de la commande DTC est de réguler le flux statorique et le couple électromagnétique sans disposer de mesures, de flux ou de couple. Les seules mesures utilisées sont les courants du stator de la machine. Le couple et le flux sont estimés à partir de ces mesures.

### 2.4.3 Stratégie de commande directe de couple et de flux

#### 2.4.3.1 Contrôle du vecteur de flux statorique

Dans le repère fixe ( $\alpha$ ,  $\beta$ ) lié au stator, le flux statorique est estimé à partir de l'équation suivante [MfMi08] :

$$\bar{v}_s = R_s \bar{i}_s + \frac{d\bar{\varphi}_s}{dx} \quad (2.8)$$

Soit encore :

$$\bar{\varphi}_s = \bar{\varphi}_{s0} + \int_0^t (\bar{v}_s - R_s \bar{i}_s) dt \quad (2.9)$$

Pour les grandes vitesses, la chute de tension due à la résistance du stator peut être négligée, la relation (2.9) se trouve simplifiée ainsi :

$$\bar{\varphi}_s \approx \bar{\varphi}_{s0} + \int_0^t \bar{v}_s dt \quad (2.10)$$

Pendant une période d'échantillonnage, le vecteur de tension appliqué au MAS reste constant, on peut écrire alors :

$$\bar{\varphi}_s(k+1) \approx \bar{\varphi}_s(k) + \bar{v}_s(k)T_e \quad (2.11)$$

Où encore :

$$\Delta\bar{\varphi}_s = \bar{\varphi}_s(k+1) - \bar{\varphi}_s(k) = \bar{v}_s(k)T_e \quad (2.12)$$

avec :

$\bar{\varphi}_s(k)$  : Vecteur de flux statorique au pas d'échantillonnage actuel ;

$\bar{\varphi}_s(k+1)$  : Vecteur du flux statorique au pas d'échantillonnage suivant ;

$\Delta\varphi_s$  : Variation du vecteur flux statorique ;

$T_e$  : Période d'échantillonnage.

L'équation (2.12), montre que la variation vecteur tension statorique est proportionnelle au vecteur flux statorique. Pour augmenter le flux statorique, il suffit d'appliquer un vecteur de tension dans sa direction et qui lui est colinéaire et inversement.

#### 2.4.3.2 Contrôle du couple électromagnétique

En faisant intervenir les flux statorique et rotorique, le couple électromagnétique est exprimé par :

$$C_e = \frac{3}{2} p (\bar{\varphi}_s \times \bar{\varphi}_r) = \frac{3}{2} p \left| \bar{\varphi}_s \right| \left| \bar{\varphi}_r \right| \sin \delta \quad (2.13)$$

Avec :

$\bar{\varphi}_s$  : vecteur de flux statorique ;

$\bar{\varphi}_r$  : vecteur de flux rotorique ramené au stator ;

$\delta$  : angle entre les vecteurs des flux statorique et rotorique.

Le couple dépend des amplitudes des deux vecteurs  $\bar{\varphi}_s$  et  $\bar{\varphi}_r$  et de leur position relative. Si l'on parvient à contrôler correctement le flux  $\bar{\varphi}_s$  en module et en position, on assure un découplage entre le contrôle du couple électromagnétique et le flux statorique.

L'estimation du couple est effectuée à partir des grandeurs statorique  $\varphi_{s\alpha}$ ,  $\varphi_{s\beta}$ ,  $i_{s\alpha}$  et  $i_{s\beta}$  par la relation suivante :

$$C_e = \frac{3}{2} p (\varphi_{s\alpha} \times i_{s\beta} - \varphi_{s\beta} \times i_{s\alpha}) \quad (2.14)$$

### 2.4.3.3 Sélection du vecteur de tension

Le choix du vecteur tension statorique  $\bar{v}_s$  dépend de la variation souhaitée pour le module du flux statorique  $\bar{\varphi}_s$ , du sens de rotation de  $\bar{\varphi}_s$  et également de l'évolution souhaitée pour le couple électromagnétique.

En se plaçant dans le repère  $(\alpha, \beta)$  lié au stator, on peut délimiter l'espace de  $\bar{\varphi}_s$  en le décomposant en six zones appelées secteurs comme illustré par la figure 2.4. L'axe  $(\alpha)$  est choisi confondu avec l'axe de la phase (a) du stator.

Lorsque le flux  $\bar{\varphi}_s$  se trouve dans une zone  $i$  ( $i=1, \dots, 6$ ), le contrôle du flux et du couple peut être assuré en sélectionnant l'un des huit vecteurs tensions suivants :

- Si  $\bar{v}_{i+1}$  est sélectionné alors  $\bar{\varphi}_s$  croît et  $C_e$  croît ;
- Si  $\bar{v}_{i-1}$  est sélectionné alors  $\bar{\varphi}_s$  croît et  $C_e$  décroît ;
- Si  $\bar{v}_{i+2}$  est sélectionné alors  $\bar{\varphi}_s$  décroît et  $C_e$  croît ;
- Si  $\bar{v}_{i-2}$  est sélectionné alors  $\bar{\varphi}_s$  décroît et  $C_e$  décroît ;
- Si  $\bar{v}_0$ , ou  $\bar{v}_7$  est sélectionné, alors la rotation du flux  $\bar{\varphi}_s$  est arrêtée, d'où une décroissance du couple alors que le module de flux  $\bar{\varphi}_s$  reste inchangé.

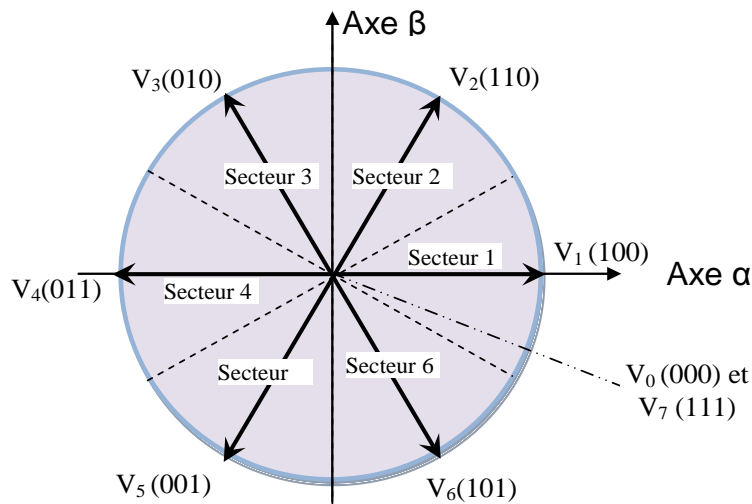


Figure 2.4. Vecteurs actifs des tensions statoriques et secteurs

Le vecteur de tension  $\bar{v}_s$  à la sortie de l'onduleur, est déduit à partir des écarts de flux et de couple estimés par rapport à leurs valeurs de référence et de la position du vecteur  $\bar{\varphi}_s$ . Un estimateur de couple, un estimateur de module de  $\bar{\varphi}_s$  et de sa position sont donc nécessaires.

#### 2.4.3.4 Reconstitution du flux statorique

La commande de tout système électrique exige entre autre, de disposer à tout instant d'une connaissance sur l'évolution de son état. Dans le cas d'une machine asynchrone, il faut disposer de mesures de grandeurs mécaniques (couple de charge, vitesse) et de grandeurs électromagnétiques (flux, courant). Ces mesures peuvent être obtenues à partir des capteurs mécaniques et électriques qui sont généralement sensibles aux perturbations. Il faut signaler aussi que certaines variables d'état ne sont pas accessibles à la mesure. Afin de résoudre ces problèmes, on peut penser à reconstituer l'état non mesurable directement à partir des grandeurs physiques facilement accessibles. La reconstitution de l'état s'effectue, soit par un estimateur en recopiant le modèle du système sans retour d'information, ce principe est illustré par la figure 2.5, soit par un observateur en recopiant la dynamique du système muni d'un retour d'information, comme indiqué par la figure 2.6. Ces méthodes sont basées sur la modélisation de la machine asynchrone [ZhXu07], [ZhZh10], [RaTr12].

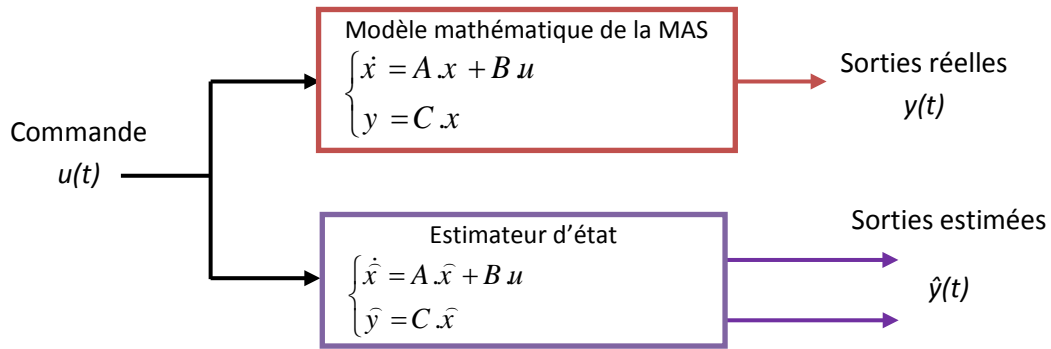


Figure 2.5. Schéma de principe d'un estimateur d'état

##### 2.4.3.4.1 Estimateur du flux et du couple

Les relations de base utilisées pour reconstituer le flux statorique et par conséquent le couple sont exprimées par :

$$\begin{cases} v_{s\alpha} = R_s i_{s\alpha} + \frac{d\varphi_{s\alpha}}{dt} \\ v_{s\beta} = R_s i_{s\beta} + \frac{d\varphi_{s\beta}}{dt} \end{cases} \quad (2.15)$$

Les tensions  $v_{s\alpha}$  et  $v_{s\beta}$  sont liées aux commandes (Sa, Sb, Sc) des interrupteurs de l'onduleur et de la tension continue alimentant ce convertisseur U et s'expriment par la relation :

$$\begin{cases} v_{s\alpha} = \sqrt{\frac{2}{3}}U(sa - \frac{1}{2}(sb + sc)) \\ v_{s\beta} = \sqrt{\frac{2}{3}}U(sb + sc) \end{cases} \quad (2.16)$$

La formulation discrétisée de cet estimateur, définie pour une période d'échantillonnage  $T_e$ , est donnée par :

$$\begin{cases} \varphi_{s\alpha}(k) = \varphi_{s\alpha}(k-1) + (v_{s\alpha}(k-1) - R_s i_{s\alpha}(k-1))T_e \\ \varphi_{s\beta}(k) = \varphi_{s\beta}(k-1) + (v_{s\beta}(k-1) - R_s i_{s\beta}(k-1))T_e \end{cases} \quad (2.17)$$

Le flux estimé est défini par son module et sa position donnés par la relation suivante :

$$\begin{cases} \varphi_s = \sqrt{\varphi_{s\alpha}^2 + \varphi_{s\beta}^2} \\ \arg \varphi_s = \arctg \frac{\varphi_{s\beta}}{\varphi_{s\alpha}} \end{cases} \quad (2.18)$$

Le couple électromagnétique peut être estimé via l'expression :

$$C_e = \frac{3}{2} p (\varphi_{s\alpha} i_{s\beta} - \varphi_{s\beta} i_{s\alpha}) \quad (2.19)$$

Les inconvénients majeurs dans la mise en œuvre pratique de ce type d'estimateur du flux sont marqués d'une part par la précision de l'estimateur qui est tributaire de la variation de la résistance du stator et d'autre part par l'existence d'une intégration en boucle ouverte qui pourra introduire une composante continue et entraîne l'instabilité de la commande. Afin de remédier à ces problèmes, plusieurs méthodes ont été proposées. Dans [MySh98] et [DsFr03], l'intégrateur est remplacé par un filtre passe bas afin d'éviter le problème de la composante continue, cela contribue à l'amélioration de l'estimation.

#### 2.4.3.4.2 Observation du flux statorique

Un observateur n'est qu'un estimateur en boucle fermée qui introduit une matrice de gain pour corriger l'erreur d'estimation. Un schéma de principe d'un observateur d'état est illustré par la figure 2.6. Il existe actuellement dans la littérature plusieurs techniques d'observation du flux statorique pour la machine asynchrone, citons en particulier :

i) Observateur de Kalman : Cet observateur, appelé aussi filtre de Kalman [ReKa60]. Le filtrage de Kalman repose sur le principe de tout observateur, illustré par la figure 2.6, sauf que pour ce type d'observateur le gain est calculé à chaque itération de façon itérative, à partir des prédictions d'erreur et des incertitudes sur les mesures et les grandeurs d'état. Ceci permet de minimiser l'erreur entre le modèle et le système réel [AbKh10] [ChFo09]. Dans ses travaux de thèse, F. Morand [FrMo05] propose un observateur de vitesse de type mode glissant et un filtre de Kalman. L'auteur montre que le suivi de vitesse est relativement bon en régime éloigné de la zone de faible vitesse. Cependant un léger biais de la vitesse observée apparaît lorsque le couple de charge augmente. Pour le moteur pas à pas, l'auteur de [MoBj07] a

utilisé le filtre de Kalman pour l'observation de la vitesse de rotation et la position du rotor. Les résultats obtenus de l'implémentation expérimentale montrent la robustesse de ce type d'observateur par rapport aux différentes perturbations qui peuvent agir sur le système. Par contre, son inconvénient réside au niveau du temps de calcul à cause de la présence de plusieurs opérations matricielles dans son algorithme.

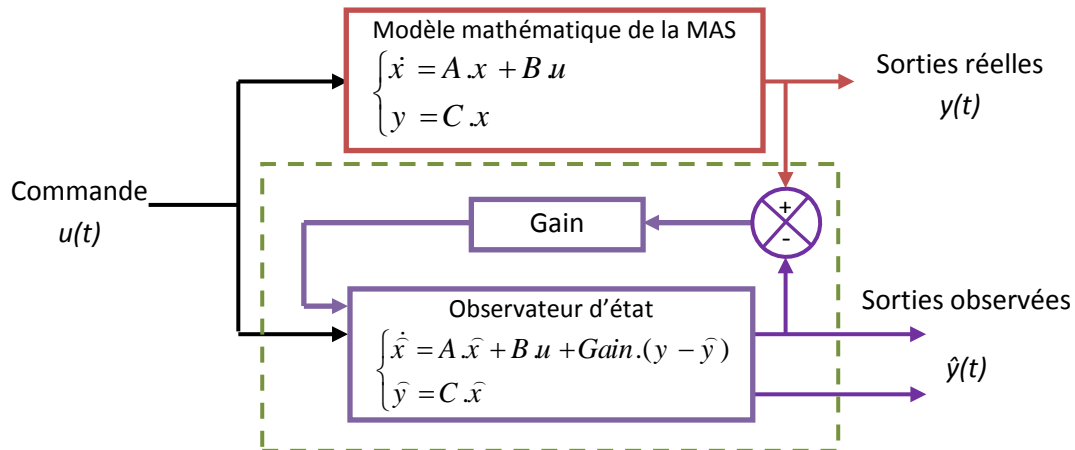


Figure 2.6. Schéma de principe d'un observateur d'état

- ii) **Observateur à grand gain** : c'est un observateur purement dédié aux systèmes non linéaire. L'idée de base de cet observateur consiste à masquer la non linéarité du système en choisissant des gains très grands sur les parties linéaires. Un observateur interconnecté à grand gain et un observateur adaptatif interconnecté sont proposés dans [DrTr08]. Ces observateurs ont pour but de reconstruire les variables mécaniques (vitesse et couple de charge) et les variables magnétiques (flux) de la machine à partir de l'unique mesure des courants statoriques. Enfin une étude expérimentale est présentée afin d'évaluer leurs stabilités et leurs performances en particulier dans la zone inobservable [DrTr08]. Dans ses travaux de thèse, T. Boumegoura [TaBo01] propose dans la première partie une approche de surveillance en ligne basée sur un observateur grand gain étendu et réduit. Cet observateur consiste à suivre la résistance rotorique. Dans la deuxième partie, il a étendu l'observation à l'état magnétique de la machine.
- iii) **Observateur à mode glissant** : cet observateur est basé sur la théorie des systèmes à structures variables. A partir de la mesure des courants et des tensions de phase, on peut estimer les variables d'état du système. Son principe consiste à comparer les sorties mesurées du système et celle estimées, ensuite, définir une surface dite de glissement qui dépend aussi de cet écart. Le signe de cette surface calculée en temps réel est multiplié par une matrice de gain, afin d'assurer une erreur d'estimation nulle. Il possède les mêmes propriétés de robustesse que les contrôleurs par mode glissant. L'auteur de [RiAl02] a développé deux types d'observateur de flux : un observateur basé sur la théorie des modes glissants et un observateur à grand gain. Il a mis en évidence la sensibilité de l'observateur par mode glissant par rapport à la variation de la résistance rotorique. Il a donné ensuite une solution pour améliorer la précision de l'observateur de flux. La solution consiste à intégrer des dispositifs d'adaptation des paramètres résistifs. Par contre la dynamique de l'observateur à grand gain est bonne dans toute la plage de variation de la vitesse sans évoquer la nécessité d'un

dispositif d'adaptation paramétrique. Dans le travail présenté dans [FaMe10], plusieurs approches d'estimation et d'observation de la machine asynchrone ont été élaborées. Ces approches répondent directement aux besoins les plus critiques des lois de commande de la machine asynchrone en termes de robustesse paramétrique et assurent un bon suivi sur toute la plage de vitesse. L'implémentation expérimentale a confirmé l'efficacité de toutes ces approches. Une étude comparative a été menée sur ces méthodes d'observation. Les estimateurs MRAS-mode glissant ont donné les meilleures performances en termes d'observation de la vitesse et d'estimation paramétrique. L'observateur par mode glissant est caractérisé par sa robustesse par rapport aux perturbations et à la variation des paramètres du moteur [CrLa06].

Les études de l'observabilité ainsi que les conditions suffisantes d'observabilité et de perte d'observabilité de la machine asynchrone ont été largement abordées et traitées dans la littérature [CaWe00], [Selb04], [MaGh05], [MaGh06] et [HoHg11]. Les performances de la commande DTC de la machine asynchrone reposent sur la bonne reconstitution du flux statorique. Dans ce cadre, on a mis en œuvre dans nos travaux de recherche deux techniques d'observation : un observateur par mode glissant et un observateur à grand gain. Ces observateurs ont pour but de reconstruire le flux statorique de la machine à partir de l'unique mesure des courants statoriques. Ensuite une étude comparative basée sur les résultats de simulation est discutée, pour évaluer leurs stabilités et leurs performances en particulier lors de la variation de la résistance statorique.

#### **a. Observateur du flux statorique par mode glissant**

C'est un observateur de flux statorique à base du modèle de courant par mode glissant. L'avantage de cet observateur est de ne pas utiliser en entrée la vitesse et la constante de temps rotorique. De plus, l'utilisation de la technique des modes glissants pour la création de cet observateur assure à la fois de bonnes performances dynamiques sur toute la gamme de vitesse et une robustesse vis à vis des différentes perturbations [MoBj07] [FaMe10].

La synthèse de l'observateur par mode glissant est effectuée à partir du modèle de la machine lié au stator et faisant intervenir le flux et les courants statoriques, soit alors :

$$\left\{ \begin{array}{l} \frac{d i_{s\alpha}}{dt} = -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) i_{s\alpha} - \omega_r i_{s\beta} + \frac{R_r}{\sigma L_r L_s} \varphi_{s\alpha} + \frac{\omega_r}{\sigma L_s} \varphi_{s\beta} + \frac{1}{\sigma L_s} v_{s\alpha} \\ \frac{d i_{s\beta}}{dt} = -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) i_{s\beta} + \omega_r i_{s\alpha} + \frac{R_r}{\sigma L_r L_s} \varphi_{s\beta} - \frac{\omega_r}{\sigma L_s} \varphi_{s\alpha} + \frac{1}{\sigma L_s} v_{s\beta} \\ \frac{d \varphi_{s\alpha}}{dt} = v_{s\alpha} - R_s i_{s\alpha} \\ \frac{d \varphi_{s\beta}}{dt} = v_{s\beta} - R_s i_{s\beta} \end{array} \right. \quad (2.20)$$

Les composantes du flux statorique ne sont pas mesurables, la vitesse et les courants statoriques sont obtenus par une mesure directe. L'observateur du courant et de flux peut être décrit par le système d'équations (2.21) [MaGh05] [HaHa12].



$$\left\{ \begin{aligned} \frac{d \hat{i}_{s\alpha}}{dt} &= -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \hat{i}_{s\alpha} - \omega_r \hat{i}_{s\beta} + \frac{R_r}{\sigma L_r L_s} \hat{\phi}_{s\alpha} + \frac{\omega_r}{\sigma L_s} \hat{\phi}_{s\beta} + \frac{1}{\sigma L_s} v_{s\alpha} + A_{i1} I_{s1} + A_{i2} I_{s2} \\ \frac{d \hat{i}_{s\beta}}{dt} &= -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \hat{i}_{s\beta} + \omega_r \hat{i}_{s\alpha} + \frac{R_r}{\sigma L_r L_s} \hat{\phi}_{s\beta} - \frac{\omega_r}{\sigma L_s} \hat{\phi}_{s\alpha} + \frac{1}{\sigma L_s} v_{s\beta} + A_{i3} I_{s1} + A_{i4} I_{s2} \\ \frac{d \hat{\phi}_{s\alpha}}{dt} &= v_{s\alpha} - R_s \hat{i}_{s\alpha} + A_{\phi1} I_{s1} + A_{\phi2} I_{s2} \\ \frac{d \hat{\phi}_{s\beta}}{dt} &= v_{s\beta} - R_s \hat{i}_{s\beta} + A_{\phi3} I_{s1} + A_{\phi4} I_{s2} \end{aligned} \right. \quad (2.21)$$

avec :

$\hat{i}_{s\alpha}, \hat{i}_{s\beta}, \hat{\phi}_{s\alpha}, \hat{\phi}_{s\beta}$  les composantes du courant statorique et du flux rotorique estimées respectives de  $i_{s\alpha}, i_{s\beta}, \phi_{s\alpha}, \phi_{s\beta}$  dans le repère  $(\alpha, \beta)$ .

$$I_s = \begin{pmatrix} I_{s1} \\ I_{s2} \end{pmatrix} = \begin{pmatrix} \text{signe}(S1) \\ \text{signe}(S2) \end{pmatrix} \quad (2.22)$$

$A_{ij} (j=1,2,3,4)$  et  $A_{\phi j} (j=1,2,3,4)$  : représentent les gains de l'observateur.

$S1$  et  $S2$  sont les surfaces de glissement

$I_s$  : le vecteur signe de la surface de glissement choisi.

Désignons par :

$\varepsilon_i$  : l'erreur d'observation du courant,

$\varepsilon_\phi$  : l'erreur d'observation du flux statorique.

$$\begin{aligned} \varepsilon_i &= \begin{pmatrix} \varepsilon_{is\alpha} \\ \varepsilon_{is\beta} \end{pmatrix} = \begin{pmatrix} i_{s\alpha} - \hat{i}_{s\alpha} \\ i_{s\beta} - \hat{i}_{s\beta} \end{pmatrix} \\ \varepsilon_\phi &= \begin{pmatrix} \varepsilon_{\phi s\alpha} \\ \varepsilon_{\phi s\beta} \end{pmatrix} = \begin{pmatrix} \phi_{s\alpha} - \hat{\phi}_{s\alpha} \\ \phi_{s\beta} - \hat{\phi}_{s\beta} \end{pmatrix} \end{aligned} \quad (2.23)$$

Sur la base des équations (2.20), (2.21) et (2.23), l'équation régissant les erreurs d'observation est exprimée par :

$$\left\{ \begin{aligned} \frac{d \varepsilon_{i\alpha}}{dt} &= -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \varepsilon_{i\alpha} - \omega_r \varepsilon_{i\beta} + \frac{R_r}{\sigma L_r L_s} \varepsilon_{\phi\alpha} + \frac{\omega_r}{\sigma L_s} \varepsilon_{\phi\beta} - (A_{i1} * \text{signe}(S1) + A_{i2} * \text{signe}(S2)) \\ \frac{d \varepsilon_{i\beta}}{dt} &= -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \varepsilon_{i\beta} + \omega_r \varepsilon_{i\alpha} + \frac{R_r}{\sigma L_r L_s} \varepsilon_{\phi\beta} - \frac{\omega_r}{\sigma L_s} \varepsilon_{\phi\alpha} - (A_{i3} * \text{signe}(S1) + A_{i4} * \text{signe}(S2)) \\ \frac{d \varepsilon_{\phi\alpha}}{dt} &= -R_s \varepsilon_{i\alpha} - (A_{\phi1} * \text{signe}(S1) + A_{\phi2} * \text{signe}(S2)) \\ \frac{d \varepsilon_{\phi\beta}}{dt} &= -R_s \varepsilon_{i\beta} - (A_{\phi3} * \text{signe}(S1) + A_{\phi4} * \text{signe}(S2)) \end{aligned} \right. \quad (2.24)$$

En ce qui concerne la détermination de la matrice gain des courants statoriques, la surface de glissement choisie est liée aux erreurs des courants statoriques [FrMo05], soit :

$$s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = D^{-1} \begin{pmatrix} i_{s\alpha} - \hat{i}_{s\alpha} \\ i_{s\beta} - \hat{i}_{s\beta} \end{pmatrix} \quad (2.25)$$

avec :

$$D = \begin{pmatrix} \frac{R_r}{\sigma L_s L_r} & \frac{\omega_r}{\sigma L_s} \\ -\frac{\omega_r}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} \end{pmatrix} \quad (2.26)$$

Lorsque les erreurs sur les courants sont nulles ( $i_{s\alpha} = \hat{i}_{s\alpha}$  et  $i_{s\beta} = \hat{i}_{s\beta}$ ) alors  $S=0$ .

La matrice  $D$  dépend des paramètres électriques et mécaniques de la machine asynchrone. Ce qui permet de déterminer la dynamique de l'observateur [HaHa12] [RaTa11]. La mesure du courant statorique et de la vitesse avec une bonne précision, permet de supposer que l'observation de ces grandeurs donne pratiquement une erreur d'observation nulle. Ainsi, la matrice gain des courants statoriques est écrite sous la forme suivante :

$$A_i = \begin{pmatrix} A_{i1} & A_{i2} \\ A_{i3} & A_{i4} \end{pmatrix} = D \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{pmatrix} \quad (2.27)$$

Où  $\delta_1$  et  $\delta_2$  sont deux constantes déterminées suite à une analyse de stabilité de façon à assurer l'attractivité de la surface de glissement. L'analyse de stabilité est définie selon l'approche de *Lyapunov*.

Par ailleurs, la détermination de la matrice gain du flux statorique est effectuée de manière à garantir l'attractivité de la surface de glissement ( $S=0$ ). A partir du système d'équation (2.24), on constate que l'annulation de  $S$  et de  $\dot{S}$  se traduit par l'annulation de l'erreur du courant statorique et de sa dérivée ( $\varepsilon_{i\alpha} = \varepsilon_{i\beta} = \dot{\varepsilon}_{i\alpha} = \dot{\varepsilon}_{i\beta} = 0$ ). Il s'en suit :

$$\begin{cases} \frac{R_r}{\sigma L_r L_s} \varepsilon_{\varphi\alpha} + \frac{\omega_r}{\sigma L_s} \varepsilon_{\varphi\beta} = (A_{i1} * \text{signe}(S_1) + A_{i2} * \text{signe}(S_2)) \\ \frac{R_r}{\sigma L_r L_s} \varepsilon_{\varphi\beta} - \frac{\omega_r}{\sigma L_s} \varepsilon_{\varphi\alpha} = (A_{i3} * \text{signe}(S_1) + A_{i4} * (\text{signe}(S_2))) \end{cases} \quad (2.28)$$

En utilisant la relation (2.28), l'équation de la valeur moyenne du vecteur signe est donnée par:

$$I_s = \begin{pmatrix} \text{signe}(S_1) \\ \text{signe}(S_2) \end{pmatrix} = \begin{pmatrix} A_{i1} & A_{i2} \\ A_{i3} & A_{i4} \end{pmatrix}^{-1} D \begin{pmatrix} \varepsilon_{\varphi\alpha} \\ \varepsilon_{\varphi\beta} \end{pmatrix} \quad (2.29)$$

En remplaçant la relation (2.29) dans les équations relatives aux erreurs du flux (2.24), on obtient :

$$\frac{d \varepsilon_\varphi}{dt} = \left( - \begin{pmatrix} A_{\varphi 1} & A_{\varphi 2} \\ A_{\varphi 3} & A_{\varphi 4} \end{pmatrix} \begin{pmatrix} A_{i 1} & A_{i 2} \\ A_{i 3} & A_{i 4} \end{pmatrix}^{-1} \begin{pmatrix} \frac{R_r}{\sigma L_s L_r} & \frac{\omega_r}{\sigma L_s} \\ -\frac{\omega_r}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} \end{pmatrix} \right) \varepsilon_\varphi \quad (2.30)$$

Comme la dynamique de l'observateur est basée sur la convergence exponentielle de l'erreur du flux, on peut alors imposer les variations de l'erreur sur le flux de la manière suivante [HaHa12] [RaTa11] :

$$\frac{d \varepsilon_\varphi}{dt} = -Q \varepsilon_\varphi \quad (2.31)$$

avec :

$$Q = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 \end{pmatrix} \quad (2.32)$$

Où  $q_1$  et  $q_2$  sont des constantes positives.

La matrice des gains du flux obtenue par substitution de l'équation (2.31) dans (2.30) est exprimée par le système d'équations suivant :

$$A_\varphi = \begin{pmatrix} A_{\varphi 1} & A_{\varphi 2} \\ A_{\varphi 3} & A_{\varphi 4} \end{pmatrix} = \begin{pmatrix} q_1 \delta_1 & 0 \\ 0 & q_2 \delta_2 \end{pmatrix} \quad (2.33)$$

La dynamique de l'observateur doit être plus rapide que celle du système ; cela exige un choix adéquat des constantes  $q_1$ ,  $q_2$ ,  $\delta_1$  et  $\delta_2$ .

Bien entendu, il faut vérifier la condition de stabilité de l'observateur. En général, la stabilité du système d'observation par mode glissant dépend de sa convergence vers sa surface de glissement, via le bon choix des paramètres  $\delta_1$  et  $\delta_2$ . Pour ce faire, on considère la fonction de *Lyapunov* définie par l'équation suivante [RiAl02] [HaRe02] [RaTa11] :

$$V = \frac{1}{2} S^T S \quad (2.34)$$

Afin de garantir l'annulation de la surface de glissement, l'erreur d'observation sur les courants doit être nulle. Ainsi, la dérivée de  $V$  est strictement négative [RiAl02] :

$$\dot{V} = S^T \dot{S} < 0 \quad (2.35)$$

Soit alors :

$$\begin{aligned}
\dot{V} &= [S_1 \ S_2]^T D^{-1} \frac{d\epsilon_i}{dt} \\
&= [S_1 \ S_2]^T D^{-1} \left( \begin{pmatrix} \frac{R_r}{\sigma L_s L_r} & -\frac{\omega_r}{\sigma L_s} \\ \frac{\omega_r}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} \end{pmatrix} \begin{bmatrix} \epsilon_{\varphi\alpha} \\ \epsilon_{\varphi\beta} \end{bmatrix} - \begin{pmatrix} A_{i1} & A_{i2} \\ A_{i3} & A_{i4} \end{pmatrix} \begin{pmatrix} \text{signe}(S_1) \\ \text{signe}(S_2) \end{pmatrix} \right) \\
&= [S_1 \ S_2]^T \left( \begin{bmatrix} \epsilon_{\varphi\alpha} \\ \epsilon_{\varphi\beta} \end{bmatrix} - \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{pmatrix} \begin{pmatrix} \text{signe}(S_1) \\ \text{signe}(S_2) \end{pmatrix} \right)
\end{aligned} \tag{2.36}$$

La condition de stabilité est donc satisfaite si l'inégalité (2-37) est vérifiée [RATr12] [RiAl02] :

$$\begin{cases} \delta_1 \geq |\epsilon_{\varphi\alpha}| \\ \delta_2 \geq |\epsilon_{\varphi\beta}| \end{cases} \tag{2.37}$$

Généralement la commande discrète par mode glissant peut prendre la forme de type relais (fonction *signe*) pour la commutation de la surface  $S$  de glissement. Toutefois, l'inconvénient de la commande de type relais entraîne l'apparition d'un phénomène connu en terme anglo-saxonne sous le nom de "*chattering*". En régime permanent, ce phénomène apparaît comme une oscillation de haute fréquence due à la discontinuité brutale au voisinage de zéro. La figure 2.7 représente la fonction de la commande discrète de type relais.

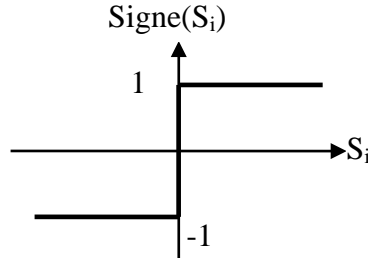


Figure 2.7. Fonction signe (commande de type relais)

Les oscillations dues au phénomène de "*chattering*", peuvent nuire le fonctionnement du circuit de puissance [RaTr12]. Ce phénomène est le principal inconvénient pour les applications de commande par mode de glissement, c'est pour cela des efforts de recherche ont été dirigés afin d'éliminer ou réduire ses effets. L'une des solutions envisagées consiste à remplacer la fonction *signe* par une fonction de *saturation* à fin que la discontinuité au voisinage de zéro soit moins brutale. Cette fonction de saturation illustrée par la figure 2.8, peut être exprimée par l'équation (2.38):

$$\text{Sat}(S_i) = \begin{cases} 1 & \text{si } S_i > +\lambda \\ -1 & \text{si } S_i < -\lambda \\ \frac{S_i}{\lambda} & \text{si } |S_i| < \lambda \end{cases} \tag{2.38}$$

Où  $i=1, 2$  et  $\lambda$  est une constante positive.

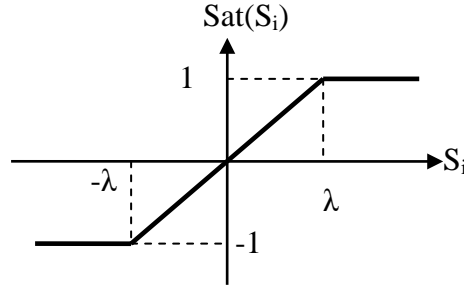


Figure 2.8. Fonction saturation (commande adoucie)

### b. Observateur à grand gain du flux statorique

Le deuxième type d'observateur déterministe non linéaire qui a été mis en place dans cette thèse est celui du grand gain. Ce dernier est synthétisé à partir du modèle de la machine asynchrone dans le référentiel lié au stator.

$$\begin{bmatrix} \dot{i}_{s\alpha} \\ \dot{i}_{s\beta} \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) & -\omega \\ \omega & -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{R_r}{\sigma L_s L_r} & \frac{\omega}{\sigma L_s} \\ -\frac{\omega}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} \end{bmatrix} \begin{bmatrix} \varphi_{s\alpha} \\ \varphi_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (2.39)$$

Désignons par :

$$kF(\omega) = \begin{bmatrix} \frac{R_r}{\sigma L_s L_r} & \frac{\omega}{\sigma L_s} \\ -\frac{\omega}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} \end{bmatrix}$$

avec :

$$k = \frac{1}{\sigma L_s}$$

$$\text{et } F(\omega) = \begin{bmatrix} \frac{1}{T_r} & \omega \\ -\omega & \frac{1}{T_r} \end{bmatrix} = \frac{1}{T_r} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \omega \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \frac{1}{T_r} I_2 - \omega J_2 \quad (2.40)$$

D'où le modèle de la machine est arrangé sous la forme canonique suivante :

$$\begin{cases} \dot{i}_s = A_i i_s + kF(\omega)\varphi_s + BV_s \\ \dot{\varphi}_s = V_s - R_s i_s \\ \dot{\omega} = \frac{3}{2J} p i_s^T J_2 \varphi_s - \frac{1}{J} p k_l \omega - \frac{1}{J} p f \omega \end{cases} \quad (2.41)$$

avec :

$$A_i = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) & -\omega \\ \omega & -\left(\frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r}\right) \end{bmatrix} \text{ et } B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \end{bmatrix}$$

Le modèle de l'observateur à grand gain mis sous la forme suivante [SaHs10] [HoFg11] [AbEl12] :

$$\begin{cases} \dot{\hat{x}} = A \hat{x} + Bu + v(e_m) \\ v(e_m) = \theta \Lambda^{-1}(\hat{x}) \Delta_g^{-1} S^{-1} C^T C e_m \end{cases} \quad (2.42)$$

Avec :

$$x = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \varphi_{s\alpha} \\ \varphi_{s\beta} \end{bmatrix}$$

$$e_m = \begin{bmatrix} i_{s\alpha} - \hat{i}_{s\alpha} \\ i_{s\beta} - \hat{i}_{s\beta} \\ 0 \\ 0 \end{bmatrix}$$

$$\Delta_{g_1} = \text{diag}(I_2, \frac{1}{\theta} I_2) \text{ et } \theta > 0 \text{ d'ou } (\Delta_g)^{-1} = \text{diag}(I_2, \theta I_2)$$

$$\Lambda = \text{diag}(I_2, kF(\omega)) \text{ et } \theta > 0 \text{ d'ou } (\Lambda)^{-1} = \text{diag}(I_2, \frac{1}{k} F^{-1}(\omega))$$

$S$  est l'unique solution symétrique définie positive de l'équation algébrique de Lyapunov suivante :

$$S + A^T S + SA = C^T C \text{ et } S^{-1} C^T = [C_1^T I_2 \quad C_2^T I_2]^T = [2I_2 \quad I_2]^T$$

$$v(e_m) = \theta_1 \begin{bmatrix} I_2 & 0 \\ 0 & \frac{F^{-1}(\omega)}{k} \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & \theta_1 I_2 \end{bmatrix} \begin{bmatrix} 2I_2 \\ I_2 \end{bmatrix} [I_2 \quad 0] e_m$$

Ce qui donne :

$$v(e_m) = \theta_1 \begin{bmatrix} I_2 & 0 \\ 0 & \frac{\theta_1 F^{-1}(\omega)}{k} \end{bmatrix} \begin{bmatrix} 2I_2 \\ I_2 \end{bmatrix} \begin{bmatrix} I_2 & 0 \end{bmatrix} e_m$$

$$v(e_m) = \begin{bmatrix} 2\theta_1 I_2 & 0 \\ \frac{\theta_1^2 F^{-1}(\omega)}{k} & 0 \end{bmatrix} e_m$$

Avec

$$\frac{1}{k} F^{-1}(\omega) = \frac{1}{k} \frac{1}{\frac{1}{T_r^2} + \omega^2} \begin{bmatrix} \frac{1}{T_r} & -\omega \\ \omega & \frac{1}{T_r} \end{bmatrix} = \frac{1}{a_2^2 + a_3^2} \begin{bmatrix} a_2 & -a_3 \\ a_3 & a_2 \end{bmatrix}$$

En remplaçant  $v(e_m)$  par son expression dans (2-39) on obtient :

$$\begin{cases} \frac{d \hat{i}_{s\alpha}}{dt} = a_1 i_{s\alpha} + a_6 i_{s\beta} + a_2 \varphi_{s\alpha} + a_3 \varphi_{s\beta} + b_1 v_{s\alpha} + 2\theta (i_{s\alpha} - \hat{i}_{s\alpha}) \\ \frac{d \hat{i}_{s\beta}}{dt} = -a_6 i_{s\alpha} + a_1 i_{s\beta} - a_3 \varphi_{s\alpha} + a_2 \varphi_{s\beta} + b_1 v_{s\beta} + 2\theta (i_{s\beta} - \hat{i}_{s\beta}) \\ \frac{d \hat{\varphi}_{s\alpha}}{dt} = a_4 i_{s\alpha} + v_{s\alpha} + \theta^2 \frac{a_2}{a_2^2 + a_3^2} (i_{s\alpha} - \hat{i}_{s\alpha}) - \theta^2 \frac{a_3}{a_2^2 + a_3^2} (i_{s\beta} - \hat{i}_{s\beta}) \\ \frac{d \hat{\varphi}_{s\beta}}{dt} = a_4 i_{s\beta} + v_{s\beta} + \theta^2 \frac{a_3}{a_2^2 + a_3^2} (i_{s\alpha} - \hat{i}_{s\alpha}) + \theta^2 \frac{a_2}{a_2^2 + a_3^2} (i_{s\beta} - \hat{i}_{s\beta}) \end{cases} \quad (2.43)$$

Avec

$$a_1 = -\left( \frac{R_s}{\sigma L_s} + \frac{R_r}{\sigma L_r} \right), \quad a_2 = \frac{R_r}{\sigma L_r L_s}, \quad a_3 = \frac{\omega_r}{\sigma L_s}, \quad a_4 = -R_s, \quad a_6 = -\omega_r, \quad \text{et } b_1 = \frac{1}{\sigma L_s}$$

$\theta > 1$  constitue le paramètre de réglage de la dynamique de l'observateur appelé gain choisi arbitrairement. Il faut noter qu'une valeur suffisamment grande de  $\theta$  permet de stabiliser la partie linéaire et d'affaiblir l'effet de la partie non-linéaire. Ainsi, le temps de convergence se trouve réduit, mais l'observation devient sensible aux bruits de mesure. Par contre, une valeur petite de  $\theta$  conduit à l'effet inverse. Finalement, le choix de  $\theta$  est effectué en garantissant un compromis entre la stabilité et la rapidité de la convergence de l'observateur [TaBo01] [TrDr08].

- Mécanisme d'adaptation de la résistance statorique

Par ailleurs, la bonne reconstitution du flux statorique est tributaire de la connaissance de la résistance statorique avec précision. Dans ce cadre, on a pensé à associer aux deux observateurs déjà mis-en en œuvre un mécanisme d'adaptation de cette résistance. Il faut souligner que cette dernière dépend de la température et de l'effet de peau. Ceci peut conduire

à des erreurs dans les composantes du flux statorique de la machine asynchrone. L'estimation de la résistance statorique est basée sur le critère de *Popov* donné par la relation (2.44) :

$$\hat{R}_s = \frac{\eta}{\sigma L_s} \int (S_1 i_{s\alpha} + S_2 i_{s\beta}) dt \quad (2.44)$$

Le schéma structurel de la commande DTC incluant un observateur de flux statorique, sans et avec mécanisme de la résistance statorique, est illustré par la figure 2.9.

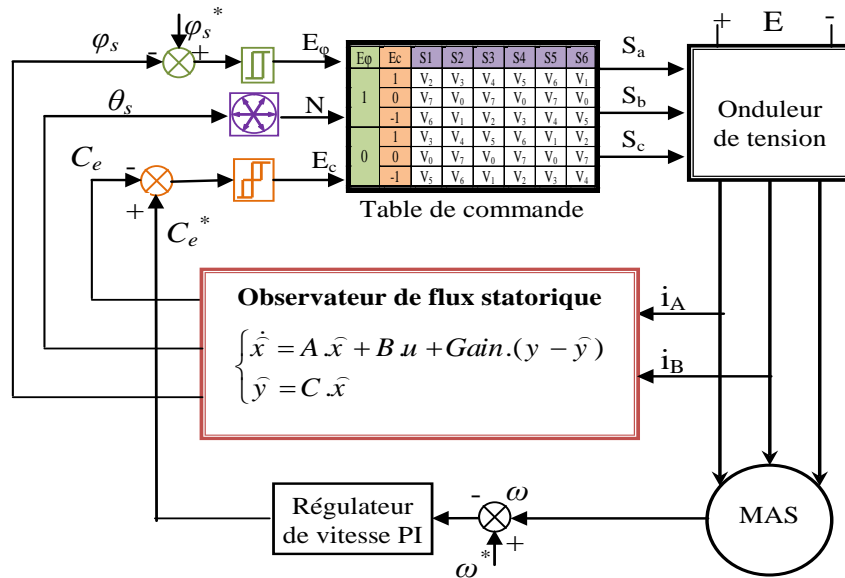


Figure 2.9. Schéma structurel de la commande DTC à base d'un observateur de flux statorique.

#### 2.4.3.5 Comparateur à hystérésis de flux à deux niveaux

Ce correcteur est simple dans son utilisation. Son but est de maintenir l'extrémité du vecteur de flux statorique  $\bar{\varphi}_s$  dans une couronne circulaire comme le montre la figure 2.10 [CCar00] [TakI89].

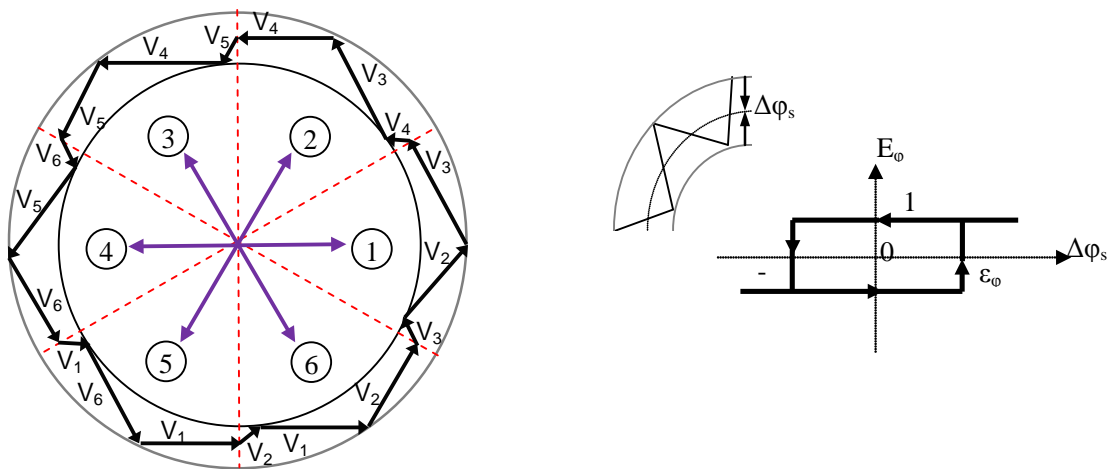


Figure 2.10: a) Choix des tensions  $V_i$  correspondant au contrôle de l'amplitude du  $\varphi_s$ . b) Comparateur du flux à hystérésis.



En effet, si on introduit l'écart  $\Delta\varphi_s$  entre le flux de référence  $\varphi_s^*$  et le flux estimé  $\varphi_s$  dans un comparateur à hystérésis à deux niveaux, il génère à sa sortie la valeur  $E_\varphi = +1$  pour augmenter le flux et  $E_\varphi = 0$  pour le réduire.

#### 2.4.3.6 Comparateur à hystérésis de couple à trois niveaux

Un comparateur à hystérésis à trois niveaux (-1, 0, 1), permet de contrôler le couple électromagnétique développé par le moteur dans les deux sens de rotation, en générant soit un couple positif ou un couple négatif. La figure 2.11 présente un comparateur à hystérésis de couple à trois niveaux.

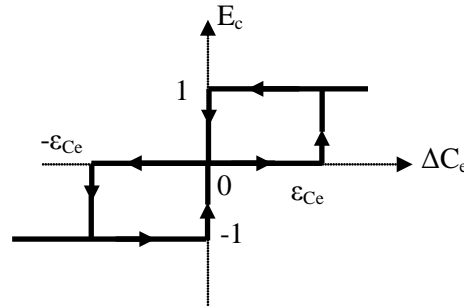


Figure 2.11. Comparateur à hystérésis de couple à trois niveaux.

En introduisant l'écart  $\Delta C_e$  entre le couple électromagnétique de référence  $C_e^*$  et celui estimé dans un comparateur à hystérésis à trois niveaux illustré par la figure 2.11, celui-ci génère la valeur  $E_c = 1$  lorsque le couple est inférieur à la limite inférieure de la bande. Ce qui nécessite son augmentation.  $E_c = -1$  si le couple est supérieur à la limite supérieure de la bande, il faut alors le réduire.  $E_c = 0$  si le couple est à l'intérieur de la bande et il faut le maintenir constant autour de sa référence.

Le choix de la valeur de bande  $\varepsilon_{C_e}$  du comparateur à hystérésis a un grand impact sur la fréquence de commutation. D'ailleurs, si la bande est étroite, l'erreur de couple peut excéder les limites de la bande d'hystérésis. Ceci engendre l'application d'une tension de référence qui permet de diminuer rapidement le couple. En conséquence, cette action se traduit par l'existence d'une importante ondulation de couple [AbHf08] [ToLi05b].

#### 2.4.4 Elaboration de la table de commutation

En fonction du secteur et de l'évolution du flux et du couple on peut choisir le vecteur tension  $V_s$  à appliquer de manière à respecter la consigne de flux et de couple. La table de vérité permettant de choisir le vecteur adéquat est représentée dans tableau 2.1. Ce dernier est élaboré en se basant sur les erreurs de flux  $\Delta\varphi_s$  et de couple  $\Delta C_e$  et selon la position du vecteur de flux statorique ( $i=1,...,6$ ).

Le tableau conventionnel 2.1 des séquences ci-dessous, proposé par Takahashi pour contrôler le flux statorique et le couple électromagnétique de la machine asynchrone est le suivant [TakI89].

Tableau 2.1. Tableau de la DTC conventionnelle définissant les séquences d'un onduleur.

Secteur		S1	S2	S3	S4	S5	S6
$\Delta\phi_s$	$\Delta C_e$						
1	1	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>	V <sub>6</sub>	V <sub>1</sub>
	0	V <sub>7</sub>	V <sub>0</sub>	V <sub>7</sub>	V <sub>0</sub>	V <sub>7</sub>	V <sub>0</sub>
	-1	V <sub>6</sub>	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>
0	1	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>	V <sub>6</sub>	V <sub>1</sub>	V <sub>2</sub>
	0	V <sub>0</sub>	V <sub>7</sub>	V <sub>0</sub>	V <sub>7</sub>	V <sub>0</sub>	V <sub>7</sub>
	-1	V <sub>5</sub>	V <sub>6</sub>	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>

Les séquences nulles V<sub>0</sub> et V<sub>7</sub>, sont indépendantes de l'état du flux, elles sont liées uniquement à l'écart  $\Delta C_e$ . Ces séquences sont appliquées lorsqu'on souhaite maintenir le couple dans sa bande :  $-\varepsilon_{Ce} \leq \Delta C_e \leq \varepsilon_{Ce}$ .

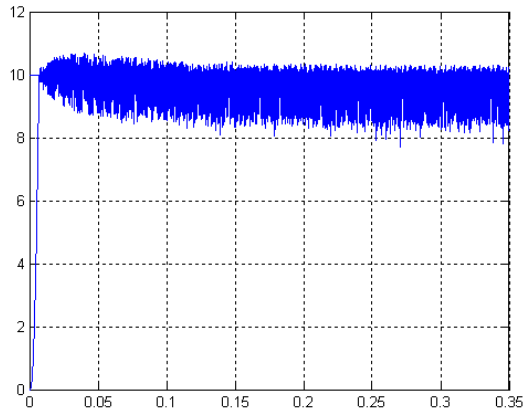
## 2.5 Résultats de simulation

On présente dans ce paragraphe les résultats de simulation relatifs à la commande DTC conventionnelle de la MAS alimentée par un onduleur de tension triphasé. Un programme de simulation, établi sous l'environnement Matlab, nous a permis de reproduire fidèlement le comportement des divers grandeurs électriques et mécaniques. Les simulations sont effectuées en temps discret pour une période d'échantillonnage de 50  $\mu$ s. Le tableau 2.2 donne les grandeurs nominales et les paramètres physique d'une machine asynchrone de Laboratoire.

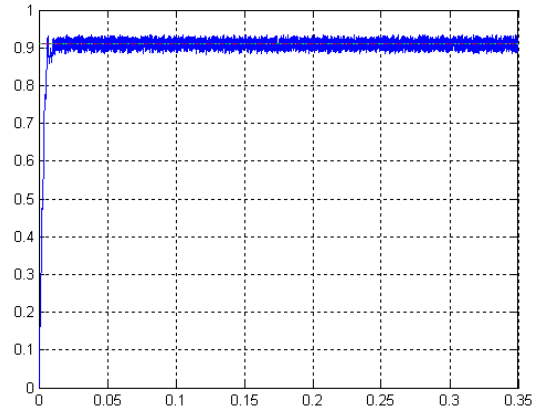
Tableau 2.2. Paramètre de la machine asynchrone

Puissance nominale	1.5 Kw
Couple nominal	10 Nm
Vitesse nominale	1435 tr/min
Nombre de pair de pôles	2
Résistance du stator : $R_s$	5.717 $\Omega$
Résistance du rotor : $R_r$	4.282 $\Omega$
Inductance du stator : $L_s$	0.464 H
Inductance du rotor : $L_r$	0.464 H
Mutuelle inductance : $M$	0.441 H
Moment d'inertie : $J$	0.0049 kg.m <sup>2</sup>

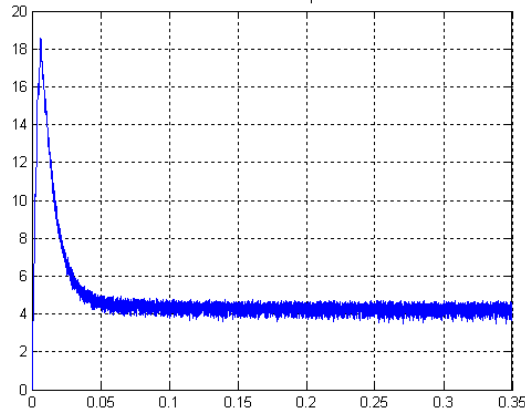
Le premier test concerne le fonctionnement de la MAS en régime nominal. Pour des consignes du flux et du couple égaux respectivement à 0.91 Wb et 10 Nm, la figure 2.12 illustre l'évolution temporelle de couple, de flux statorique, des courants statoriques, de la vitesse et de la trajectoire de l'extrémité du vecteur flux statorique. La fréquence de découpage est fixée à 20Khz. Le couple de charge considéré est proportionnel à la vitesse (la constante de proportionnalité vaut 0.0668).



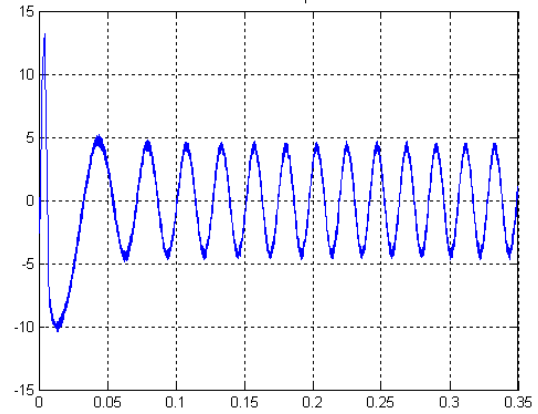
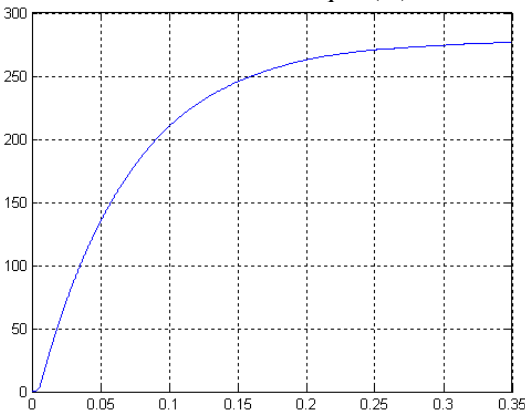
a. Couple (N.m)



b. Flux statorique(Wb)



c. Courant statorique (A)

d. Courant  $i_{s_q}$ (A)

e. Vitesse (rad/sec)

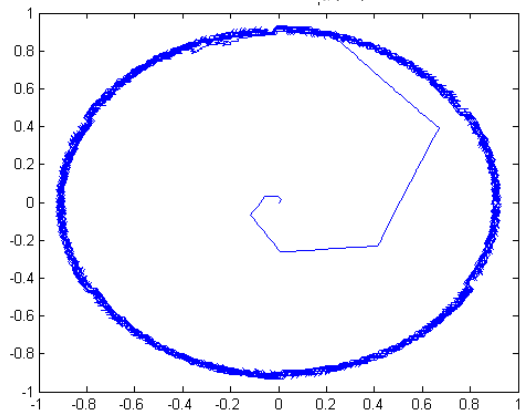
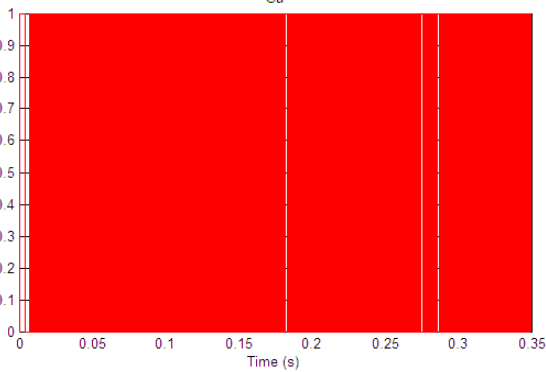
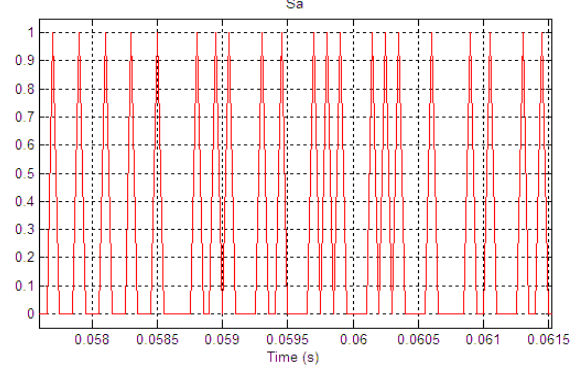
f. flux statorique dans le plan  $(\alpha, \beta)$ g. Commutation de l'interrupteur  $S_a$ h. Commutation de l'interrupteur  $S_a$  (zoom)

Figure 2.12 Résultats de simulation de la commande DTC de Takahashi

On constate d'après la figure 2.12.a que le couple électromagnétique présente un régime transitoire rapide et des oscillations importantes au niveau de son amplitude autour de sa valeur de référence. Ceci constitue un inconvénient majeur de la commande DTC.

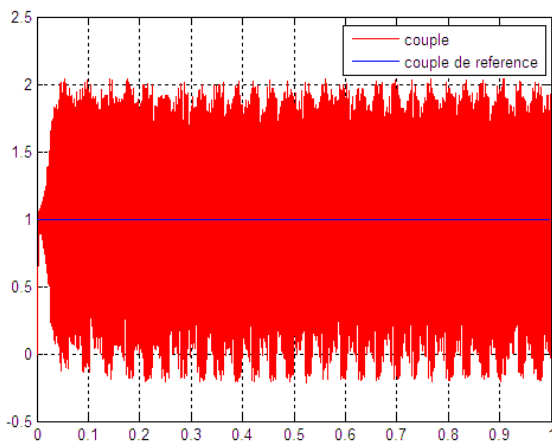
Le flux statorique illustré par la figure 2.12.b, atteint immédiatement sa valeur de référence de 0.91 Wb avec un léger dépassement oscillatoire de 0.07 Wb d'amplitude autour de sa valeur de référence. La trajectoire de l'extrémité du flux statorique (figure 2.12.f) prend une forme presque circulaire d'un rayon égale à 0.91 Wb.

Le courant statorique présenté par la figure 2.12.c est caractérisé par une forte amplitude au démarrage allant jusqu'à 18A, puis il se stabilise à sa valeur nominale.

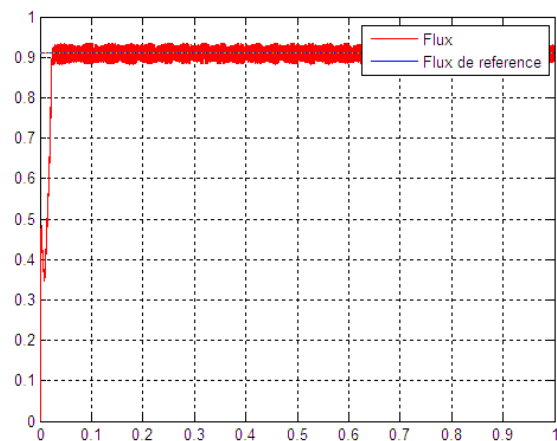
La figure 2.12.e illustre l'évolution temporelle de la vitesse de rotation. La montée en vitesse est quasi linéaire au début du démarrage, la vitesse atteinte est proche de 300 rd/s.

La figure 2.12.g présente l'état de commutation de l'interrupteur "Sa", on remarque que la fréquence de commutation est variable, se qui conduit à des pertes de commutation, ceci est l'inconvénient majeur de la DTC. De la figure 2.12.h, on peut calculer la fréquence de commutation qui est en moyenne de 8 KHz.

On a jugé utile de déterminer les performances de la commande DTC de la machine asynchrone fonctionnant à basses vitesses. Le flux statorique étant maintenu à sa valeur nominale. Le couple de référence est choisi égal à 1 N.m. le flux statorique est reconstruit à l'aide d'un estimateur. Les résultats obtenus sont illustrés par la figure 2.13. On peut remarquer que le flux statorique est affecté par le phénomène de démagnétisation qui affecte la stratégie DTC à basses vitesses.



a. Couple (N.m)



b. Flux statorique(Wb)

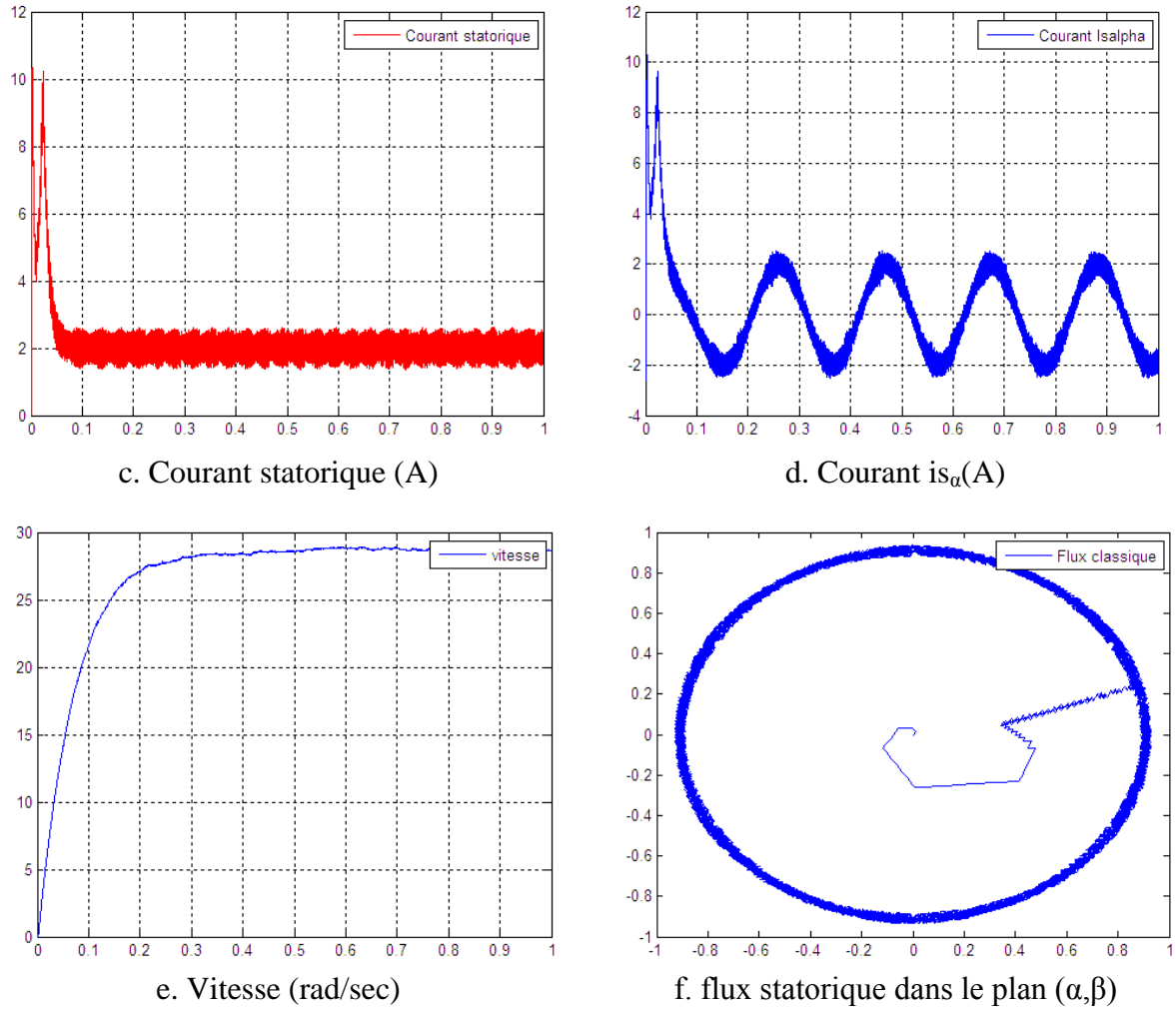


Figure 2.13. Résultats de la commande DTC de Takahashi à base d'un estimateur de flux et sans variation de la résistance statorique.

La simulation suivante étudie les performances de la commande DTC vis-à-vis des variations de la résistance du stator. En considérant le régime de fonctionnement précédent et à partir du régime permanent en charge, on augmente brusquement la résistance du stator de 50% à l'instant 0.4s. Les résultats obtenus sont illustrés par la figure 2.14. Le désaccord entre les valeurs de la résistance statoriques adoptée dans le modèle de la MAS et celle dans l'estimateur du flux statorique mène à une dégradation des performances de la DTC. Le flux statorique n'est pas maintenu constant et la machine a tendance à s'arrêter.

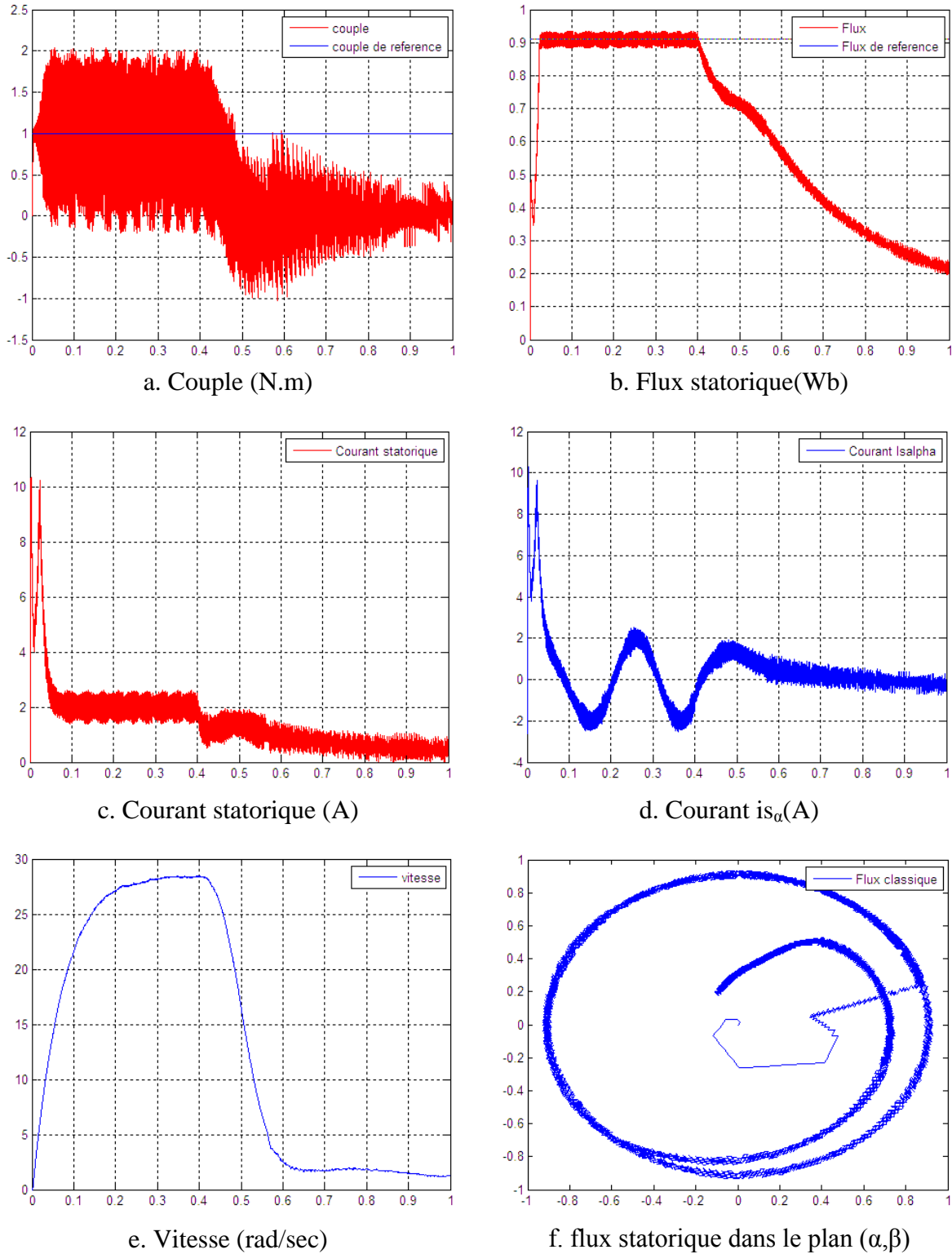
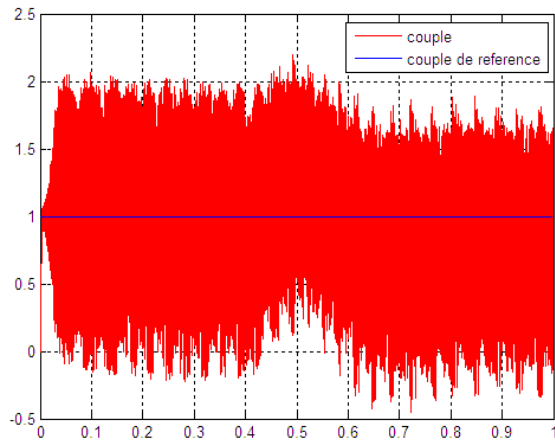


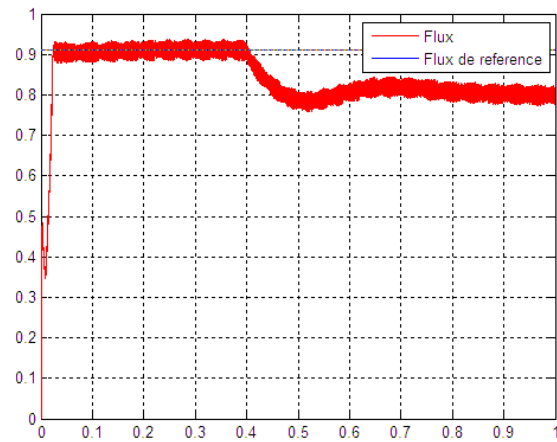
Figure 2.14. Résultats de la commande DTC de Takahashi à base d'un estimateur de flux et avec variation de la résistance statorique.

Afin de palier en partie les insuffisances enregistrées lors du fonctionnement de la machine asynchrone à basse vitesse et avec variation de la résistance du stator, on a inclut dans la structure de commande de la DTC un observateur de flux par mode glissant. En considérant le même régime de fonctionnement que précédemment avec une augmentation de 50% de la

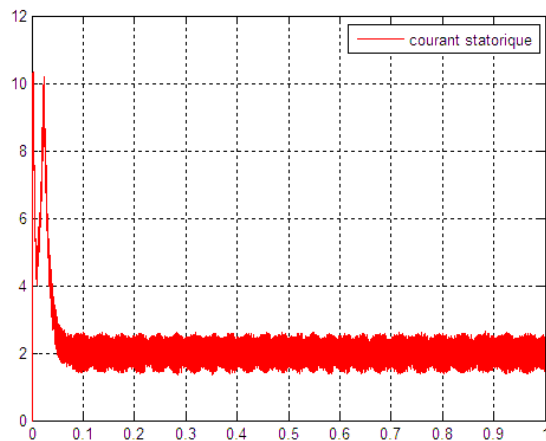
résistance du stator. Les résultats sont récapitulés par la figure 2.15. Par rapport au cas précédent, on peut remarquer une nette amélioration, le flux statorique subit une faible atténuation et la vitesse du moteur décroît par rapport au régime initial.



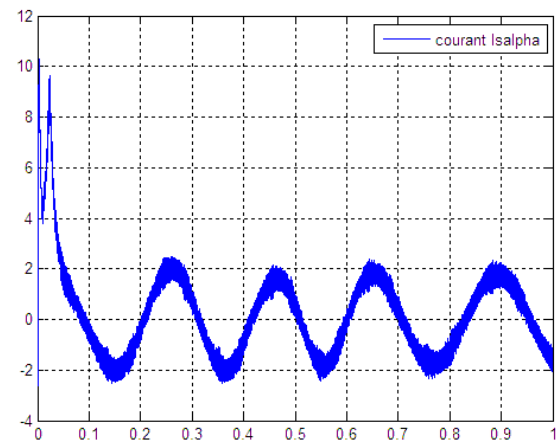
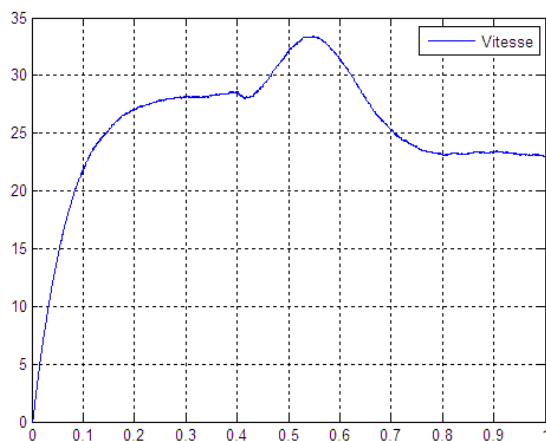
a. Couple (N.m)



b. Flux statorique(Wb)



c. Courant statorique (A)

d. Courant  $i_{s\alpha}$ (A)

e. Vitesse (rad/sec)

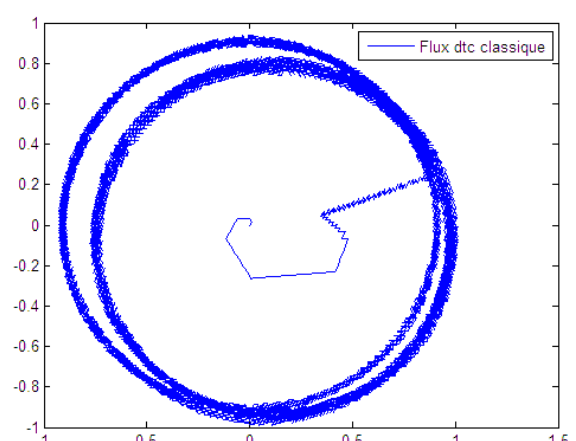
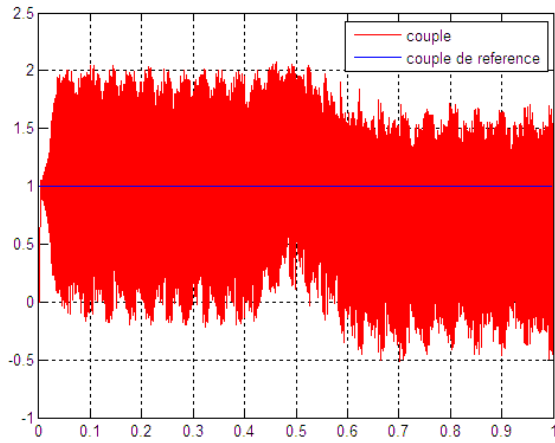
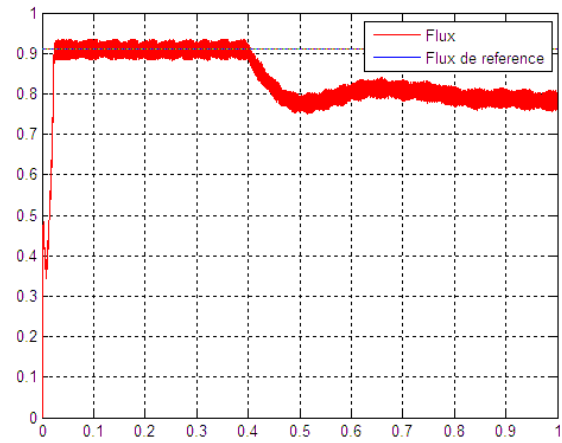
f. flux statorique dans le plan  $(\alpha,\beta)$

Figure 2.15. Résultats de la commande DTC de Takahashi à base d'un observateur par mode glissant et avec variation de la résistance statorique.

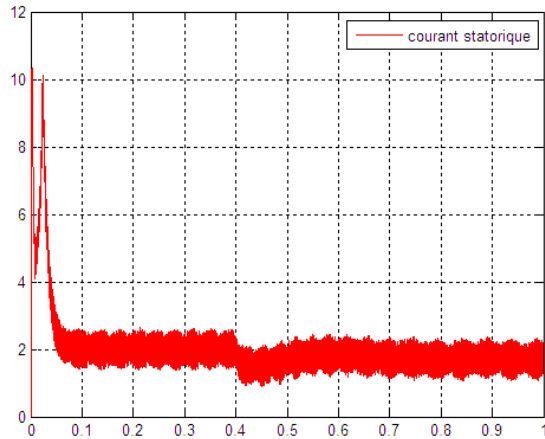
En outre, pour le même régime de fonctionnement que précédemment, on a substitué l'observateur par mode glissant par celui à grand gain. Les résultats sont récapitulés dans la figure 2.16. En comparant les résultats de la commande DTC obtenus via l'exploitation des deux observateurs on peut remarquer que l'observateur par mode glissant assure le meilleur régime de fonctionnement. La vitesse de rotation garde un niveau supérieur par rapport au cas de l'observateur à grand gain. La commande DTC munie de l'observateur par mode glissant est plus robuste que celle qui utilise l'observateur à grand gain.



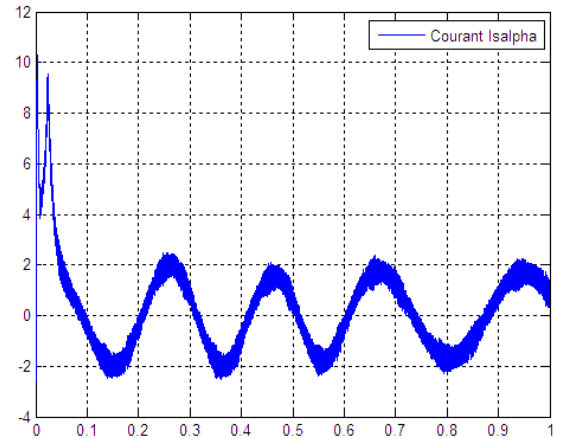
a. Couple (N.m)



b. Flux statorique(Wb)



c. Courant statorique (A)



d. Courant  $i_{s\alpha}$ (A)



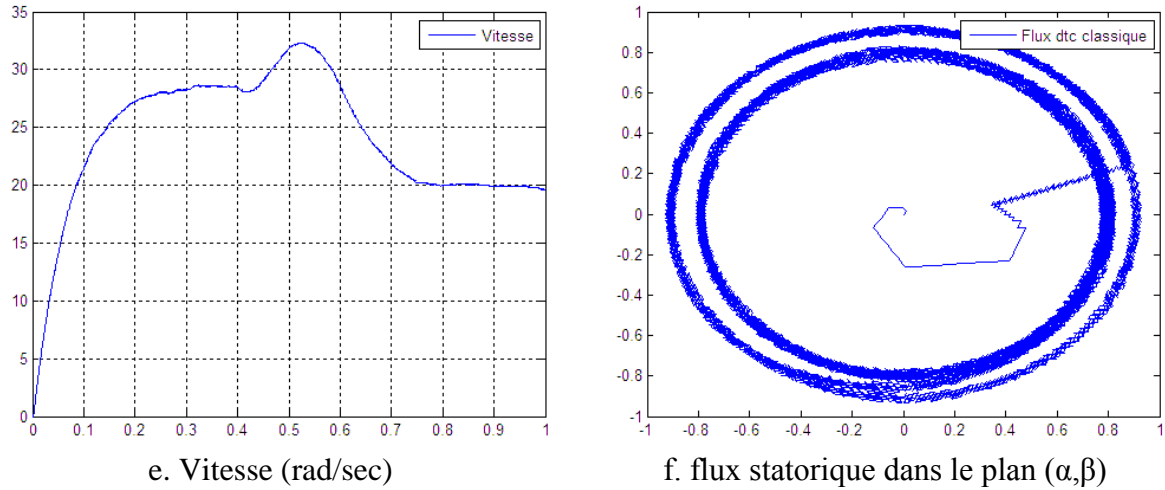
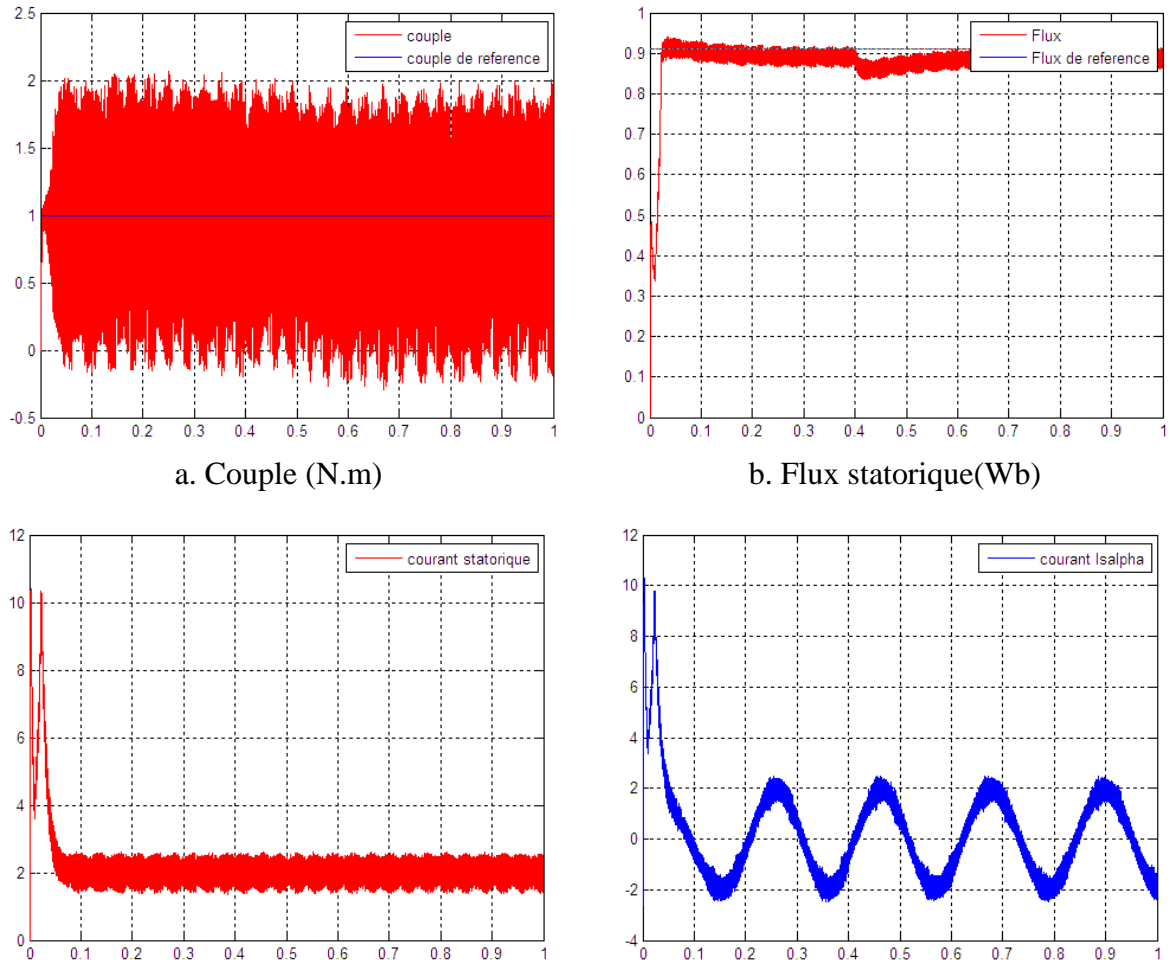


Figure 2.16. Résultats de la commande DTC de Takahashi à base d'un observateur à grand gain et avec variation de la résistance statorique.

La dernière simulation a porté sur la commande DTC en utilisant un observateur par mode glissant adaptatif. L'augmentation de la résistance du stator est accompagnée par le mécanisme d'adaptation de cette résistance. Les résultats sont résumés dans la figure 2.17. On peut remarquer que malgré la variation de la résistance du stator, le flux statorique est maintenu constant à sa valeur nominale.



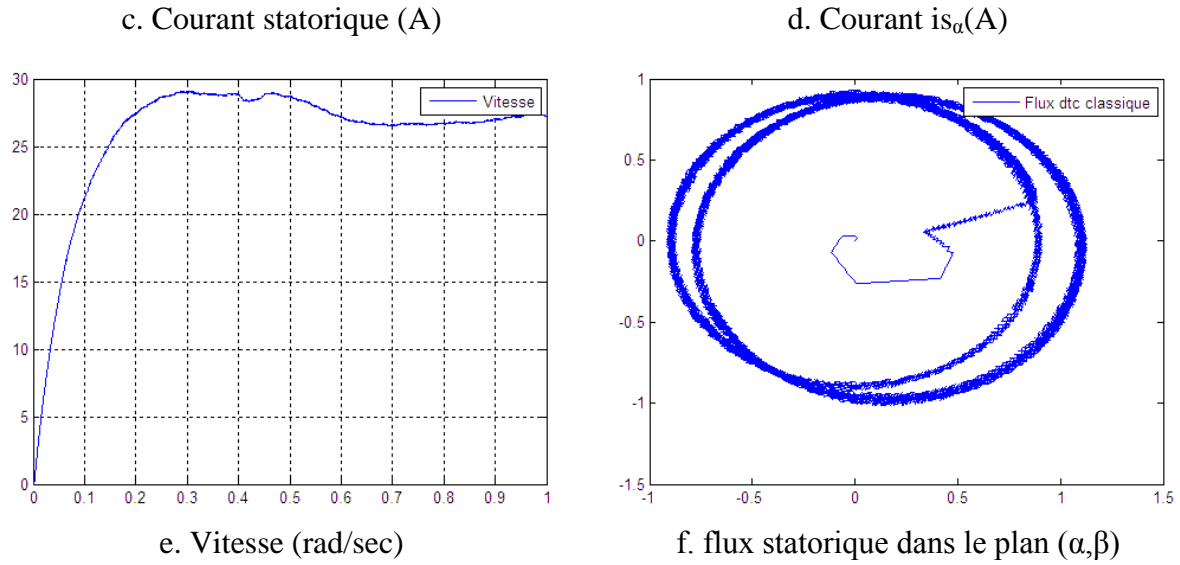


Figure 2.17. Résultats de la commande DTC de Takahashi à base d'un observateur par mode glissant équipé d'un mécanisme d'adaptation de la résistance statorique.

## 2.6 Conclusion

Dans la première partie de ce chapitre, on a présenté le modèle électrique global de la machine asynchrone. Ensuite, on a mis en œuvre une stratégie de commande DTC basée sur le principe le découplage entre le couple et le flux. Dans la deuxième partie, deux approches d'observation non linéaires par mode glissant et à grand gain du flux statorique de la machine asynchrone ont été mis en œuvre. Ces deux approches ont permis de répondre principalement aux besoins les plus critiques des lois de commande de la MAS en matière de robustesse vis-à-vis les variations des paramètres de la machine et assurent un bon fonctionnement sur toute la plage de vitesse. Enfin, on a présenté et on a comparé les résultats d'une étude d'exploration des performances de la commande DTC avec observateur et estimateur du flux statorique d'une machine asynchrone alimentée via un onduleur de tension. L'observateur par mode glissant a donné les meilleures performances en matière d'observation du flux statorique et de robustesse paramétrique. Il sera retenu dans toute la suite de cette thèse.

Bien que la commande DTC offre une réponse précise et rapide du couple électromagnétique et du flux statorique, Cependant, les inconvénients majeurs de cette commande sont liés à l'existence des ondulations importantes dans le couple et la variation de la fréquence de commutation due à l'utilisation des comparateurs à hystérésis. L'objet du prochain chapitre est de développer une solution pour pallier à ces problèmes via l'utilisation des techniques intelligentes.

## **Chapitre 3 : Commande DTC d'une MAS via l'utilisation des techniques intelligentes**

### 3.1 Introduction

Actuellement les techniques de l'intelligence artificielle sont connues par leur pouvoir de résoudre les problèmes liés à l'automatisation des processus industriels, tel que le contrôle, la commande, l'identification et l'estimation des paramètres des systèmes électriques. La technique intelligente basée sur la logique floue appartient à la classe des systèmes à base modèle de connaissance. Elle s'applique de plus en plus dans le contrôle de la machine asynchrone et l'adaptation de sa commande [LoBa99] [BibP07] [KoKa08]. Les approches basées sur la logique floue sont considérées comme une solution très puissante pour la commande de systèmes non linéaires ou les systèmes pour les quels il n'existe pas des modèles mathématiques. Les réseaux de neurones se distinguent par leur capacité de traitement, d'apprentissage et d'approximation qui est une autre technique de l'intelligence artificielle. Ils sont considérés comme une classe complètement différente des calculateurs classiques [NgNg10] [SmGa09].

On propose dans ce chapitre de développer une commande DTC en exploitant les techniques intelligentes. Ce chapitre se décompose en deux parties. Dans la première partie, on s'intéresse à la commande DTC à base de la logique floue. Il paraît nécessaire de commencer par présenter la logique floue et d'explicitier de manière simple sa mise en œuvre. Ensuite on va développer la commande directe floue de couple (DTFC) de la machine asynchrone. Puis, une étude comparative entre les résultats de simulation de la DTFC et celle de la commande DTC conventionnelle est présentée et discutée. La deuxième partie est consacrée à la commande DTC à base de réseaux de neurones. On commence par présenter une description théorique des réseaux de neurones artificiels, ensuite on va développer la commande directe neuronale de couple (DTNC) de la machine asynchrone. Cette partie s'achèvera par une étude comparative entre les résultats de la DTNC et la commande DTC conventionnelle. Ce chapitre est clôturé par une étude comparative entre la commande DTFC et la commande DTNC.

### 3.2 Commande directe floue de couple (DTFC) d'une MAS

#### 3.2.1 Introduction

La théorie de la logique floue a été établie par L. Zadeh en 1965 [LoZa65]. Cette logique permet la représentation et le traitement de connaissances imprécises ou approximatives. Le développement de cette logique se fait à travers les méthodes par lesquelles l'homme essaye de reproduire le comportement et le mode de raisonnement qui lui sont propres. Le nombre d'applications basées sur la théorie de la logique floue a augmenté considérablement ces dernières années. Ceci est dû au fait que la logique floue est exprimée usuellement par des règles linguistiques de la forme *IF THEN-ELSE* [ ]; elle est utilisée pour résoudre les problèmes de décision dans le domaine de contrôle ou bien pour décrire le comportement dynamique d'un système inconnu ou mal défini [ChKh06]. Dans la référence [RaPr11], un état de l'art sur l'utilisation de la logique floue dans le secteur industriel a été présenté. A titre d'exemple, on cite dans ce qui suit un ensemble de travaux afférant à la commande floue de la machine asynchrone.

L'avancement des recherches dans le domaine de la logique floue a montré la capacité et la puissance des modèles flous dans la commande des systèmes électriques non linéaires. Plusieurs chercheurs utilisent cette puissance d'apprentissage pour commander les systèmes électriques, particulièrement les machines asynchrones [AbbA09] [TeRa12]. Dans [KoKa08], l'auteur a traité la commande adaptative floue et l'estimation de la vitesse d'une machine asynchrone. Pour la régulation de la vitesse de la machine asynchrone, il a utilisé un régulateur PI-flou à gains adaptatifs flous muni d'une nouvelle technique d'estimation floue de la vitesse basée sur la stratégie MRAS. Les travaux de recherche présentés dans la thèse de [ChKh06] portent essentiellement sur deux axes : le développement d'un algorithme de modélisation et d'identification flou capable de modéliser et d'identifier des processus non linéaires, ainsi que la mise en œuvre de deux structures de commande adaptative directe et indirecte basées sur la théorie de la logique floue. Dans le premier axe, une nouvelle méthode combinant l'algorithme de regroupement flou et le filtre de Kalman a été proposée. Dans le deuxième axe, l'auteur a introduit la logique floue type-2, qui a été utilisée dans la commande adaptative d'une certaine classe de systèmes non linéaires et inconnus.

L'auteur de [KaHa09] a proposé un modèle qui offre la meilleure stabilité d'un véhicule électrique sur une route courbée. Le système de traction proposé se compose de deux machines qui assurent l'entraînement de deux roues motrices arrière. La structure de commande proposée est basée sur la commande DTC floue pour chaque roue motrice. Les résultats de simulation montrent une bonne stabilité du véhicule sur une route droite, une route droite de pente, et une route courbée à gauche et à droite.

Dans la référence [SmGa09], les auteurs présentent une comparaison entre quatre différentes stratégies de contrôle de vitesse d'une machine asynchrone basées sur les techniques de l'intelligence artificielle. Deux stratégies sont élaborées en utilisant des régulateurs PI conventionnels, la troisième stratégie exploite un contrôleur à logique floue et la dernière est conçue à partir de la théorie floue hybride. Une comparaison est établie entre les différents contrôleurs suivant un indice de performance fondé sur l'erreur de vitesse. Toutes les méthodes sont appliquées à la commande DTC et chaque stratégie de contrôle a été testée pour évaluer sa robustesse.

M. Nasir Uddin et M. Hafeez ont présenté dans [MnUd12] un comparateur à hystérésis basé sur la logique floue pour la commande DTC d'une machine asynchrone. La logique floue est utilisée pour ajuster la largeur de la bande du régulateur d'hystérésis du couple afin de réduire les ondulations du couple et du flux et par conséquent, améliorer la réponse dynamique de la machine. Les effets de la largeur de la bande d'hystérésis sur l'amplitude des ondulations de couple de la MAS sont également abordés dans cet article. Suivant la valeur du couple estimé et du courant statorique, le contrôleur flou permet de sélectionner la largeur de la bande optimale du régulateur à hystérésis de couple. Afin de tester les performances du contrôleur flou proposé, une implémentation en temps réel sur un banc d'essai à base d'une carte DSP a été effectuée.

Dans notre travail, on s'intéresse à l'amélioration des performances de la commande des machines asynchrones, particulièrement la commande DTC. Cette commande, déjà étudiée

dans le chapitre précédent, présente un inconvénient majeur lié à l'existence des ondulations importantes dans le couple et le flux statorique. Dans cette première partie de ce chapitre, on va proposer une solution à ces problèmes par l'utilisation d'un contrôleur intelligent basé sur la logique floue.

### 3.2.2 Principe de la logique floue

Le principe de réglage par logique floue s'approche de la démarche humaine du fait que les variables traitées ne sont pas des variables logiques mais des variables linguistiques, proches du langage humain. De plus, ces variables linguistiques sont traitées à l'aide des règles qui font référence à une certaine connaissance du comportement du système [MCir02]. Toute une série de notions fondamentales est développée dans la logique floue. Ces notions permettent de démontrer et de justifier certains principes de base.

#### 3.2.2.1 Ensembles flous

Dans la théorie des ensembles conventionnels, une chose appartient ou n'appartient pas à un certains ensembles [CirM02]. La théorie de ces ensembles flous repose donc sur la notion d'appartenance partielle : chaque élément appartient partiellement ou graduellement aux ensembles flous définis au préalable. Le contour de chaque ensemble flou de la figure 3.1 n'est pas "net", mais "flou" ou "graduel", ce qui différencie les deux théories sur les ensembles. Dans les ensembles flous, la permission qu'une chose appartienne partiellement à un certain ensemble s'appelle le degré d'appartenance. ZADEH a défini les ensembles flous comme étant des termes linguistiques du genre : zéro, grand, négatif, petit...



Figure 3.1. Comparaison d'un ensemble classique et d'un ensemble flou

Dans les ensembles conventionnels, le degré d'appartenance est 0 ou 1 alors que dans la théorie des ensembles flous, le degré d'appartenance peut varier entre 0 et 1, on parle alors de fonction d'appartenance  $\mu$ .

Un exemple simple d'ensembles flous est la classification des personnes selon leurs âges en trois ensembles : jeune, moyen et vieux. Deux façons sont possibles pour établir cette classification. Contrairement à la 1<sup>ère</sup> représentation, la 2<sup>ème</sup> montre que les limites entre ces trois groupes ne varient pas soudainement, mais progressivement. Par exemple, une personne de 34 ans appartient à l'ensemble "jeune" avec une valeur  $\mu=0.75$  de la fonction d'appartenance et à l'ensemble moyen avec une valeur  $\mu=0.25$ . La figure 3.2 donne le degré d'appartenance d'une personne, selon son âge, à un certain ensemble flou. Pour cette raison, ce type de figure s'appelle « fonction d'appartenance ».

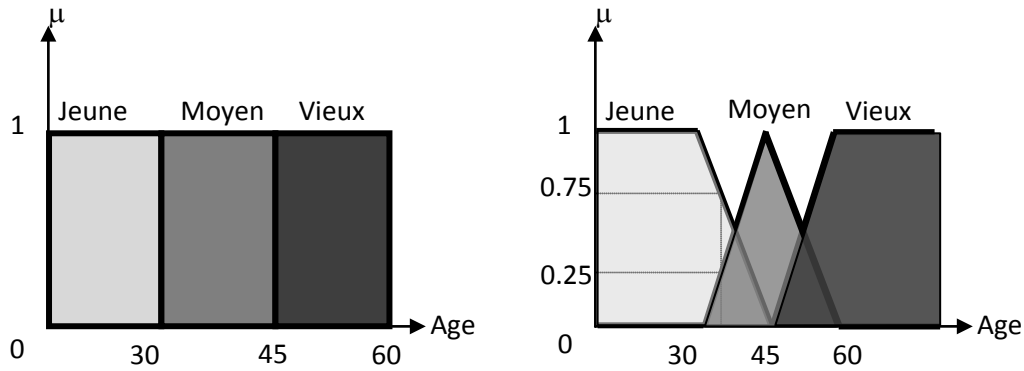


Figure 3.2. Classification des personnes suivant leurs âges

On peut ainsi résumer la terminologie pour cet exemple dans l'illustration suivante :

Variable linguistique : âge

Valeur d'une variable linguistique : jeune, moyen, vieux...

Ensembles flous : jeune, moyen, vieux

Plage de valeurs : (0, 30, 45, 60...)

Degré d'appartenance à un ensemble flou E :  $\mu_E(x)=a$ , avec  $0 < a < 1$ .

### 3.2.2.2 Différentes formes de fonction d'appartenance

En général, on utilise pour les fonctions d'appartenance trois formes géométriques : trapézoïdale, triangulaire et de cloche. Les deux premières sont les plus souvent employées en raison de leurs simplicités. L'allure de la forme trapézoïdale est complètement définie par quatre points  $P_1$ ,  $P_2$ ,  $P_3$  et  $P_4$  comme présenté par la figure 3.3. Le triangle peut être considéré un cas particulier du trapèze lorsque deux points coïncident ( $P_2=P_3$ ). La forme rectangulaire est comprise dans le trapèze si ( $P_1, P_2$ ) et ( $P_3, P_4$ ) se trouvent sur une verticale.

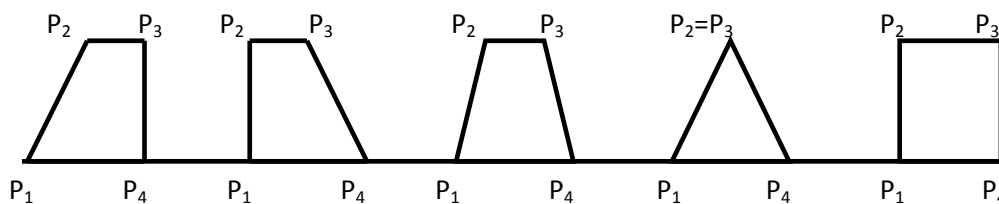


Figure 3.3. Fonctions de forme trapézoïdale

Dans la plupart des cas, en particulier pour le réglage par logique floue, les deux premières formes sont suffisantes pour délimiter des ensembles flous. Cependant, la forme de cloche qui donne dans certains cas une meilleure représentation des ensembles flous, peut être aussi utilisée.

### 3.2.2.3 Opérateurs de la logique floue

Une fois les ensembles flous sont définis, des opérateurs mathématiques concernant ce type d'ensemble ont été développés. Les opérateurs élaborés ressemblent beaucoup à ceux reliés à la théorie des ensembles conventionnels. Les opérateurs d'union, d'intersection et de négation

existent pour les deux types d'ensemble. Les opérateurs habituels, soit l'addition, la soustraction, la division et la multiplication de deux ou plusieurs ensembles flous existent aussi. Toutefois, ce sont les deux opérateurs d'union et d'intersection qu'on utilise le plus souvent dans la commande par la logique floue [MCir02].

Ces opérateurs permettent d'écrire des combinaisons logiques entre les notions floues, c'est-à-dire de faire des calculs sur des degrés de vérité. Comme pour la logique conventionnelle, on peut définir des opérateurs ET, OU et négation.

Les formulations utilisées pour l'intersection et l'union sont respectivement des normes triangulaires (T-normes) et les conormes triangulaires (T-conormes ou S-normes). Soient A et B deux ensembles flous ayant respectivement  $\mu_A$  et  $\mu_B$  comme fonctions d'appartenance. L'intersection de A et B est un ensemble flou  $A \cap B$  dont la fonction d'appartenance est définie selon le choix. Il en est de même pour l'union  $A \cup B$  et le complément, le tableau 3.1 présente le choix des différentes T-normes et S-normes :

Tableau 3.1. Différentes T-normes et S-normes

	<b>T-norme <math>A \cap B</math></b>	<b>S-norme <math>A \cup B</math></b>
<b>Zadeh</b>	$\text{Min}(\mu_A, \mu_B)$	$\text{Max}(\mu_A, \mu_B)$
<b>Bandler</b>	$\mu_A \cdot \mu_B$	$\mu_A + \mu_B - \mu_A \cdot \mu_B$
<b>Lukasiewicz</b>	$\text{Max}(0, \mu_A + \mu_B - 1)$	$\text{Min}(1, \mu_A + \mu_B)$

Le complément d'un ensemble flou A est l'ensemble flou  $\bar{A}$  dont la fonction d'appartenance  $\mu_{\bar{A}}$  vérifie  $\mu_{\bar{A}} = 1 - \mu_A$ . On peut directement constater que la logique conventionnelle est un cas particulier de la logique floue.

### 3.2.3 Commande par logique floue

La commande par logique floue permet d'obtenir une loi de réglage souvent très efficace sans avoir recours à des modélisations approfondies. En se basant sur des variables linguistiques, le régulateur flou utilise des inférences avec plusieurs règles. Par ces derniers, il est possible de tenir compte des expériences acquises par les opérateurs d'un processus technique.

Dans la suite, on présente les bases générales de la commande par logique floue et la procédure de la conception d'un tel réglage. Ceci va permettre d'appliquer cette technique dans la commande des machines asynchrones.

#### 3.2.3.1 Principes généraux d'une commande par logique floue

Le schéma synoptique de la figure 3.4 résume le fonctionnement de la commande par logique floue. Il comporte quatre blocs principaux : La fuzzification, la base de connaissance, l'inférence et la défuzzification.



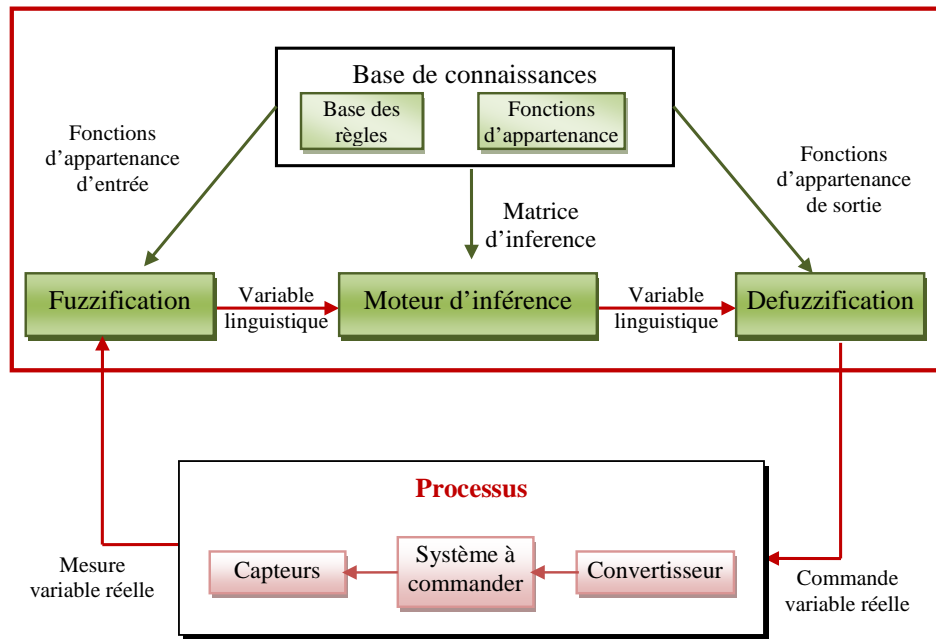


Figure 3.4. Principe de la commande par logique floue

Les blocs cités au paravent permettent de générer respectivement les fonctions suivantes :

- i) Bloc de fuzzification : Il établit d'une part les plages de valeurs pour les fonctions d'appartenance à partir des valeurs des variables d'entrées et d'autre part il effectue une conversion des données d'entrée en valeurs linguistiques convenables considérées comme l'étiquette des ensembles flous.
- ii) Bloc de base de connaissance : Il comporte une connaissance dans le domaine d'application ciblé et le résultat de commande prévu. Il est constitué par une "base de données" et une "base de règles linguistiques floues de commande". La base de données effectue des définitions qui sont nécessaires pour établir les règles de commande et manipuler les données floues. La base de règles représente la stratégie de commande obtenue par le biais des règles de commande linguistiques.
- iii) Bloc moteur d'inférence : C'est le cœur de la commande floue, il possède la capacité de simuler les décisions humaines et de déduire les actions de commande floue à l'aide de l'implication floue et des règles d'inférence dans la logique floue.
- iv) Bloc défuzzification : Il établit les plages de valeurs pour les fonctions d'appartenance à partir des valeurs des variables de sortie et effectue une défuzzification qui fournit un signal de commande non-flou à partir du signal flou déduit.

Comme le système à commander ne reçoit que des valeurs déterministes, la commande floue convertit les valeurs déterministes à son entrée en valeurs floues, les traite avec les règles floues et reconvertit le signal de commande de valeurs floues en valeurs déterministes.

### 3.2.3.2 Avantages et inconvénients de la commande par logique floue

La commande par logique floue réunit un certain nombre d'avantages et d'inconvénients. Les avantages essentiels sont énumérés ainsi [KoKa08] [BiBo06] :

- La non nécessité d'une modélisation. Cependant, il peut être utile de disposer d'un modèle convenable.
- La possibilité d'implanter des connaissances (linguistiques) de l'opérateur,
- La maîtrise du procédé avec un comportement complexe (fortement non-linéaire et difficile à modéliser),
- L'obtention fréquente de meilleures prestations dynamiques,
- Deux solutions sont possibles pour l'implémentation d'une telle commande : solution logicielle (microprocesseur, DSP et PC) ou solution matérielle (fuzzy processeurs) [BibP07].

Les inconvénients de la commande par logique floue sont résumés par [KoKa08] :

- L'absence de guide précis pour la conception d'un réglage (choix des grandeurs à traiter, détermination de la phase de fuzzification, des inférences et de la défuzzification),
- L'approche artisanale et non systématique (implantation des connaissances de l'opérateur souvent difficile),
- La cohérence des inférences est non garantie à priori (apparition de règles d'inférence contradictoires possible).

### 3.2.4 Commande directe floue de couple de la machine asynchrone

La commande DTC conventionnelle offre une réponse rapide et précise du couple électromagnétique et du flux statorique. Par contre, les inconvénients majeurs de cette commande sont l'existence des ondulations importantes dans le couple, le flux statorique et le courant due à l'utilisation des comparateurs à hystérésis [DenJ06]]. Dans cette partie, la commande directe floue du couple (DTFC : Direct Torque Fuzzy Control) a été proposée afin d'améliorer les performances de la commande DTC conventionnelle. Cette méthode propose de remplacer les comparateurs à hystérésis et la table de sélection par un contrôleur basé sur un système d'inférence flou [SoGd09] [SoGd10b].

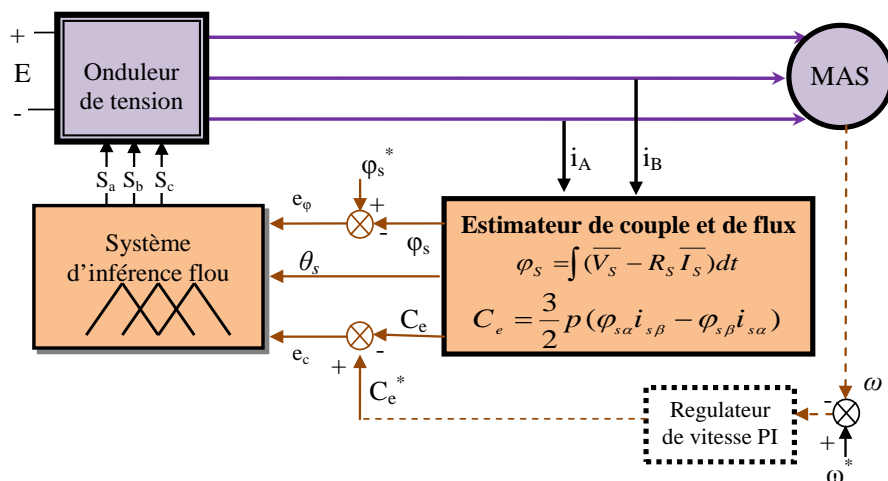


Figure 3.5. Schéma de principe de la commande directe floue de la MA

La figure 3.5 montre le schéma de principe de la commande DTFC d'une machine asynchrone. Les erreurs de couple et de flux obtenues ainsi que l'angle  $\theta_s$  sont requises par le

système d'inférence flou afin d'évaluer le vecteur tension de référence pour conduire le couple et le flux vers leurs valeurs désirées.

### 3.2.4.1 Stratégie de fuzzification

Les figures 3.6 à 3.8 présentent les fonctions d'appartenance pour les variables d'entrées et de sorties du système d'inférence flou. Des fonctions d'appartenance trapézoïdales et triangulaires ont été choisies. L'entrée erreur de flux est composée de trois ensembles flous N (négatif), Z (zéro) et P (positif). Cinq ensembles flous forment les fonctions d'appartenance de l'erreur de couple, NG (négatif grand), NP (négatif petit), Z (zéro), PP (positif petit) et PG (positif grand) ont été considérés [SoGd09].

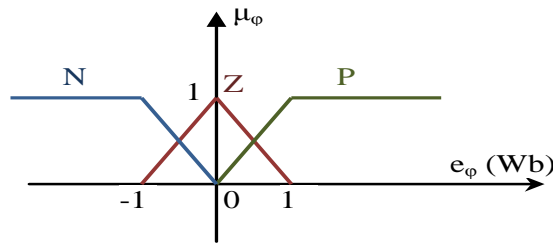


Figure 3.6. Les fonctions d'appartenance de l'erreur de flux  $e_\phi$ .

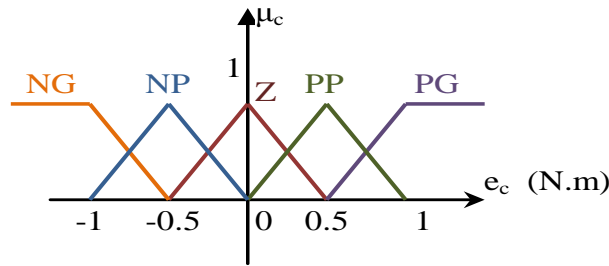


Figure 3.7. Les fonctions d'appartenance de l'erreur de couple  $e_c$ .

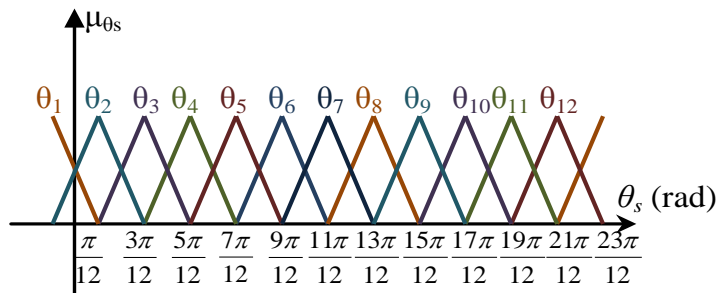


Figure 3.8. Les fonctions d'appartenance de l'angle  $\theta_s$ .

Afin d'avoir un réglage fin, le discours de l'univers de l'angle du vecteur flux statorique est divisé en douze ensembles flous de  $\theta_1$  à  $\theta_{12}$ . Le nouveau espace d'évolution du flux statorique dans le plan  $(\alpha, \beta)$  sur 12 secteurs est introduit dans la figure 3.9 [AriA00].

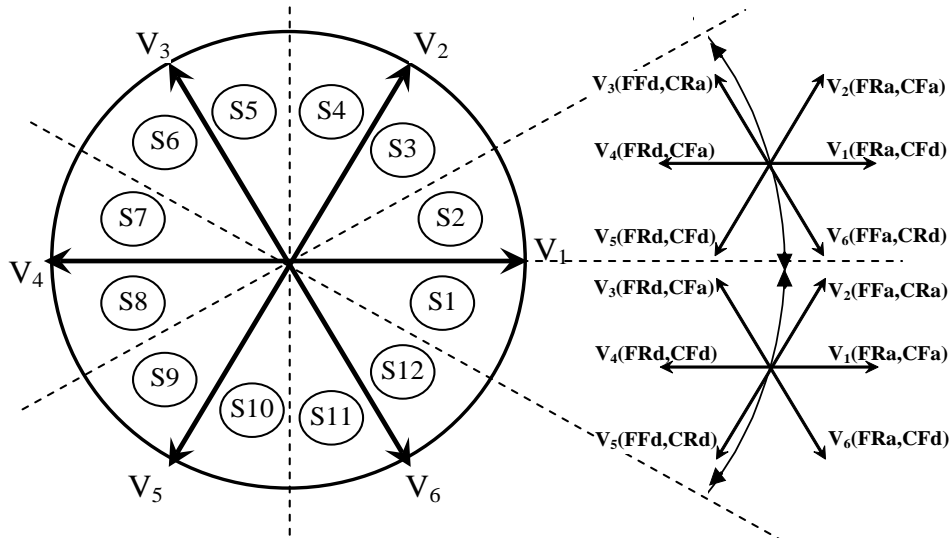


Figure 3.9. Vecteur flux statorique dans le plan  $(\alpha, \beta)$ . FRa/FRd: flux rapide augmentation /diminution. FFa/FFd: flux faible augmentation /diminution. CRa/CRd: Couple rapide augmentation /diminution. CFa/CFd: Couple faible augmentation /diminution.

La variable de sortie du système d'inférence est divisée en huit singletons, deux tensions nulles ( $V_0$  et  $V_7$ ) et six tensions non nulles. Les fonctions d'appartenances de la variable de sortie sont illustrées dans la figure 3.10.

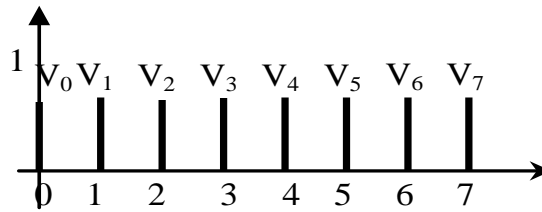


Figure 3.10. Les fonctions d'appartenances de la sortie

### 3.2.4.2 Base de règles

Le nombre des ensembles flous et celui des règles ne sont pas exhaustifs. On a choisi ces ensembles flous pour un meilleur résultat. Plus on augmente le nombre de ces ensembles plus on aura plus de règles et par conséquent le travail devient minutieux. Les différentes combinaisons possibles de cinq ensembles flous pour l'erreur de couple, trois ensembles flous pour l'erreur du flux et douze secteurs pour l'angle de flux statorique forment 180 règles dans la base du système d'inférence.

La base de règles est déduite du diagramme du flux statorique dans le plan  $(\alpha, \beta)$  illustré par la figure 3.9. Par exemple, si l'angle  $\theta_s$  du flux statorique est localisé dans l'intervalle  $\theta_2$ , lorsqu'on désire diminuer le couple doucement et augmenter le flux rapidement alors le vecteur  $V_1$  constitue le meilleur choix. Le même raisonnement est utilisé pour construire la base de règles pour la commande directe floue de couple présentée par le tableau 3.2.

Tableau 3.2. Base de règles adoptées

$e_\phi$	$e_T$	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$	$\theta_8$	$\theta_9$	$\theta_{10}$	$\theta_{11}$	$\theta_{12}$
P	PL	V1	V2	V2	V3	V3	V4	V4	V5	V5	V6	V6	V1
	PS	V2	V2	V3	V3	V4	V4	V5	V5	V6	V6	V1	V1
	Z	V0	V7	V7	V0	V0	V7	V7	V0	V0	V7	V7	V0
	NS	V6	V6	V1	V1	V2	V2	V3	V3	V4	V4	V5	V5
	NL	V6	V6	V1	V1	V2	V2	V3	V3	V4	V4	V5	V5
Z	PL	V2	V2	V3	V3	V4	V4	V5	V5	V6	V6	V1	V1
	PS	V2	V3	V3	V4	V4	V5	V5	V6	V6	V1	V1	V2
	Z	V7	V0	V0	V7	V7	V0	V0	V7	V7	V0	V0	V7
	NS	V7	V0	V0	V7	V7	V0	V0	V7	V7	V0	V0	V7
	NL	V5	V6	V6	V1	V1	V2	V2	V3	V3	V4	V4	V5
N	PL	V2	V3	V3	V4	V4	V5	V5	V6	V6	V1	V1	V2
	PS	V3	V3	V4	V4	V5	V5	V6	V6	V1	V1	V2	V2
	Z	V0	V7	V7	V0	V0	V7	V7	V0	V0	V7	V7	V0
	NS	V4	V5	V5	V6	V6	V1	V1	V2	V2	V3	V3	V4
	NL	V5	V5	V6	V6	V1	V1	V2	V2	V3	V3	V4	V4

La base de règles reflète le savoir acquis par l'opérateur humain qui manipule le processus à commander [HusF08]. Un système de règles floues permet de décrire, sous forme de règles linguistique, une fonction de transfert entre les variables d'entrées et celle de sorties d'un système [ChuC90]. Ces règles constituent le moteur d'inférence de la commande floue. Elles expriment un lien entre des propositions floues élémentaires ou des conjonctions de propositions élémentaires sous la forme suivante [ChuC90] :

$$R_i : \text{Si } e_\phi \text{ est } A_i \text{ et } e_c \text{ est } B_i \text{ et } \theta_s \text{ est } C_i \text{ alors } v \text{ est } V_i. \quad (3.1)$$

Avec  $A_i$ ,  $B_i$  et  $C_i$  désignent respectivement les variables linguistique de l'erreur flux, l'erreur couple et l'angle du flux statorique.  $V_i$  est la variable linguistique de sortie et  $R_i$  la règle numéro  $i$ .

### 3.2.4.3 Inférence des règles

Plusieurs approches sont proposées pour le traitement numérique des règles d'inférence à savoir [LoBa99] :

- i) Méthode d'inférence Max\_Min : Cette méthode réalise l'opérateur "ET" par la fonction "Minimum", le "Alors" de chaque règle d'inférence par la fonction "Minimum" et la liaison entre les règles (opérateur "OU") par la fonction Max.
- ii) Méthode d'inférence Max\_Produit : elle ressemble à la méthode précédente, sauf que la conclusion "ALORS" est réalisé par la fonction produit.
- iii) Méthode d'inférence Somme\_Produit : Dans ce cas il s'agit de faire une somme pondérée. Au niveau de la condition, l'opérateur ET est représenté par la fonction produit tandis que l'opérateur OU est réalisé par la fonction somme. Au niveau de la conclusion, l'opérateur OU est représenté par la fonction somme et l'implication (ALORS) est représentée par la fonction produit.

Dans le présent travail, on a utilisé la méthode d'inférence Max\_Min. Le degré d'activation  $\alpha_i$  de chaque règle "i" est calculé par l'équation (3.2) suivante :

$$\alpha_i = \min (\mu_{A_i}(e_\phi), \mu_{B_i}(e_c), \mu_{C_i}(\theta_s)). \quad (3.2)$$

$$\mu'_{Vi}(v) = \max(\alpha_i, \mu_{Vi}(v)) \quad (3.3)$$

Avec  $\mu_{Ai}(e_\varphi)$ ,  $\mu_{Bi}(e_c)$ ,  $\mu_{Ci}(\theta_s)$  et  $\mu_{Vi}(v)$  désignent respectivement les degrés d'appartenances de  $e_\varphi$ ,  $e_c$ ,  $\theta_s$  et  $v$  aux ensembles flous  $A_i$ ,  $B_i$ ,  $C_i$  et  $V_i$ .

Dans notre cas, la sortie est constituée par un ensemble de singleton, on va appliquer la méthode MAX comme illustrée par l'équation (3.4) :

$$\mu'_{Vout}(v) = \max_{i=1}^{180} (\mu'_{Vi}(v)) \quad (3.4)$$

### 3.2.4.4 La défuzzification

La défuzzification est l'opération qui permet de passer d'une représentation sous forme de variables linguistiques à une représentation sous forme de variables numériques applicable physiquement : c'est la phase inverse de la fuzzification.

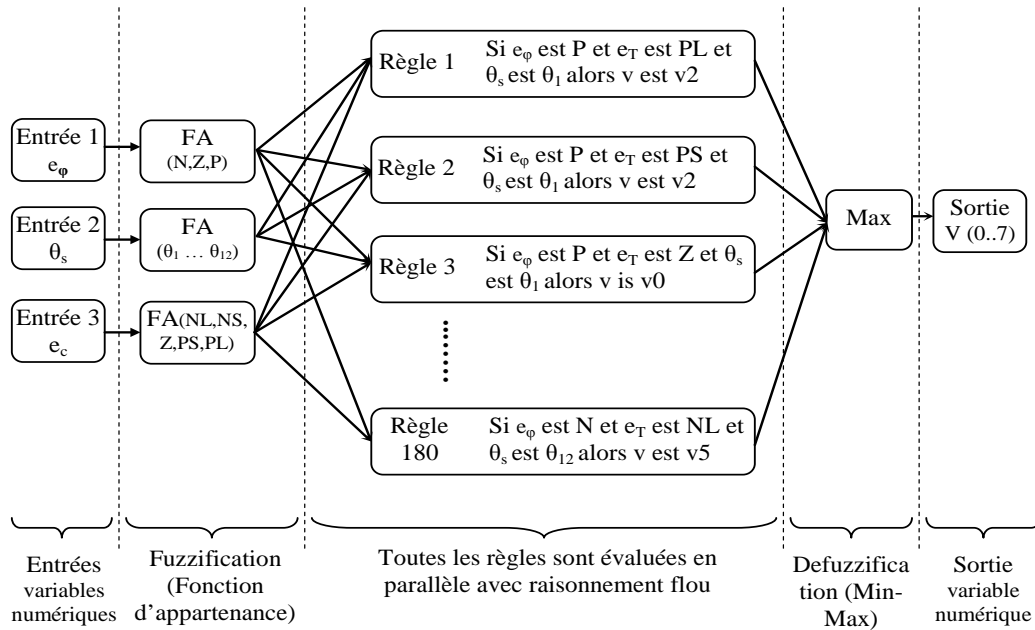


Figure 3.11. Diagramme de système d'inférence flou

Il existe plusieurs méthodes possibles de défuzzification. Les plus reconnues et exploitées sont celles qui sont basées sur les méthodes du centre de gravité et le calcul du maximum. Dans notre étude, on a utilisé la dernière méthode. Le diagramme de la figure 3.11 représente le raisonnement concret du système d'inférence flou. Pour chaque combinaison d'entrée, généralement plus qu'une règle est validée. Le degré d'activation de chaque règle est calculé en fonction des variables d'entrées. Ensuite l'étape de defuzzification est appliquée pour produire à la sortie la tension de commande.

### 3.2.5 Résultats de simulation

Après avoir introduit les principes de base de la logique floue, il s'avère primordiale de simuler le fonctionnement de la MAS utilisant une commande DTC floue.

Les différentes simulations effectuées concernant les mêmes régimes de fonctionnement que ceux introduits dans la commande DTC conventionnelle présentée dans le deuxième chapitre.

Les figures 3.12 (a,b) à 3.14 (a,b) illustrent une réduction importante des ondulations du couple, du flux et du courant de la machine asynchrone avec la commande DTC flou. La figure 3.15 décrit la trajectoire de l'amplitude du flux statorique dans le plan  $(\alpha, \beta)$ . On peut remarquer que la commande DTC floue est plus rapide que la commande conventionnelle. Au démarrage, les figures 3.12.c et 3.12.c montrent bien que l'établissement du flux statorique et du couple de la DTFC sont plus rapides.

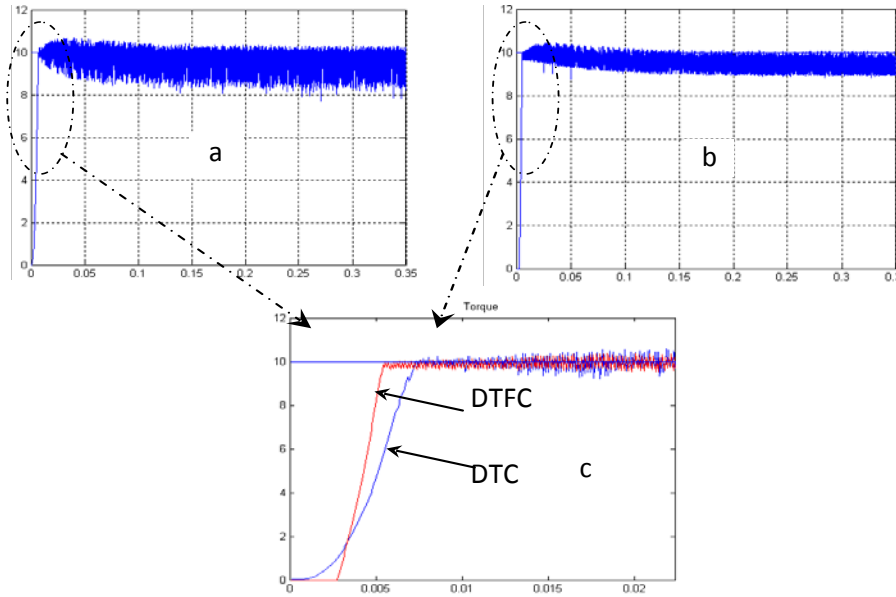


Figure 3.12. Couple électromagnétique. (a). DTC conventionnelle. (b). DTC flou. (c). comparaison DTC conventionnelle et DTC floue

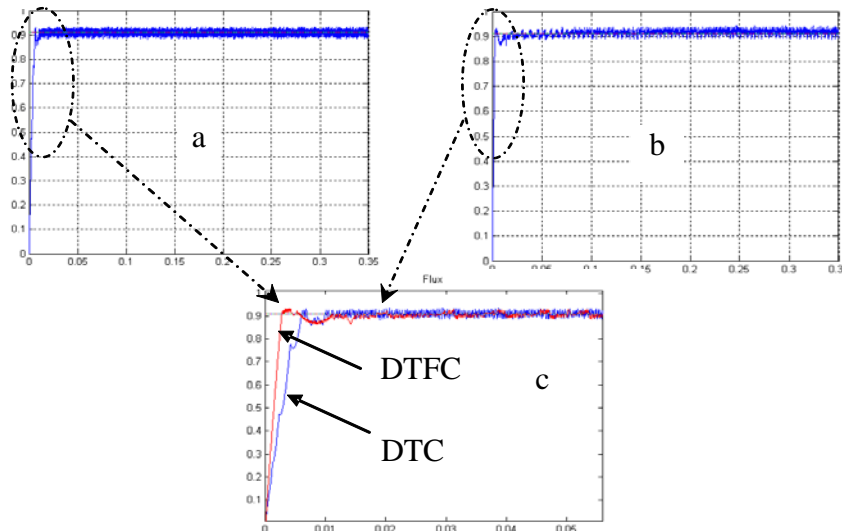


Figure 3.13 .Amplitude du flux statorique. (a). DTC conventionnelle. (b). DTC flou. (c). comparaison DTC conventionnelle et DTC floue

Les figures 3.16.a et 3.16.b illustrent l'état de commutation de l'interrupteur "Sa". Ces figures montrent que l'onduleur possède des phases de mise en veille des interrupteurs, ce qui permet

de diminuer les pertes de commutation et d'obtenir une fréquence de commutation presque constante autour de 5 kHz. Cette fréquence de commutation obtenue avec la commande directe floue de couple est inférieure à celle de la DTC conventionnelle.

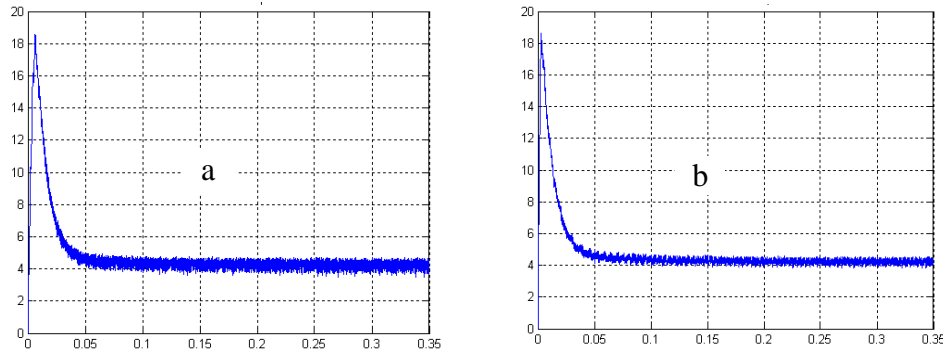


Figure 3.14. Le courant statorique. (a).DTC conventionnelle. (b). DTC floue.

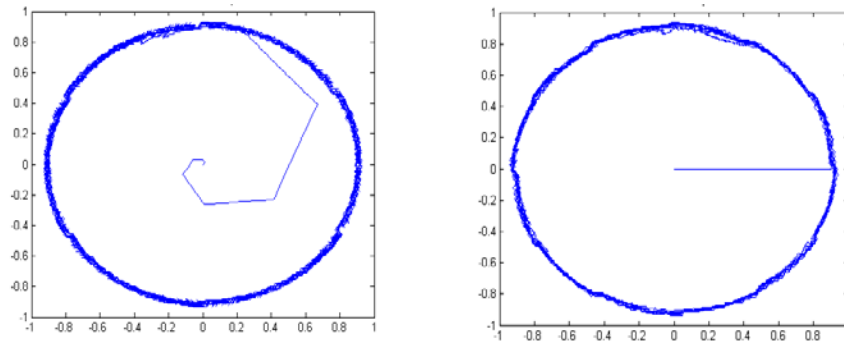


Figure 3.15. Amplitude du flux statorique dans le plan  $(\alpha, \beta)$ . (a).DTC conventionnelle. (b). DTC floue.

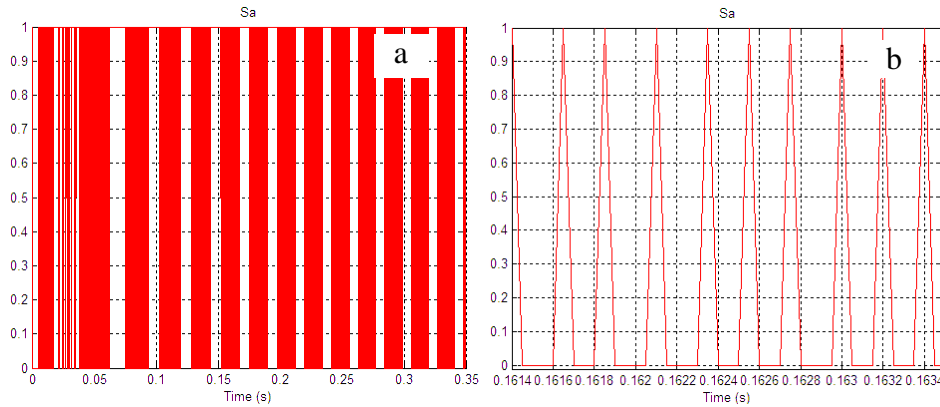


Figure 3.16. Commutation de l'interrupteur Sa. (a).DTC Floue. (b). DTC Floue (zoom).

Après cette analyse des résultats de simulations présentés dans cette partie, on peut dire que le comportement de la commande directe floue de couple est plus performant que celui de la commande DTC conventionnelle. Ceci met en relief les objectifs de ce chapitre qui consiste en une réduction des ondulations du couple, du flux et du courant statorique, tout en garantissant une bonne dynamique du couple.



### 3.2.6 Conclusion

Dans cette partie, on a développé la commande DTC en faisant appel à un système d'inférence flou pour remplacer les deux comparateurs à hystérésis et la table de sélection. La commande floue développée a comme entrée l'erreur de couple, l'erreur de flux et l'angle du flux statorique. La sortie permet d'évaluer le vecteur tension de référence pour conduire le couple et le flux vers leurs valeurs de référence durant une période de temps fixe. Les résultats de simulations montrent que le réglage flou possède une bonne dynamique de couple et de flux avec une réduction des ondulations assez remarquables.

Dans le paragraphe suivant, on va présenter la conception et le développement d'une autre technique de commande intelligente de la machine asynchrone à base d'un réseau de neurones. Dans cette étude, les réseaux de neurones sont choisis afin de profiter de leur capacité de traitement et essentiellement d'apprentissage. Une étude comparative entre les deux techniques abordées sera présentée et discutée à la fin de ce chapitre.

## 3.3 Commande Directe Neuronale de Couple (DTNC) d'une MAS

### 3.3.1 Introduction

Le réseau de neurones artificiels, qui est une technique est très populaire dans plusieurs domaines d'application de technologie et de recherche scientifique. Cette technique peut être utilisée dans des cas de problèmes difficiles qui ne peuvent pas être décrits par des approches mathématiques précises où ils sont très compliqués à manipuler [MoMo08]. Les champs d'application de ces réseaux de neurones sont très vastes : classification, traitement d'image et de parole, estimation et identification de processus [LinF01] [GoeA06], commande des systèmes électriques [BhiS06] [SivA09] [RajK08] [AbbA09]. Dans [AbHa12], les auteurs ont intégré les réseaux de neurones artificiels dans la commande d'une machine asynchrone. Les auteurs ont présenté en premier lieu des travaux exploitant les régulateurs Proportionnels-Intégrals (PI). En deuxième lieu, ils ont mentionné que dans certains cas, où la dynamique du système change avec le temps et/ou avec des conditions de fonctionnement, l'efficacité du régulateur PI se dégrade et la qualité du réglage se détériore. Pour pallier à ces problèmes et afin d'assurer une bonne performance de la commande, les auteurs ont intégré les réseaux de neurones artificiels dans la commande en vitesse. Plusieurs tests ont été simulés pour apprécier l'apport des RNA. Les résultats obtenus permettent d'affirmer une amélioration des performances et de robustesse dans la commande de la machine asynchrone.

A titre d'exemple, on présente quelques travaux de recherche afférant à l'exploitation des réseaux de neurones dans les applications de motorisation asynchrone fonctionnant à vitesse variables. Ngac Nguyen dans sa thèse [NgNg10], a proposé des approches basées sur les réseaux de neurones pour l'identification et la commande des applications exploitant le filtre actif parallèle et le moteur synchrone. Une structure neuronale a été développée pour réaliser toutes les fonctionnalités d'un filtre actif parallèle pour compenser les harmoniques de courant. Une commande neuronale a été mise en place pour la machine synchrone avec les contraintes réduisant les ondulations de couple. Elle consiste en des méthodes de commande directe en couple pour obtenir les courants statoriques optimaux qui donnent exactement le couple électromagnétique désiré. Ces commandes adoptent deux blocs neuronaux, l'un dédié

au calcul des courants optimaux et l'autre pour assurer leur génération à travers un onduleur de tension régulé en courant. La validation de ces approches neuronales est effectuée par des tests de simulation et par des essais expérimentaux. Une étude comparative avec les méthodes de commande conventionnelle montrent des excellentes caractéristiques en termes de performance et de robustesse.

Le but de la thèse de Adel Merabet [AdMe07] est la mise en œuvre d'une loi de commande non linéaire prédictive de haute performance pour une machine asynchrone. Deux méthodes sont utilisées pour la conception du modèle de prédiction ; l'une à partir des réseaux de neurones pour une commande prédictive neuronale, et l'autre utilise les outils de la géométrie différentielle pour la commande non linéaire prédictive. Dans la commande prédictive neuronale, une modélisation de la machine par un réseau de neurones de type multicouches est présentée pour la conception d'un prédicteur non linéaire. Les performances de robustesse aux variations de paramètres sont améliorées par ce contrôleur non linéaire prédictif. D'après les résultats obtenus, le travail réalisé avec la commande prédictive non linéaire constitue une contribution majeure dans le domaine des entraînements à vitesse variable pour les machines asynchrones.

Dans notre travail, on s'intéresse à l'amélioration des performances de la commande DTC des machines asynchrones. Il faut signaler que la commande DTC présente un inconvénient majeur qui est lié surtout à l'existence des ondulations importantes dans le couple et le flux statorique. Dans ce cadre, on propose une solution à ces problèmes par l'utilisation de la notion d'apprentissage par les réseaux de neurones.

### 3.3.2 Présentation générale

Les réseaux de neurones artificiels (RNA) constituent une présentation mathématique et informatique inspirée du cerveau humain. En effet, le réseau de neurones artificiel est capable d'exécuter certaines fonctions du cerveau humain telles que la mémorisation, l'apprentissage à base des exemples et la proposition de décision [HayS99]. Par analogie au cerveau humain qui est constitué de millions de cellules reliées entre elles. Ce modèle est formé de plusieurs unités appelées neurones, c'est une unité calculatoire caractérisée par "n" entrées liées entre elles transportant des données numériques jusqu'à la sortie. Le neurone représente l'unité de base d'un réseau de neurones. Ceci est illustré par la figure 3.17. Le modèle mathématique d'un neurone est donné par la relation (3.5) :

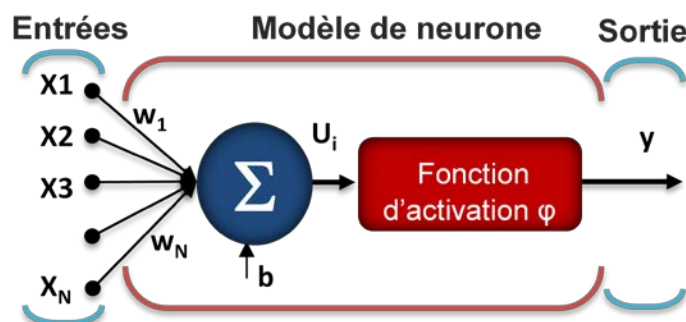


Figure 3.17. Unité de base : Le neurone

$$Y = \varphi \left( \sum_{i=1}^N w_i * X_i + b \right) \quad (3.5)$$

Avec les  $X_i$  et  $Y$  représentant les entrées et la sortie du neurone,  $b$  est le biais du neurone et les  $W_i$  sont les poids synaptiques des liaisons entre les entrées et la sortie. Ils mesurent l'importance de chaque connexion qui ne sont pas toutes égales. Le neurone formel va réaliser la somme pondérée des poids de différentes entrées qui représente l'activité de l'entrée. Ensuite, la fonction d'activation va calculer la sortie en fonction de cette somme. Le choix de cette fonction d'activation est un élément important des réseaux de neurones. Parmi ces fonctions, on peut citer : la fonction seuil, tangente hyperbolique, tangente gaussienne, ...

### 3.3.3 Architectures d'un réseau de neurones

Le réseau de neurones est constitué de plusieurs neurones qui sont organisés généralement sous forme de niveaux différents appelés couches de réseau. Les neurones appartenant à la même couche possèdent les mêmes caractéristiques et utilisent le même type de fonction d'activation. Le mode de connexion entre les neurones d'un réseau définit son architecture et influence son fonctionnement. En effet, il existe plusieurs modèles de réseaux de neurones qui diffèrent par leur topologie.

#### 3.3.3.1 Le perceptron :

C'est le premier modèle de réseau de neurones qui a été conçu par Rosenblatt en 1959. La figure 3.18 présente le modèle d'un perceptron. Il se caractérise par la présence de deux couches seulement : la couche d'entrée et la couche de sortie et les neurones sont totalement connectés.

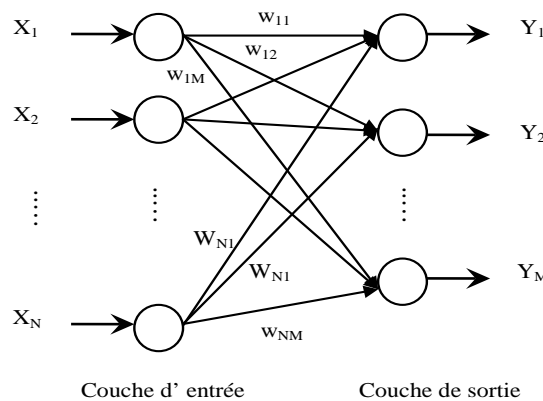


Figure 3.18: Modèle perceptron

Ce réseau utilise comme fonction d'activation la fonction seuil. Il a été exploité dans le domaine de reconnaissance des formes. Aussi ce modèle est capable de résoudre quelques opérations logiques simples (ET ou OU).

La principale limite d'un perceptron est qu'il ne peut résoudre que des problèmes linéairement séparables. Le perceptron est incapable d'apprendre des fonctions logiques telles que la fonction XOR. Il faut avoir recours aux réseaux multicouches comme solution. Dans la littérature [CirM02] [BosB06], il existe plusieurs types de RNA, on présente dans la suite le réseau de neurones MLP (Multi Layered Perceptron), appelé aussi perceptron multicouches.

C'est le plus populaire et le plus utilisé car il est capable d'approximer un très grand nombre de fonctions [HayS99].

### 3.3.3.2 Le perceptron multicouches

Ce modèle est basé sur le perceptron simple auquel on a ajouté un nombre fini de couches cachées. Chaque neurone d'une même couche  $i$  est relié à tous les neurones de la couche  $(i+1)$ , la couche  $i$  n'a de connexion qu'avec la couche  $(i+1)$ .

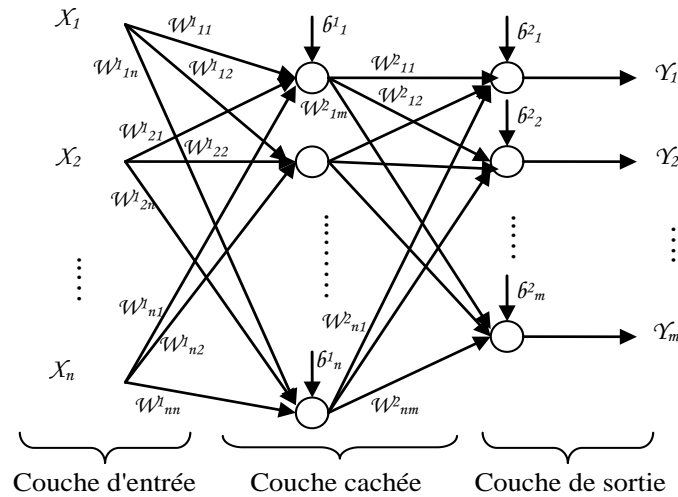


Figure 3.19. Architecture d'un réseau de neurones.

La figure 3.19 présente un exemple de réseau de neurones à trois couches. La première est celle des entrées. Certains auteurs ne la considère pas une couche neuronale car elle est linéaire et ne fait que distribuer les variables d'entrées. La deuxième est dite couche cachée. Elle constitue le cœur du réseau de neurones. La troisième est souvent dite la couche de sortie. Les fonctions d'activation des couches cachées sont de type sigmoïde. Cependant, la fonction d'activation de la couche de sortie est du type linéaire. Ce type de perceptron est très utilisé dans le domaine de l'identification et le contrôle des processus [BosB06].

L'un des problèmes rencontrés lors de l'utilisation des RN réside dans le choix de sa topologie. Comme exemple, il n'y a aucune règle générale qui détermine le nombre de neurones pour la couche cachée. Le choix est arbitraire dont on teste par la simulation sa validité. Il existe certainement d'autres variantes de réseaux de neurones [HayS99], mais elles sont peu utilisées dans la commande. Citons les réseaux de Kohonen qui sont utilisées principalement dans la classification. Les réseaux Hopfield qui nécessitent plus de temps de calcul.

### 3.3.4 Notion d'apprentissage

Parmi les propriétés fondamentales d'un réseau de neurones, est sûrement la capacité d'apprendre son environnement en vue d'améliorer sa performance lors de la conduite d'un processus. C'est la phase du développement d'un réseau de neurones durant laquelle son comportement est modifié jusqu'à l'obtention du comportement désiré. Il peut être considéré comme un problème de mise à jour des poids des connexions au sein du réseau, afin de réussir la tâche qui lui est demandée. L'apprentissage est la caractéristique principale des RNA et il

peut se faire de différentes manières et selon différentes règles. Il existe 2 grandes classes d'algorithmes d'apprentissage :

- i) L'apprentissage supervisé,
- ii) L'apprentissage non supervisé.

### 3.3.4.1 L'apprentissage supervisé

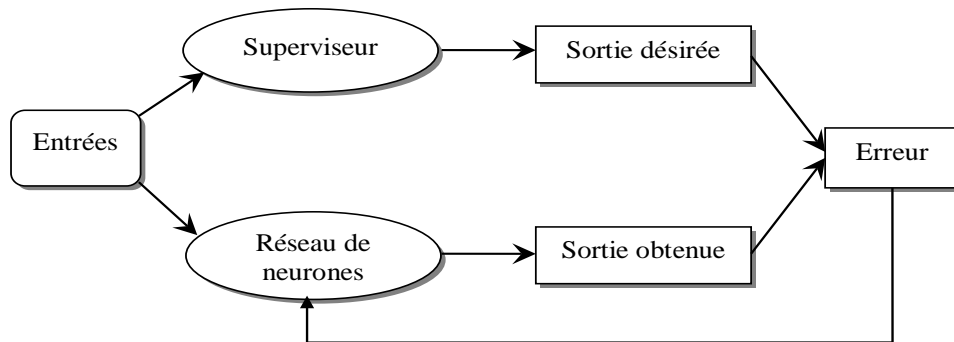


Figure 3.20. Apprentissage supervisé.

L'apprentissage supervisé est illustré par la figure 3.20. Le réseau de neurones s'adapte par comparaison entre le résultat calculé en fonction des entrées fournies et la réponse attendue en sortie. Ainsi, le réseau va se mettre à jour jusqu'à ce qu'il trouve la bonne sortie correspondante à une entrée donnée.

### 3.3.4.2 L'apprentissage non supervisé

Dans ce type d'apprentissage non supervisé donné par la figure 3.21, on ne dispose d'aucune information préalable d'appréciation du comportement, le réseau doit donc apprendre tout seul sans une intervention externe. On dispose dans ce cas d'un environnement qui fournit des stimuli. En réagissant à ces stimuli, l'algorithme d'apprentissage va réajuster les paramètres du réseau jusqu'à la stabilisation.

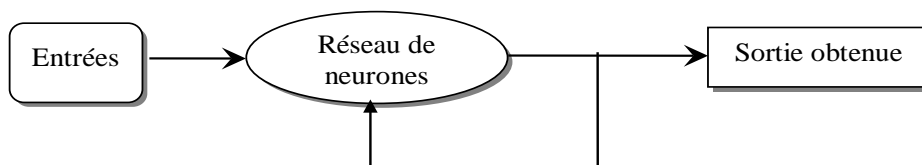


Figure 3.21. Apprentissage non supervisé.

Ce type d'apprentissage est basé sur des probabilités. On présente une entrée au réseau et on la laisse évoluer librement jusqu'à ce qu'elle se stabilise.

## 3.3.5 Conception d'un réseau de neurones

La procédure de conception d'un réseau de neurones peut être organisée en quatre étapes essentielles :

### 3.3.5.1 Choix et préparation des échantillons

La conception d'un réseau de neurones commence toujours la préparation des échantillons de données. Cette étape est décisive et va aider le concepteur à choisir le type de réseau de

neurones le plus adéquat pour résoudre son problème. Ces échantillons vont aider à définir le type de réseau de neurones, le nombre de neurones d'entrée, le nombre de neurones de sortie, l'algorithme d'apprentissage et la façon de mener la phase de tests et de validation.

### 3.3.5.2 Conception de la structure du réseau de neurones

La structure du réseau de neurones dépend souvent du type des échantillons. Tout d'abord, il faut choisir un type de réseau : un perceptron standard, un perceptron multicouche, un réseau de Hopfield, un réseau de Kohonen, le nombre de neurones, etc. Il faut tester plusieurs possibilités et choisir la topologie qui offre les meilleurs résultats.

### 3.3.5.3 Apprentissage

Une fois l'architecture d'un réseau de neurones choisie, il est essentiel de réaliser un apprentissage afin de calculer les poids permettant au réseau de neurones d'être aussi proche que possible de l'objectif visé. Cet apprentissage s'effectue avec un algorithme d'apprentissage.

### 3.3.5.4 Validation et Tests

Une fois le réseau est entraîné, il faut toujours procéder à des tests afin de vérifier que le réseau obtenu réagit correctement.

### 3.3.6 L'algorithme d'apprentissage de RNA

L'algorithme d'apprentissage le plus répandu est celui de la rétropropagation de l'erreur qui a été publié par Rumelhart en 1986 [CirM02]. Cet algorithme consiste à corriger les poids du réseau de neurones à chaque fois qu'un exemple est présenté. Cette correction est faite de telle sorte à minimiser l'erreur entre la sortie désirée et la réponse du réseau obtenue [HayS99]. Ceci est réalisé grâce à la méthode de descente de gradient. Le principe de la rétropropagation est illustré par la figure 3.22.

A chaque itération un exemple est présenté au réseau, une sortie réelle est calculée en propageant le calcul d'une couche à une autre jusqu'à la couche de sortie. C'est la phase de propagation du réseau. Cette phase se termine par le calcul de la somme quadratique des erreurs données par l'équation (3.6).

$$E(k) = \frac{1}{N} \sum_{i=1}^N (d_i(k) - y_i(k))^2 \quad (3.6)$$

avec  $d_i$  est la sortie désirée,  $y_i$  la sortie calculée par le réseau qui correspond à l'entrée  $x_i$ ,  $k$  est le nombre d'itération et  $N$  est le nombre d'exemple de la base d'apprentissage.

Cette erreur est ensuite rétropropagée dans le réseau en partant de la couche de sortie et en arrivant à la couche d'entrée tout en passant par les couches cachées. Durant cette phase, les poids de chaque neurone seront modifiés suivant la formule de l'équation (3.7) [GoeA06] [HayS99].

$$w_{ji}(k+1) = w_{ji}(k) - \eta \frac{\partial E(k)}{\partial w_{ji}(k)} \quad (3.7)$$

Avec  $w_{ji}$  est le poids de connexion du  $j$  éme neurone d'une couche avec le  $i$  éme neurone de la couche précédente. Le paramètre  $\eta$  est la constante d'apprentissage.

L'initialisation des paramètres de l'algorithme peut avoir une grande influence sur la convergence de l'algorithme. Pour des valeurs faibles de la constante  $\eta$  d'apprentissage, la convergence risque d'être lente. Dans le cas où  $\eta$  est trop élevée, ceci peut causer des oscillations et une instabilité du réseau. Dans l'équation (3.8) un terme supplémentaire appelé moment est ajouté afin d'éviter les oscillations de l'algorithme. Les poids sont corrigés et le processus continu jusqu'à ce que les variations des poids sont négligeables.

$$w_{ji}(k+1) = w_{ji}(k) - \eta \frac{\partial E(k)}{\partial w_{ji}(k)} + \alpha \Delta w_{ji}(k) \quad (3.8)$$

Pour obtenir des bons résultats pendant la phase d'apprentissage, il faudrait idéalement que les éléments de l'ensemble des exemples soient les plus représentatifs possibles de la fonction que l'on désire approximer [TOS08]. Une fois l'apprentissage est effectué, le réseau peut être utilisé pour la tâche prévue.

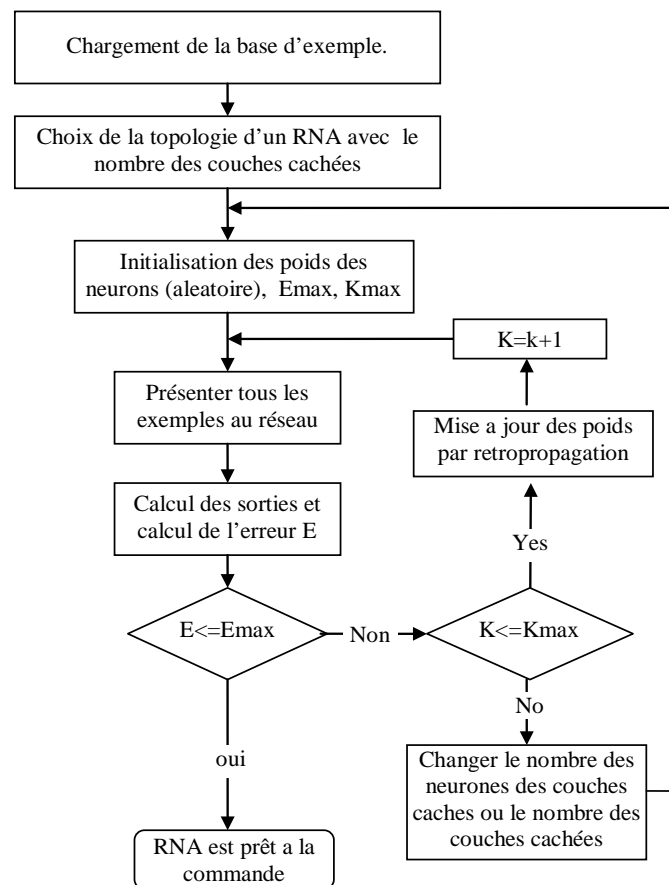


Figure 3.22. Algorithme d'apprentissage par la méthode de rétropropagation de l'erreur.

Le résultat de l'apprentissage dépend de plusieurs facteurs [CirM02] [HayS99] [BiB06] à savoir :

- i) Architecture du réseau,

- ii) Valeur des poids initiaux  $w(0)$ ,
- iii) La valeur de la constante  $\eta$  d'apprentissage.

### 3.3.7 Application des RNA pour la commande DTC d'une MAS

Pour surmonter en partie les inconvénients de la DTC conventionnelle, on propose dans ce paragraphe la commande directe de couple basé sur un réseau neuronal multicouches (DTNC : Direct Torque Neural Control). On a choisi l'architecture perceptron multicouches avec l'algorithme d'apprentissage de rétropropagation de gradient. La structure se montre alors capable de résoudre la problématique liée à la commande de la machine asynchrone [SoGd10a]. La figure 3.23 montre le schéma de principe de la commande DTNC. L'architecture comprend un réseau de neurones multicouches permettant de remplacer les deux comparateurs à hystérésis et la table de sélection.

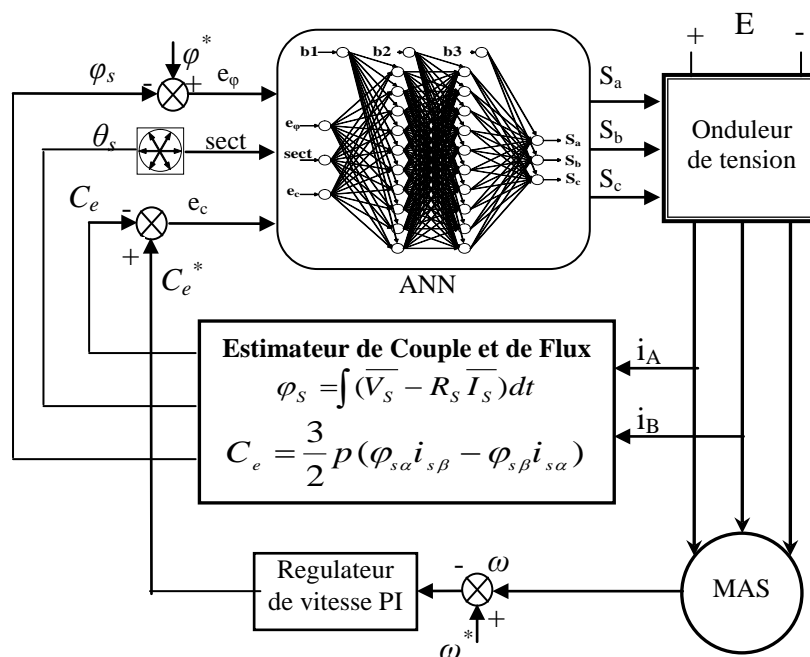


Figure 3.23. Schéma de principe de la commande DTNC de la MAS.

Ce réseau de neurones est composé d'une couche d'entrée, deux couches cachées et une couche de sortie. La couche d'entrée est composée de trois neurones, désignés respectivement par l'erreur de couple, l'erreur de flux et l'angle de flux. Les deux couches cachées sont constituées chacune de dix neurones. La couche de sortie se compose de trois neurones qui produisent la tension de référence à appliquer aux bornes de la machine asynchrone à travers l'onduleur de tension. Une fonction d'activation de type linéaire sur les couches d'entrée et de sortie et une autre fonction d'activation non linéaire de type tangentes hyperboliques est utilisée pour les couches cachées. La structure du réseau de neurones utilisée dans notre cas est donnée par la figure 3.24.



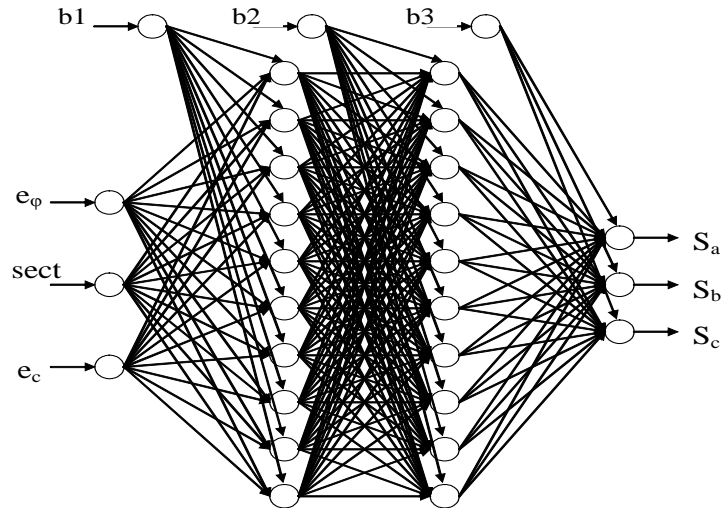


Figure 3.24. Structure du réseau neuronal utilisé pour la commande

Une fois la structure est définie, l'apprentissage du réseau neurone est une étape obligatoire durant laquelle les poids synaptiques de différentes liaisons entre les neurones du réseau seront initialisés d'une façon aléatoire et vont être corrigés à l'aide de l'algorithme de rétropropagation de gradient. On dispose ainsi de trois matrices de poids  $w_1$ ,  $w_2$  et  $w_3$  et de trois vecteurs de biais  $b_1$ ,  $b_2$  et  $b_3$  avec des valeurs initiaux.

La phase d'apprentissage du réseau va être réalisée à l'aide de la table de sélection présentée par le tableau 2.1. L'ensemble d'apprentissage comprend une matrice  $X$  contenant 36 vecteurs représentant les différentes combinaisons possibles des données d'entrées  $E_\phi$ ,  $E_c$  et  $N$  formant la table de sélection et la matrice  $Y$  des sorties désirées formées de 36 vecteurs correspondants chacun à la sortie associée au vecteur d'entrée que le réseau doit délivrer.

Les sorties de la première couche sont initialisées pour chaque exemple de la base d'apprentissage avec le vecteur d'entrée correspondant. Pour les autres couches jusqu'à la fin du réseau, la sortie de chaque neurone est calculée en fonction des valeurs de neurones de la couche précédente.

La valeur correspondante à la sortie de la dernière couche, est utilisée pour ajuster les poids de manière à minimiser la somme des erreurs quadratiques entre les composantes de tension de sortie du réseau et ceux désirées représentant la base d'apprentissage en utilisant la rétropropagation. Après avoir appris tous les exemples de la base d'apprentissage, le réseau adapté doit être capable de réaliser la commande DTNC de la machine asynchrone.

### 3.3.8 Résultat de simulation

Les erreurs de couple, de flux et de l'angle du flux statorique vont former la couche d'entrée du réseau. L'algorithme d'apprentissage est appliqué avant le démarrage de la machine. On a fixé le nombre d'itération d'apprentissage  $K_{max}$  à 3000 et l'erreur  $E_{max}$  à  $10^{-3}$ . Le coefficient d'apprentissage  $\eta$  est égale 0.75 et le coefficient de moment  $\alpha$  est égale 0.8. Les valeurs de consignes de couple et de flux statorique sont respectivement 10 N.m et 0.91 Wb.

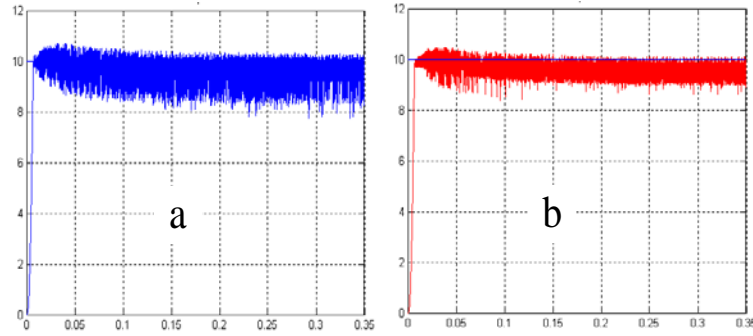


Figure 3.25. Couple électromagnétique. (a). DTC conventionnelle. (b). DTC Neuronale.

Le résultat de simulation de la figure 3.25 montre que le régime permanent est caractérisé par une réduction remarquable des ondulations de couple de la commande directe neuronale de couple. La figure 3.26 illustre une réduction d'ondulation de flux statorique de la commande directe neuronale de couple.

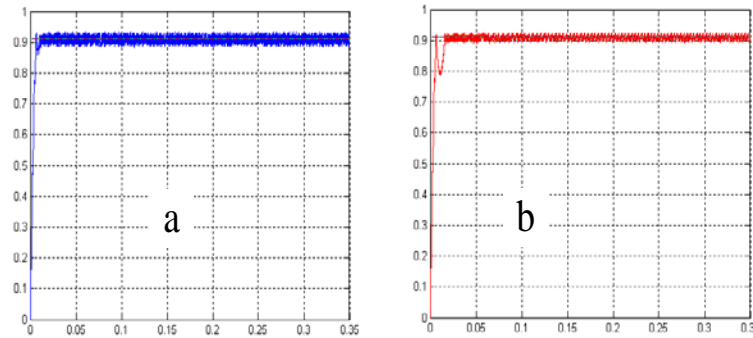


Figure 3.26. Amplitude de flux statorique. (a). DTC conventionnelle. (b). DTC Neuronale.

Les mêmes constatations sont enregistrées pour le courant statorique illustré par la figure 3.27. On remarque bien l'apport de la commande neuronale au niveau de la réduction des ondulations. La figure 3.28 illustre la trajectoire de l'amplitude du vecteur flux statorique dans le plan  $(\alpha, \beta)$ . On peut observer que l'épaisseur du cercle qui correspond à la commande neuronale présente moins d'ondulation que celle de la commande conventionnelle. Ceci est dû à la réduction des ondulations du flux.

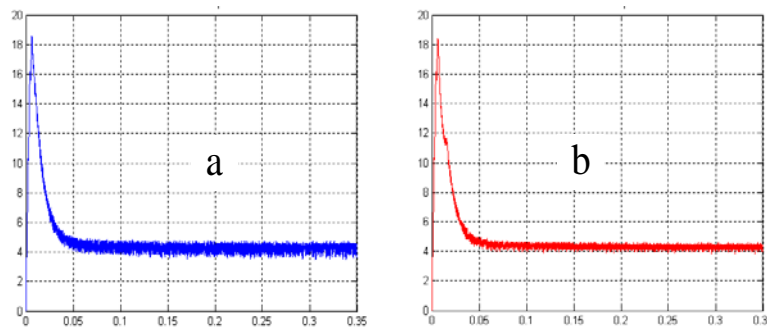


Figure 3.27. Amplitude du courant statorique. (a). DTC conventionnelle. (b). DTC Neuronale.

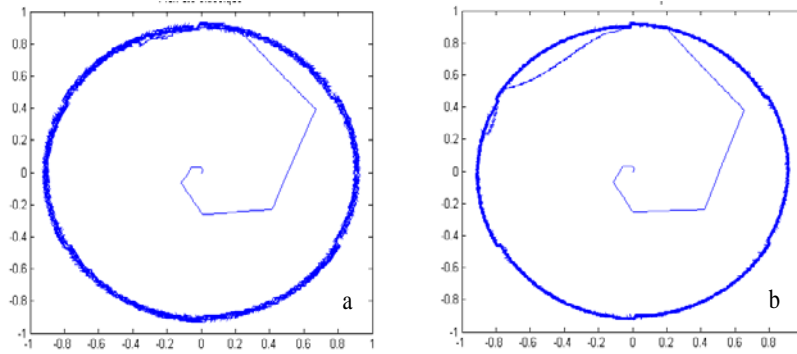


Figure 3.28. Amplitude du flux statorique dans le plan  $(\alpha, \beta)$ . (a).DTC conventionnelle. (b). DTC Neuronale.

Les figures 3.29.a et 3.29.b illustrent l'état de commutation de l'interrupteur "Sa". On peut remarquer que l'onduleur possède des phases de mise en veille des interrupteurs, ce qui permet de maîtriser la fréquence de commutation et d'avoir une fréquence en moyenne autour de 7 kHz. Cette fréquence de commutation obtenue sous la Commande Directe Neuronale de Couple est inférieure à celle de la DTC.

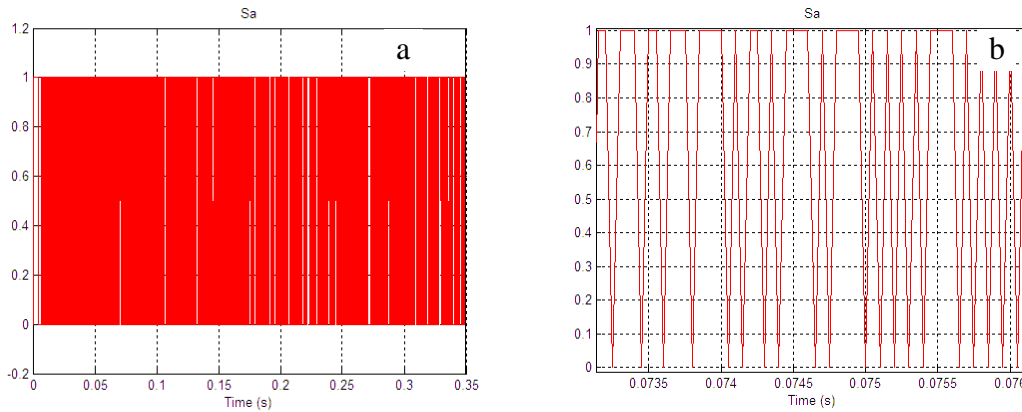


Figure 3.29. Commutation de l'interrupteur Sa. (a).DTC Neuronale. (b). DTC Neuronale (zoom).

Finalement, on peut déduire que la commande neuronale est plus performante que la commande DTC conventionnelle. Le réseau de neurones utilisé permet de surmonter en partie les inconvénients liés à la commande conventionnelle tout en réservant de bonne dynamique du couple et du flux statorique.

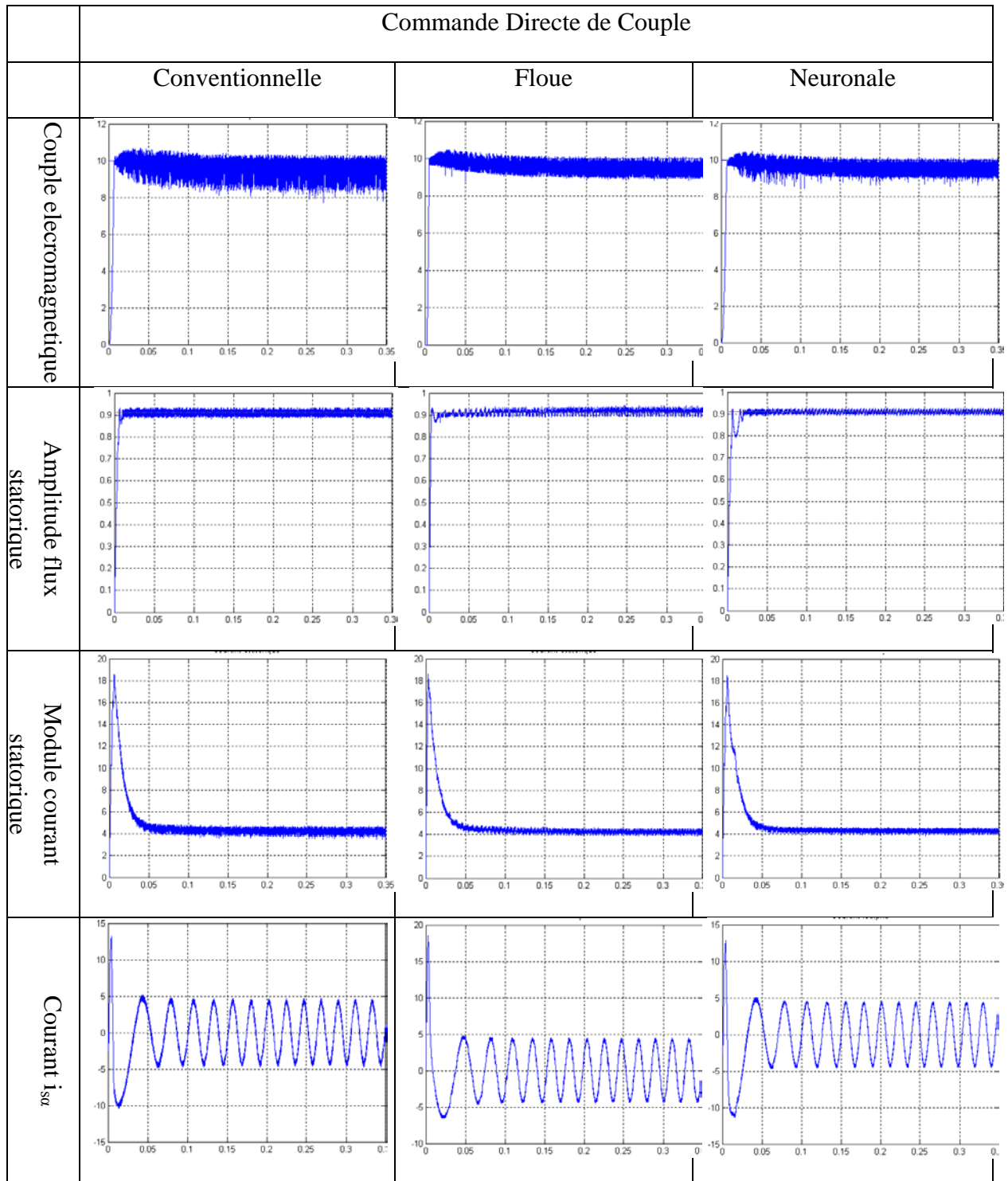
### 3.3.9 Conclusion

Dans le paragraphe précédent, on a exposé la Commande Directe Neuronale de Couple en faisant appel à un réseau de neurones pour remplacer les deux comparateurs à hystérésis et la table de sélection. Le réseau de neurone utilisé pendant la phase d'apprentissage et la phase de commande est un perceptron multicouches avec fonction d'activation sigmoïdale. Il faut signaler que la commande DTNC assure une bonne réponse dynamique de couple et de flux avec aussi une réduction des ondulations.

## 3.4 Etude comparative

Pour comparer les performances dynamiques et la robustesse de la commande des machines asynchrones à base des techniques intelligentes, on a présenté les différents résultats de

simulation de la commande DTC conventionnelle, la commande Directe Floue de Couple et la commande Directe Neuronale de Couple dans la figure 3.30. Le point de fonctionnement considéré est celui du régime nominal.



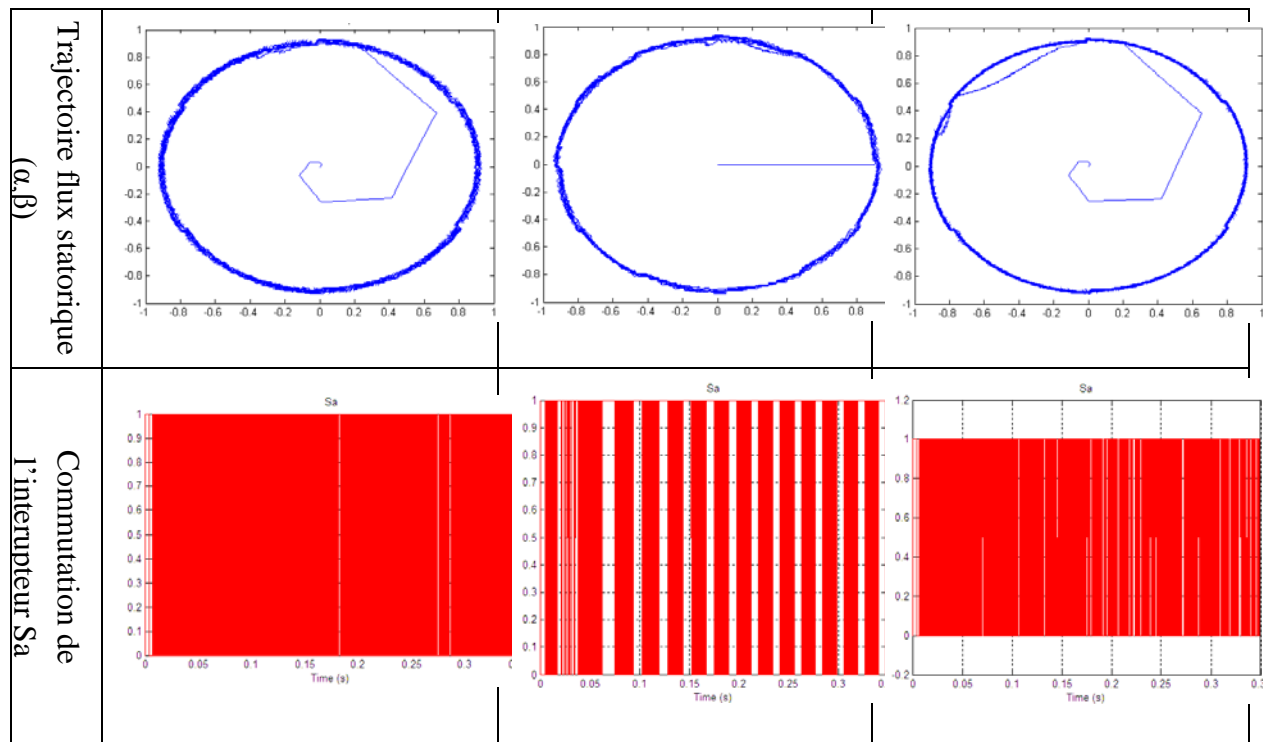


Figure 3.30. Résultats de simulations de la commande DTC et la DTC intelligente.

Les résultats de simulation de la commande floue assurent de meilleures performances par rapport aux autres techniques. Il est intéressant de remarquer que la commande floue se distingue par une importante atténuation des ondulations de couple, courant et flux. Ceci est confirmé par les résultats de comparaison des écarts des ondulations de couple, de flux et de courant illustrés dans le tableau 3.3.

Tableau 3.3. Résultats de comparaison des écarts des ondulations

	Commande Directe de Couple					
	Conventionnelle		Floue		Neuronale	
	Max-min	%	Max-min	%	Max-min	%
Couple électromagnétique (N.m)	2,705	27,05	1,332	<b><u>13,32</u></b>	1,832	18,32
Amplitude flux statorique (Wb)	0,0541	5,94	0,0539	5,92	0,034	<b><u>3,73</u></b>
Module courant statorique (A)	1,335	33,37	0,8232	<b><u>20,58</u></b>	0,8504	21,26
Fréquence de commutation de l'Interrupteur Sa (KHz)	Fréquence variable de moyenne 8KHz		<b><u>Fréquence presque constante 5KHz</u></b>		Fréquence presque constante 7KHz	

La figure 3.31 présente le spectre de fréquence du fondamental et des harmoniques pour les signaux couple électromagnétique, le flux statorique et le courant statorique. Ceci est limité pour les deux techniques de commande : Commande DTC conventionnelle et Commande Direct Floue de Couple. On observe bien une amélioration des performances apportées par la commande floue en termes de réduction des harmoniques.

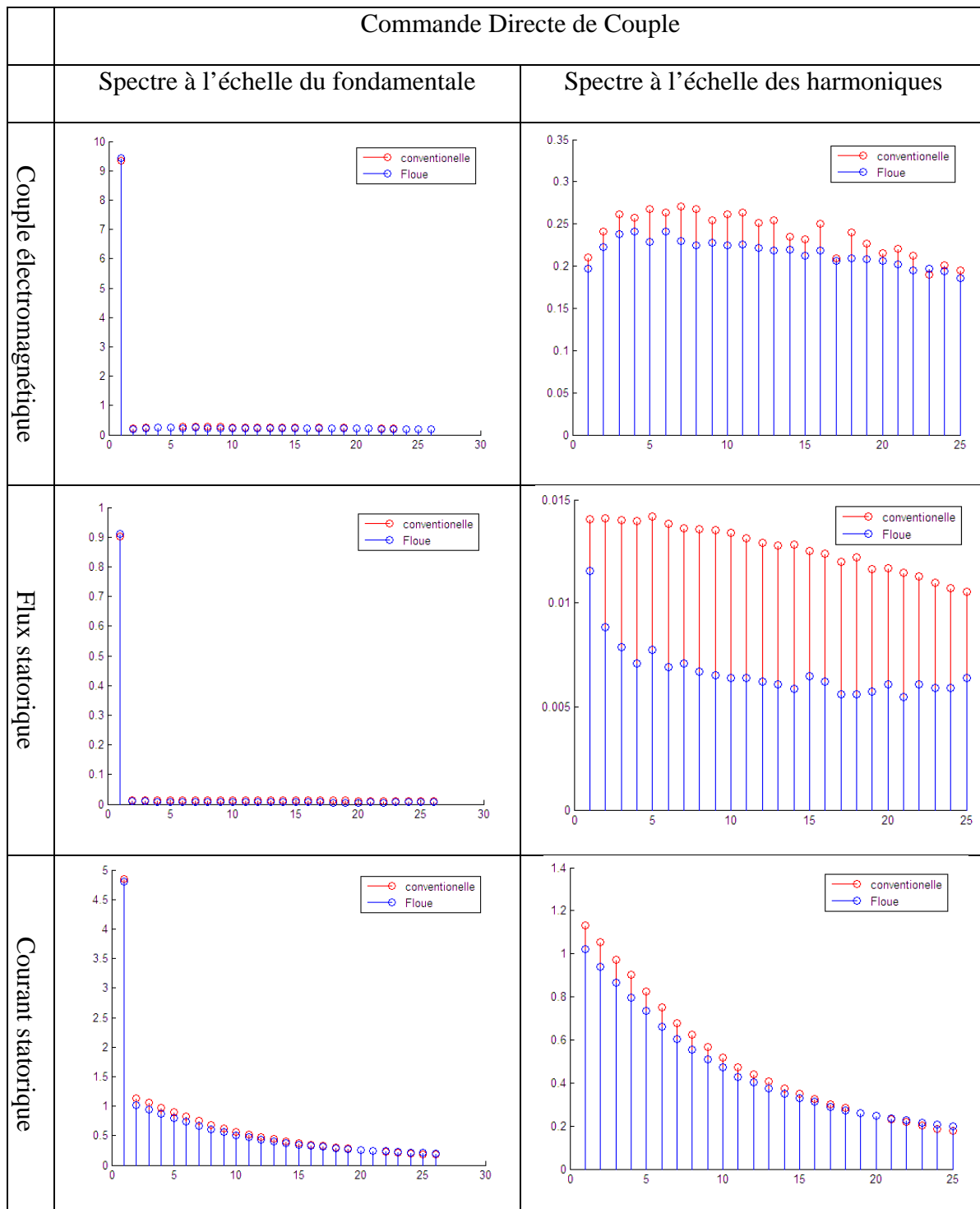


Figure 3.31. Spectre de fréquence.

À partir de ces résultats illustrés par les figures 3.30 et 3.31 et le tableau 3.3, on peut constater que les performances de la machine asynchrone, contrôlé par un contrôleur neuronal est non satisfaisante. Ceci est dû au fait qu'il n'y a pas une règle générale pour choisir les paramètres du réseau de neurones. Il est généralement difficile de déterminer ces choix seulement à partir des essais. En revanche, le contrôle par logique floue a montré de bonnes performances au niveau des ondulations de couple, de flux et de courant, ainsi qu'au niveau de la fréquence de commutation des interrupteurs des bras de l'onduleur qui se trouve assez contrôlée.

### 3.5 Conclusion

Dans ce chapitre, on a exposé l'ensemble des techniques intelligentes (la logique floue et les réseaux de neurones) afin d'améliorer certaines performances de la commande DTC de la machine asynchrone. Les résultats de simulation des commandes intelligentes montrent bien des améliorations remarquables au niveau des ondulations de couple et de flux. Ces améliorations portent sur une diminution importante des ondulations ainsi qu'une rapidité au niveau de la dynamique de couple et de flux pendant la phase de démarrage de la machine. Comme on peut remarquer qu'à partir des résultats de simulations, la commande floue a apporté plus d'amélioration à la commande DTC de la MAS que celle de la commande neuronale. Cela est confirmé par la diminution des harmoniques des signaux de couple électromagnétique, de flux statorique et du courant statorique. La diminution des pertes de commutation des interrupteurs est la conséquence de la fréquence de commutation qui se trouve assez raisonnable et reste presque constante autour de 5 kHz.

Le chapitre suivant est consacré au développement des architectures matérielles pour l'implémentation des différentes commandes de la machine asynchrone sur une plate forme à base d'un FPGA.

## **Chapitre 4 : Implémentation de la commande DTC intelligente de la MAS sur FPGA**



## 4.1 Introduction

La première partie de ce chapitre traite l'environnement et la méthodologie de conception d'une architecture matérielle pour implémenter la commande DTC d'une machine asynchrone sur cible FPGA. La méthodologie de conception s'appuie sur le concept d'adéquation algorithme architecture (AAA), qui permet de minimiser la consommation des ressources matérielles tout en respectant les contraintes temporelles de l'algorithme. On présente tout d'abord le flot de conception adopté, ensuite la modélisation et la spécification fonctionnelle de haut niveau de la chaîne de commande. A la fin de cette partie, on présente la méthodologie suivie et les outils utilisés lors de la simulation et de la synthèse pour aboutir à l'implémentation de la commande DTC d'une machine asynchrone à base d'un module matérielle pour cible FPGA. La deuxième partie concerne l'implémentation de la commande directe floue de couple de la machine asynchrone sur FPGA. Une étude détaillée sur la description matérielle sous VHDL des modules qui composent le système d'inférence flou sera présentée et discutée. Ensuite, on présente la description sous VHDL et les résultats de synthèse de la Commande Directe Neuronale de Couple de la machine asynchrone afin de l'implémenter sur une plate-forme à base d'un module matériel FPGA.

## 4.2 Étude et implémentation de l'algorithme de commande directe de couple de la machine asynchrone

### 4.2.1 Introduction

L'implémentation optimale d'un algorithme de commande d'une machine ne peut pas être élaborée par des techniques de conception classiques, mais nécessite des outils CAO et des méthodologies adoptées [MtAb00] [KeYo02] [KeHa02] [LoCh06] [ErMo07] [MoNa07]. Ces méthodologies sont capables de trouver une meilleure solution d'implémentation caractérisée par l'adéquation entre les spécificités de l'algorithme et l'architecture de commande à implémenter. D'où la problématique est de développer une méthodologie pour la conception des architectures de commande sous contraintes et avec plusieurs critères d'optimisation.

### 4.2.2 Flot et environnement de conception adoptés

Le flot de conception d'une application sur FPGA, proposée par la figure 4.1, est généralement réalisé en plusieurs phases. Il débute par une phase de modélisation et spécification au niveau système de la chaîne de commande, tout en respectant la démarche d'Adéquation Algorithme Architecture. Cette spécification subira une phase de transformation afin d'optimiser la spécification et de rendre son comportement plus performant au niveau surface ou vitesse. On décrit ensuite la description sous VHDL de tout le système pour permettre une meilleure simulation et enfin la phase de synthèse, de placement routage et d'implémentation sur FPGA. Chaque phase nécessite une vérification du bon fonctionnement global de tout le système, comme elle peut subir des modifications si nécessaire. Certaines phases peuvent être divisées en une ou plusieurs étapes de conception. Une analyse détaillée de ces phases fait l'objet des paragraphes suivants.

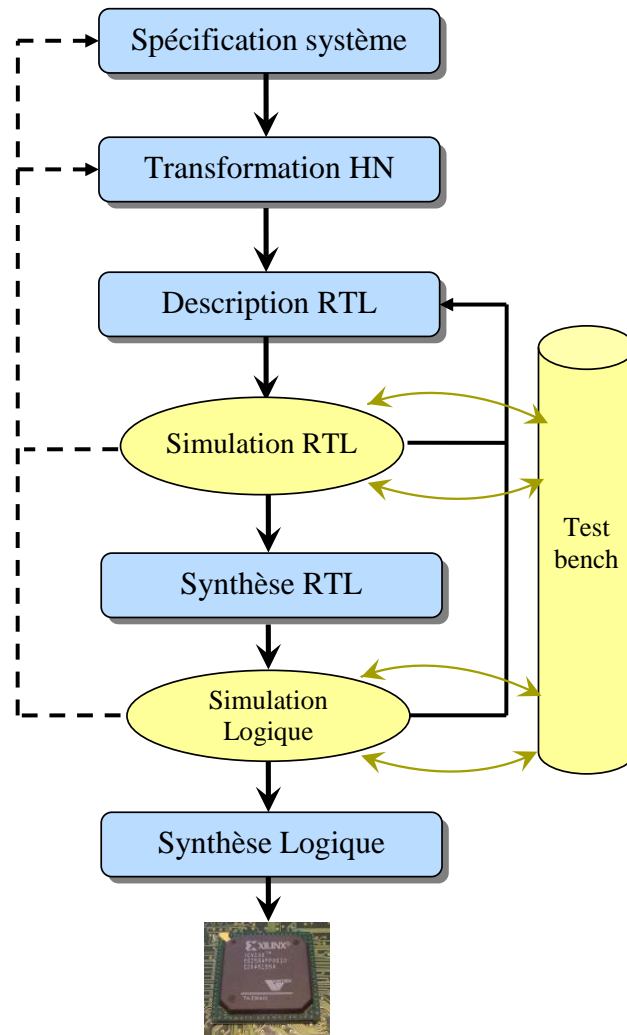


Figure 4.1. Flot de conception adopté

## 4.2.3 Modélisation et Spécification de la chaîne de commande

### 4.2.3.1 Description fonctionnelle

Cette étape est le point de départ du flot de conception. Elle débute par une phase d'analyse de l'environnement et se termine par la spécification fonctionnelle du système. L'analyse de l'environnement, appelée encore modélisation de l'environnement, s'attache à la description de toutes les entités. Il s'agit de trouver les relations et les détails nécessaires et suffisants des éléments caractéristiques (événements, données, actions ou activités) pour décrire les entités physiques situées dans l'environnement du système. Le point de vue fonctionnel doit être adopté plutôt que l'aspect physique. Le paragraphe suivant représente une description complète du système à commander en se limitant aux aspects purement fonctionnels.

### 4.2.3.2 Architecture fonctionnelle de la chaîne de commande

Pour présenter notre chaîne de commande, on a choisi de regrouper en huit blocs les fonctions principales les plus utilisées dans un système de commande d'une machine asynchrone, présentée par la figure 4.2.

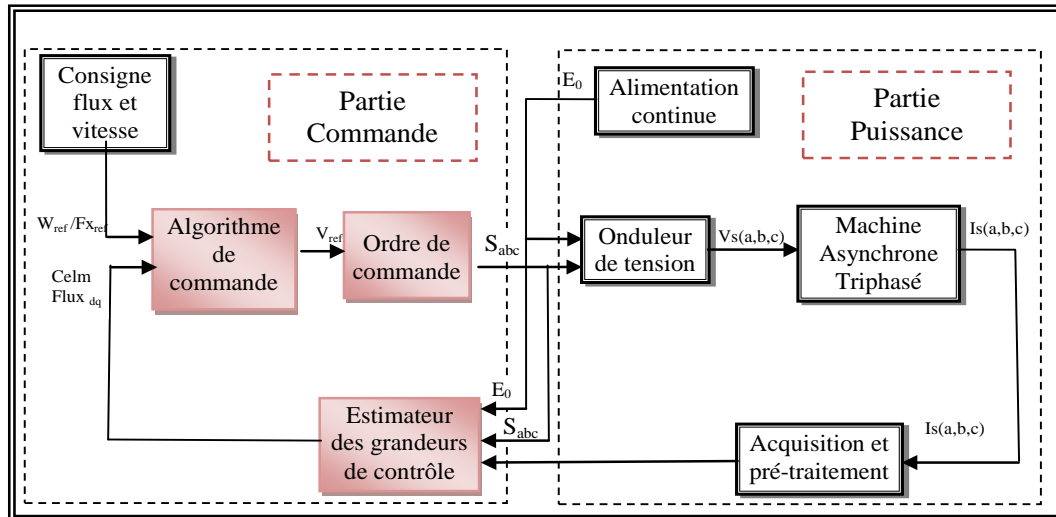


Figure 4.2. Modèle de la structure générale de la commande

La chaîne de commande présentée précédemment comporte huit blocs fonctionnels définis comme suit :

- **Algorithme de commande** : c'est le module de calcul des lois de commande adoptées pour commander la machine asynchrone en fonction des grandeurs de consigne et des grandeurs mesurées et estimées. La commande de la machine asynchrone est assurée à travers le bloc ordre de commande. Différents outils mathématiques peuvent être utilisés pour réaliser le contrôle. On peut citer l'Automatique linéaire et non linéaire, les réseaux de neurones, la logique floue, l'optimisation, etc. [RUEL02]
- **Alimentation continue** : représente le module qui permet d'alimenter en énergie électrique l'onduleur à commander. Ce module peut être un redresseur de tension ou de courant, une batterie,...
- **L'onduleur de tension** : ce module a pour tâche de transformer une tension DC constante en une tension AC triphasée de fréquence et d'amplitude variables. Ce module alimente électriquement la machine asynchrone,
- **La machine asynchrone triphasée** : représente la charge de la chaîne de commande, qui permet de transformer l'énergie électrique en énergie mécanique,
- **Acquisition et pré-traitement** : ce module regroupe les capteurs de courant, de vitesse ou de flux et la fonction de pré-traitement permettant le traitement électronique des signaux électriques issus des capteurs. On peut trouver dans ce module les capteurs, les convertisseurs analogiques numériques, l'électronique d'amplification,...
- **Estimateur des grandeurs de contrôle** : ce module reconstruit l'évolution des grandeurs nécessaire dans la commande qui ne sont pas issues directement des capteurs,
- **Ordre de commande** : la commande de l'onduleur est assurée via ce module, il permet de générer les ordres d'ouverture et de fermeture des interrupteurs de puissance qui forment l'onduleur. On peut citer par exemple la génération d'une Modulation de largeur d'Impulsion (MLI),
- **Consigne flux et vitesse** : ce module représente l'entrée de la chaîne de commande. Il permet de fixer les grandeurs de consigne pour les différentes régulations.

#### 4.2.4 Spécification de la commande directe de couple de la MA

Cette étape conduit à la transformation de manière hiérarchique des descriptions comportementales des blocs fonctionnels obtenus précédemment, en plusieurs descriptions de tailles plus petites mais surtout de complexité moindre [MtAb00]. On parle dans ce cas de description comportementale. De plus cette étape, a l'avantage d'offrir au concepteur un cheminement clair et méthodique l'aidant à construire sa propre architecture.

##### 4.2.4.1 Décomposition algorithmique

En effet, dans le cas des commandes de systèmes électriques et particulièrement de moteur asynchrone, l'algorithme de commande peut être décomposé en quatre blocs algorithmiques schématisés par la figure 4.3. Chaque bloc représentant le regroupement en catégories de sous algorithmes de la commande d'une machine asynchrone.

Les quatre blocs algorithmiques présentant la structure algorithmique de commande d'une machine asynchrone sont les suivants [KeYo02] :

- le bloc algorithmique Traitement qui regroupe les sous algorithmes, spécifiques de transformation et de conditionnement (algorithmes de Concordia, etc),
- le bloc algorithmique Estimateur qui regroupe les sous algorithmes spécifiques liés à l'estimation de grandeurs impliquées dans le processus de commande,
- le bloc algorithmique Régulation qui regroupe les sous algorithmes spécifiques de correction des grandeurs de contrôle,
- le bloc algorithmique Modulation qui regroupe les sous algorithmes spécifiques de modulation de puissance.

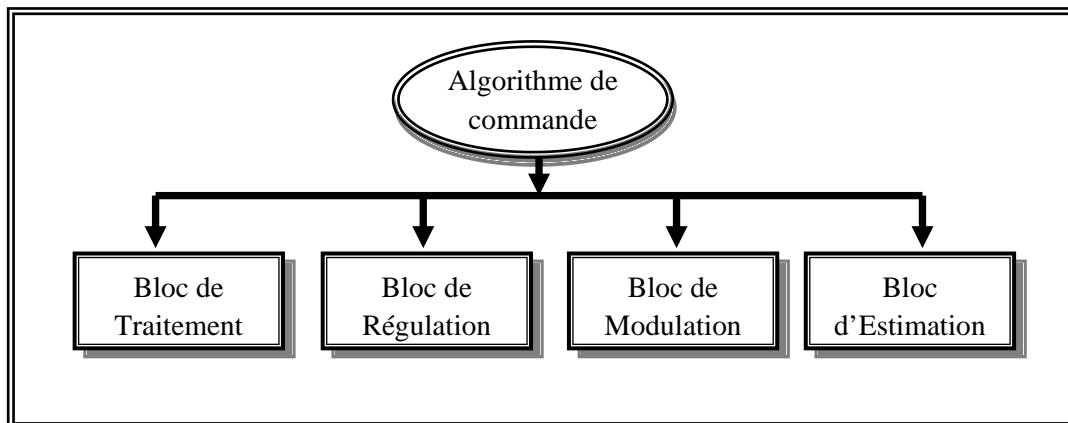


Figure 4.3. Décompositions algorithmiques

##### 4.2.4.2 Spécification fonctionnelle

Pour simplifier de plus la démarche de conception de l'implémentation, la spécification fonctionnelle consiste à décomposer les blocs issus de la phase de décomposition algorithmique définie dans la figure 4.3 en des sous blocs algorithmiques appelés fonctions de service. Le choix de la spécification fonctionnelle est réalisé suivant la connaissance du concepteur et en respectant le critère de réutilisabilité ultérieure lors d'une nouvelle phase de conception. Ceci permet, d'une part de faciliter la conception à réaliser et d'autre part de

minimiser le temps de développement. En effet, cette étape est importante lorsque les algorithmes à implémenter sont de nature complexes.

#### 4.2.4.3 Spécification architecturale

Dans cette phase, la spécification architecturale regroupe les fonctions de service déjà définies dans le paragraphe précédent. Ce regroupement est effectué selon les critères suivant :

- le critère de mixité en regroupant dans le même module les fonctions de services pouvant donner lieu à la conception et le développement d'un circuit intégré mixte,
- le critère d'homogénéité en regroupant dans le même module les fonctions de service dont la structure peut être considérée comme homogène,
- le critère de performance fonctionnelle en regroupant dans le même module les fonctions de service dont dépendent les résultats de fonctionnalité et de performance de la commande.

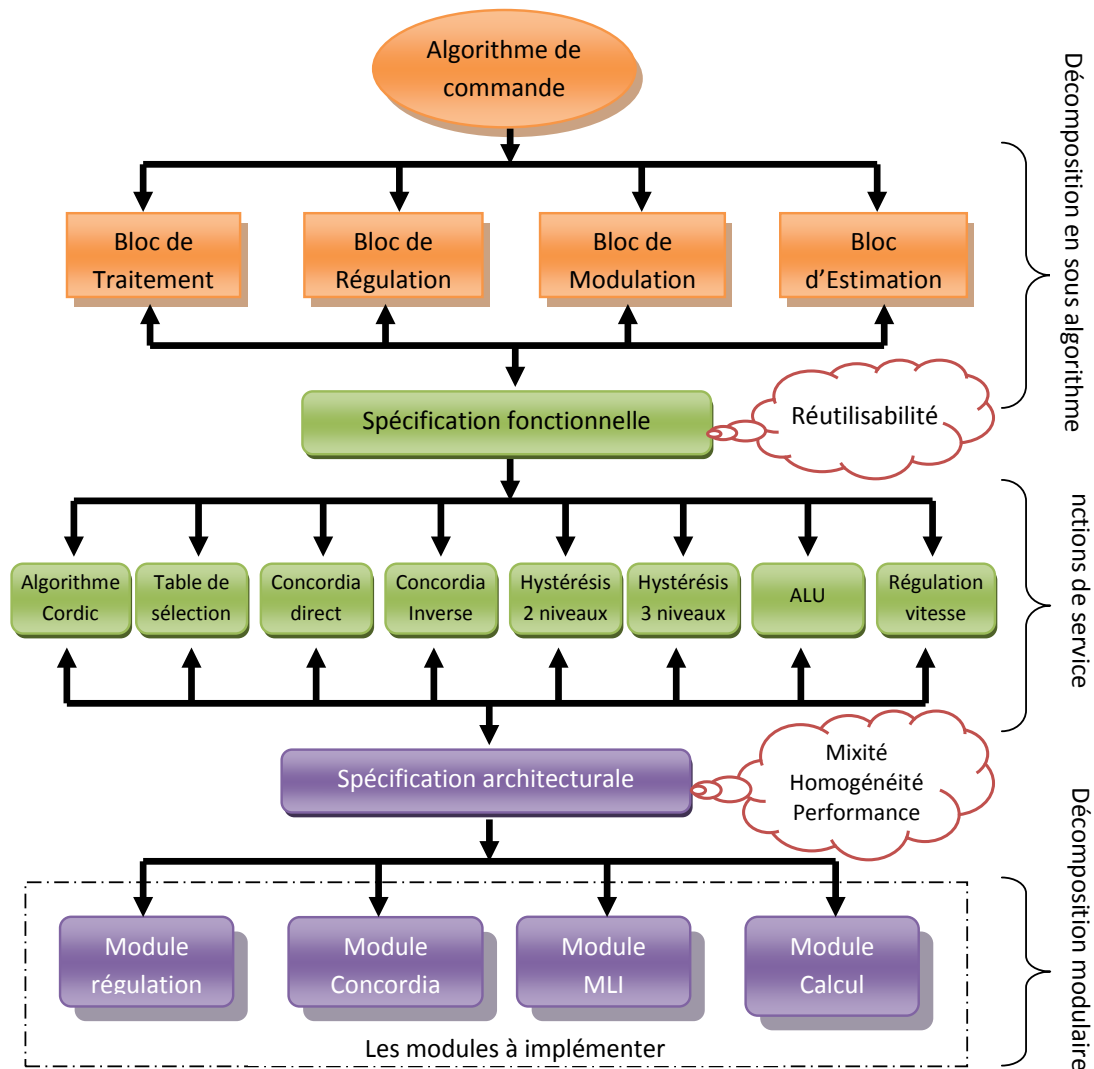


Figure 4.4. La méthodologie de décomposition et de spécification de l'algorithme de commande

La figure 4.4 présente la méthodologie de décomposition et de spécification de l'algorithme de commande d'une machine asynchrone. Enfin dans cette phase de spécification, il est

important de noter que le partage d'une fonctionnalité en modules reste délicat à réaliser. En effet, une description grossière produira des sous-blocs avec une fonctionnalité variée, donc complexe et peu réutilisable. Par contre, une description fine pourra devenir trop complexe en surchargeant le concept avec des détails inutiles [KeYo02] [MoNa07b].

#### 4.2.5 Transformations de haut niveau

Les transformations de haut niveau ont comme but d'optimiser la spécification et de rendre son comportement plus performant au niveau surface ou vitesse. Ces transformations peuvent être affectées au niveau de la spécification de départ, en faisant des modifications et des approximations afin de diminuer la complexité et d'assurer une adéquation algorithme dédiée à l'implémentation matérielle permettant d'avoir des circuits plus performants. Cette phase nécessite souvent une expertise en commande des systèmes. En fait, l'algorithme ne doit pas perdre ces performances à cause de ces modifications [AbAb03].

En outre, ces transformations peuvent être utilisées au niveau de la description RTL en cherchant des descriptions équivalentes de la spécification de départ mais plus performante au niveau surface ou vitesse. Parmi les transformations possibles, on peut faire appeler à :

- Exploiter les équivalences mathématiques,
- La régularité : mise sous forme récurrente des algorithmes,
- Les boucles : bien formées, pouvant être déroulées,
- Les procédures et les fonctions : limiter leur nombre, leurs variables locales,
- Utiliser des opérateurs logiques spécifiques : MAC (Multiplication - Accumulation),
- Une multiplication par une puissance de deux peut être transformée en un décalage par le nombre approprié de bits,
- L'accroissement du parallélisme.

Dans l'étude de ce travail on aborde l'implémentation matérielle de la commande DTC d'une machine asynchrone qui consiste à déterminer la tension de référence  $V_s$  à délivrer aux bornes de la machine asynchrone par l'onduleur de tension. Ce qui nous ramène à chercher une optimisation des opérateurs arithmétiques utilisés au sein de notre application, tels que la multiplication et la division. Cette partie est mieux développée dans la dernière partie de ce chapitre.

#### 4.2.6 Description RTL

Comme mentionné dans la deuxième partie du premier chapitre portant sur l'état de l'art, on a choisi l'environnement VHDL comme étant le langage de la modélisation comportementale et structurelle de haut niveau, de simulation et de synthèse pour la commande DTC de la machine asynchrone à implémenter, ainsi que pour les commandes intelligentes.

On a adopté lors de la phase de description la méthodologie AAA, qui permet de mettre en facteur les modules répétitifs de calcul de l'algorithme, dans le but de minimiser les ressources matérielles consommées tout en respectant les contraintes temporelles exigées à l'exécution de l'algorithme à implémenter.

#### 4.2.6.1 Elaboration des modules de la chaîne de commande

##### 4.2.6.1.1 Module onduleur

L'onduleur comme il est décrit dans le premier chapitre, a pour tâche de transformer une tension continue en une tension triphasée de fréquence et d'amplitude variables. Ce convertisseur se compose de trois bras, connecté chacun à une phase du moteur et comportant six interrupteurs de puissance. Ce convertisseur présente huit configurations possibles selon l'état de ses interrupteurs, ce qui correspond dans le repère  $\alpha$ - $\beta$  à huit vecteurs de tension statoriques appliqués aux bornes de la machine asynchrone. Les  $S_a$ ,  $S_b$  et  $S_c$  représentent les états d'ouverture ( $S_i=0$ , avec  $i=a$  ou  $b$  ou  $c$ ) ou de fermeture ( $S_i=1$ , avec  $i=a$  ou  $b$  ou  $c$ ) des interrupteurs de puissance de l'onduleur. A chaque séquence des états des interrupteurs lui correspond une tension triphasée appliquée aux bornes de la MAS.

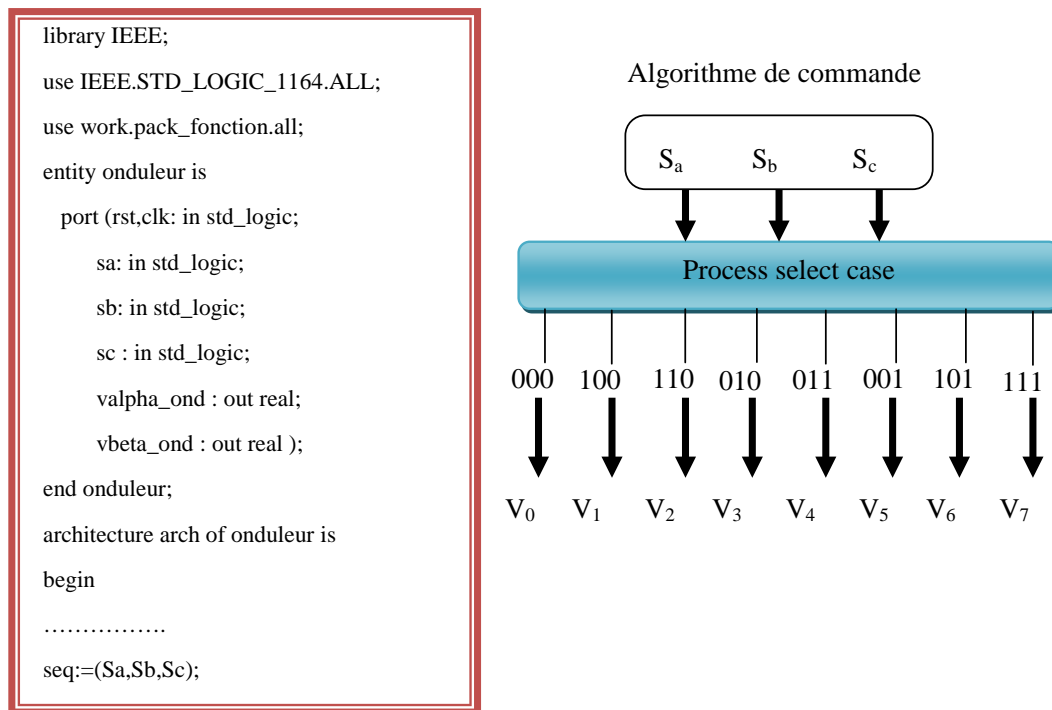


Figure 4.5. Module VHDL de l'onduleur de tension. (a). Entité de module de l'onduleur de tension. (b). Architecture de module de l'onduleur de tension

L'entité du module VHDL de l'onduleur est présentée par la figure 4.5.a. Ce module a comme entrée les états d'ouverture et de fermeture des interrupteurs de l'onduleur  $S_a$ ,  $S_b$  et  $S_c$  issue de l'algorithme de commande. La sortie du module représente les deux composantes de la tension statorique dans le repère de concordia ( $V_{\alpha\_ond}$  et  $V_{\beta\_ond}$ ). L'architecture de ce module est composée d'un Process qui permet de sélectionner un cas parmi huit en fonction de la combinaison ( $S_a S_b S_c$ ) présentée par la figure 4.5.b.

##### 4.2.6.1.2 Module Machine asynchrone

Dans ce cadre, il est indispensable dans notre travail de disposer d'un module VHDL réutilisable de ce composant. La discrétisation du modèle de la machine asynchrone décrite dans le premier chapitre est modélisé par le langage VHDL par deux *Process*.

L'entité du module VHDL de la machine asynchrone est présentée par la figure 4.6.a. Ce module a comme entrée les deux composantes de tension dans le repère de concordia  $V_{\alpha\_ond}$  et  $V_{\beta\_ond}$  issue de l'onduleur et a comme sortie cinq signaux : les deux composantes du courant dans le repère de Concordia ( $I_{\alpha}$  et  $I_{\beta}$ ), les deux composantes du flux statorique dans le repère de Concordia ( $\Phi_{\alpha}$  et  $\Phi_{\beta}$ ) et la vitesse angulaire rotorique  $\omega_r$ . L'architecture de ce module est composée de deux *Process* présentées par la figure 4.6.b.

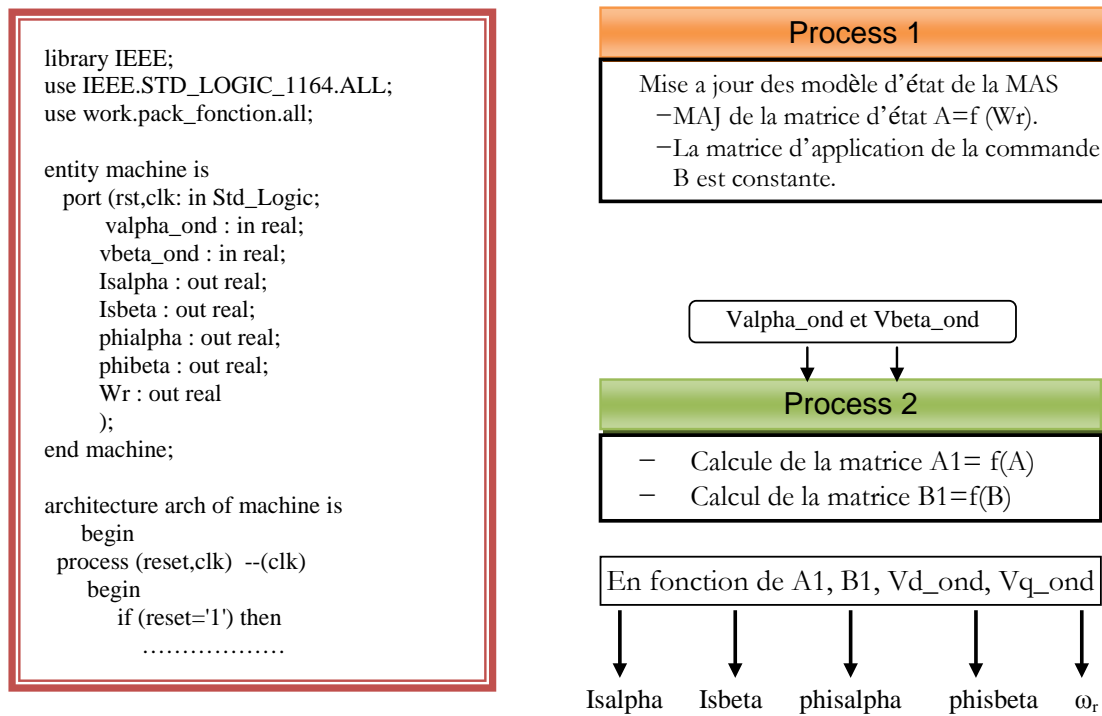


Figure 4.6. Modèle VHDL de la Machine asynchrone. (a). Entité de module de MAS. (b). Architecture de module de MAS.

#### 4.2.6.1.3 Module de l'algorithme de commande de la MAS

Dans ce module, on calcule les différentes lois de commande en fonction des grandeurs de consigne et des grandeurs mesurées ou estimées.

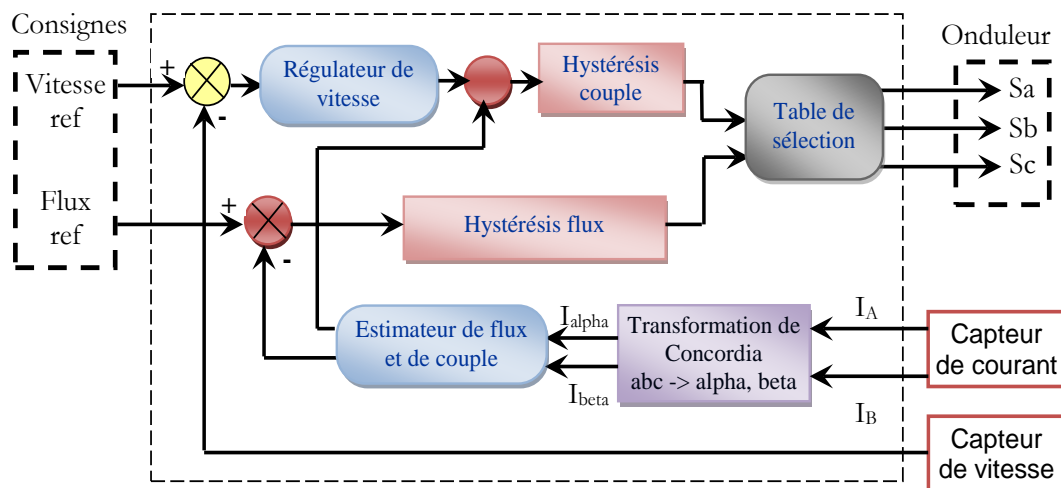


Figure 4.7. Structure algorithmique de commande DTC d'une MAS



L'algorithme de commande DTC de la machine asynchrone peut être décomposé en quatre blocs algorithmiques selon l'homogénéité de structure et de mixité du traitement. Ces regroupements se basent sur la description structurale des commandes modernes de moteur asynchrone.

Les quatre blocs algorithmiques constituant la structure algorithmique de commande de la machine asynchrone présentés par la figure 4.7 sont les suivants :

- le bloc Estimateur,
- le bloc de Traitement,
- le bloc Régulation,
- le bloc table de sélection.

Le modèle d'architecture de l'algorithme de commande DTC est présenté par la figure 4.8 [SoGd13b]. Ce modèle a comme entrées : les valeurs des courants ( $I_a$ ,  $I_b$ ) issues de la machine à travers les capteurs et les tensions ( $V_a$ ,  $V_b$ ,  $V_c$ ). Seulement deux courants de phase ( $I_a$ ,  $I_b$ ) sont mesurés. Cependant, ce modèle a comme sortie trois signaux logiques ( $S_a$ ,  $S_b$ ,  $S_c$ ) qui représentent l'état d'ouverture ou de fermeture des interrupteurs de puissance IGBT de l'onduleur durant une période fixe. A chaque combinaison de ces signaux logique lui correspond une tension parmi huit tensions illustrées par la figure 2.4 du deuxième chapitre. Pour chaque séquence de ces états lui correspond une tension triphasée appliquée aux bornes de la MAS. Ce modèle d'architecture est contrôlé par une machine à état fini (FSM).

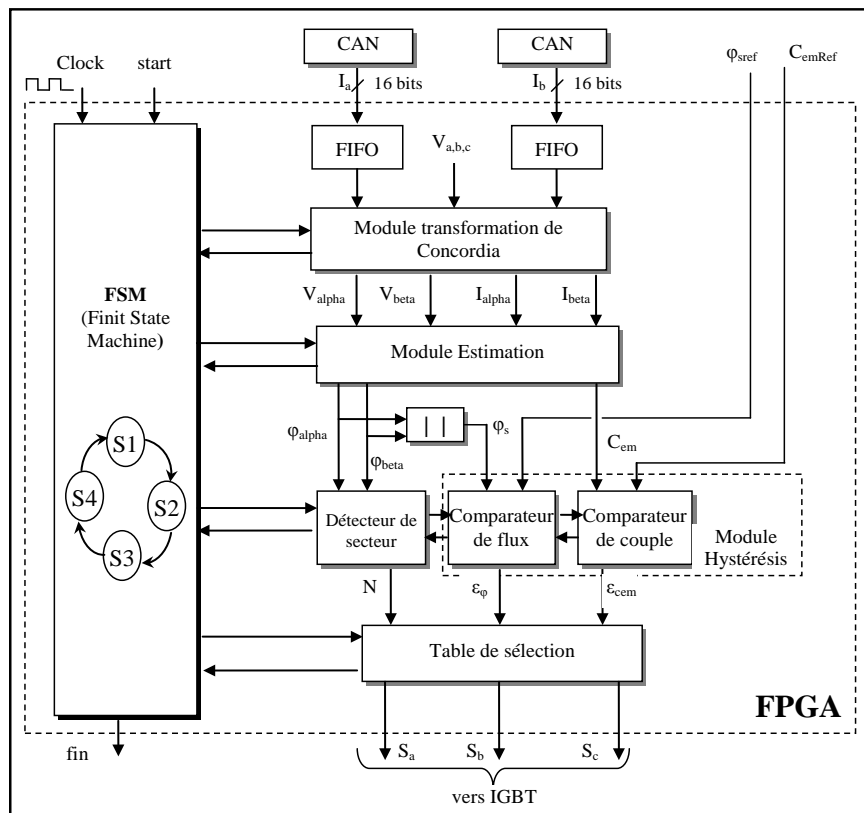


Figure 4.8. Modèle d'architecture de l'Algorithme DTC de la MAS

Cette architecture est décomposée en modules, représentés sous forme des entités VHDL qui communiquent entre eux. Chacun de ces modules est par la suite décomposé en d'autres sous-modules. Le processus de décomposition est répété jusqu'à ce que les modules obtenus soient suffisamment simples. La figure 4.9 représente l'entité du module Estimation. Ce module est décomposé en deux sous-modules (deux entités VHDL) : le séquenceur illustré par la figure 4.9.a et le chemin de données ou datapath présenté par la figure 4.9.b [SoGd13b]. En adoptant la méthodologie AAA, ce chemin de données assure un compromis entre la consommation des ressources matérielles et les contraintes temporelles exigées. Dans cette architecture, le multiplieur est en facteur, puisque il est le plus consommateur en ressource matérielles. Les opérateurs de même type ont la même couleur (les registres, et les opérateurs logiques, les multiplieurs, les additionneurs/soustracteurs). Les signaux de contrôle sont représentés en violet. En fait, le chemin de données est défini comme un réseau de registres, d'opérateurs, de multiplexeurs et de liaison [LoCh06]. Une fois le chemin de données est fixé, on élabore le séquenceur adéquat.

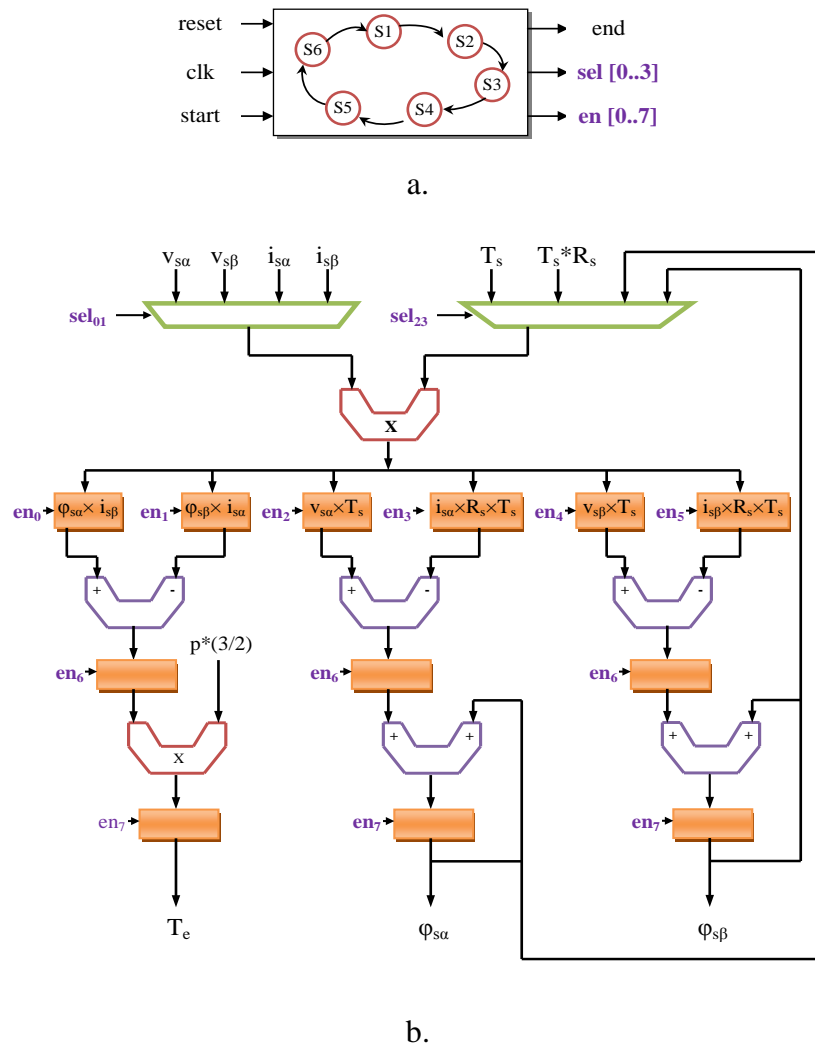


Figure 4.9. Architecture module Estimation. (a) Séquenceur de l'estimation. (b) Chemin de données

#### 4.2.7 Résultat de simulation

Le modèle de simulation de la commande DTC de la machine asynchrone a été décrit en VHDL. Ce modèle a été validé par Matlab qui a aboutit aux mêmes résultats. Une

comparaison entre la simulation du système complet sous Matlab, déjà présenté dans le deuxième chapitre et la simulation du même système sous Modelsim fait l'objet de la figure 4.10. Les mêmes consignes de couple  $C_{em} = 10 \text{ N.m}$  et de flux  $\phi_{ref} = 0.91 \text{ Wb}$  sont considérés.

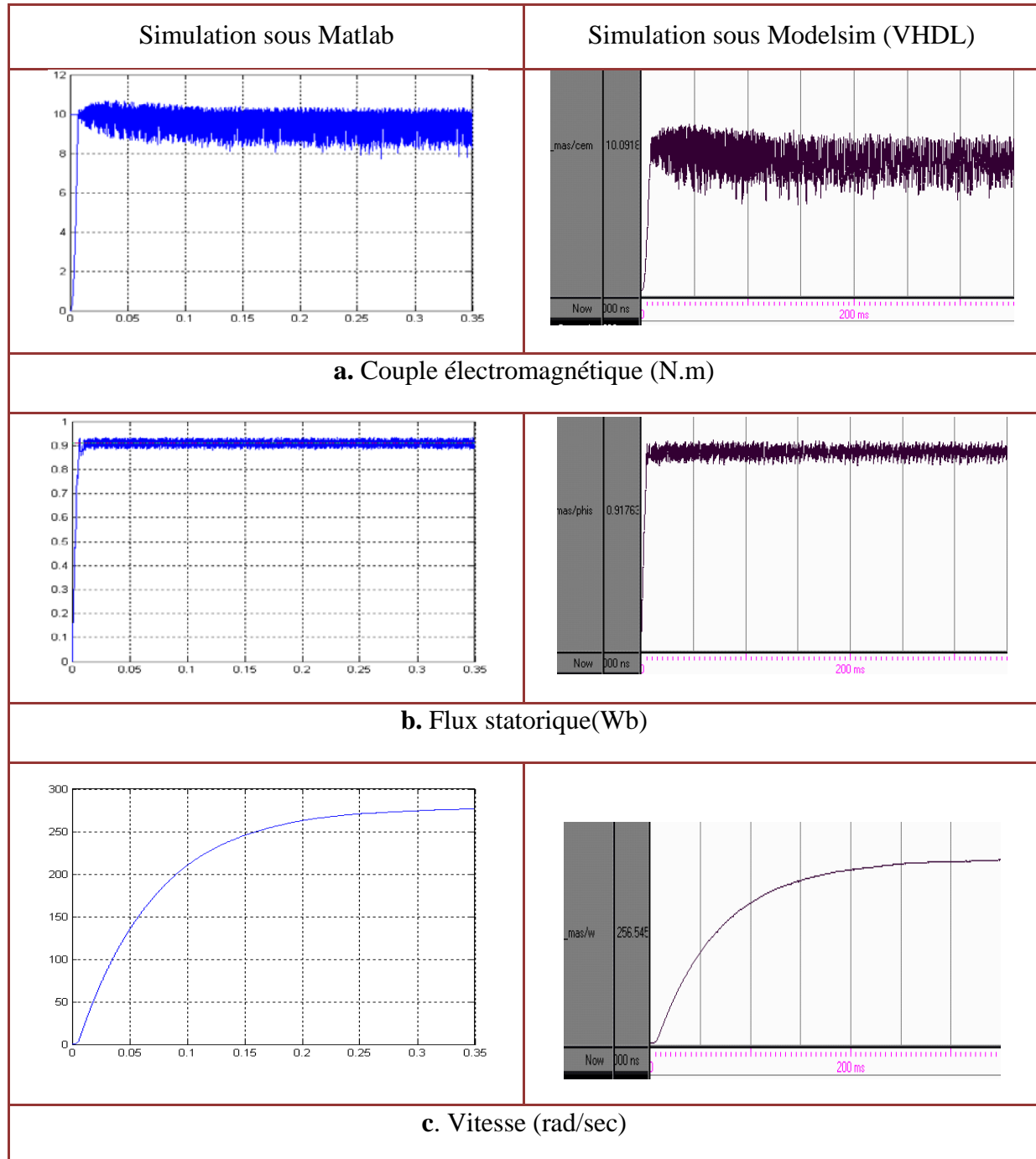


Figure 4.10. Simulation de la commande DTC de la machine asynchrone

On peut remarquer, que l'évolution temporelle de couple, de flux statorique et de la vitesse de rotation pour une durée de temps égale à 350 ms sous Modelsim, ne diffère pas des résultats obtenus par le modèle de simulation sous Matlab. On peut conclure que les réponses obtenues avec Modelsim et par Matlab montrent une bonne conformité des résultats.

### 4.2.8 La synthèse

Le synthétiseur est un compilateur particulier capable, à partir d'une description VHDL, de générer automatiquement un schéma de câblage permettant la reconfiguration du circuit cible FPGA. A travers le style d'écriture de la description synthétisable, le synthétiseur va reconnaître un certain nombre de primitives qu'il va implémenter et connecter entre elles. Ces primitives seront au minimum des portes NAND, NOR, NOT, des bascules D, des buffers d'entrées-sorties, comme elles peuvent être aussi un compteur, un multiplexeur, une RAM, un registre, un additionneur, un multiplieur ou tout autre bloc particulier [PaNo03]. C'est le style d'écriture qui va guider le synthétiseur dans ses choix d'implémentation au niveau circuit, lors d'une description VHDL en vue de synthèse. Cependant, les outils de synthèse et les technologies cibles imposent souvent certaines contraintes ou limitations qu'il faut prendre en compte pour aboutir à une description synthétisable. Parmi ces limitations au niveau RTL on cite :

- Initialisations des variables et des signaux sont ignorées ;
- Pas d'équation logique sur une horloge ;
- Restriction sur les boucles (LOOP) ;
- Pas de type REAL ;
- Que des tableaux monodimensionnels sont autorisés et ils doivent avoir un indice de type ou sous-type entier
- Pas de fonction mathématique (cosinus, sinus, racine carré, ...).

Pour répondre aux contraintes et aux limitations technologique, on présente dans ce qui suit les solutions adoptées pour notre conception :

- On introduit un comportement de type *set/reset* pour les initialisations des variables et des signaux ;
- L'algorithme de Cordic est exploité pour modéliser les fonctions mathématiques (cosinus, sinus, racine carrée) utilisées dans notre algorithme de commande ;
- Pour représenter les nombres réels, on a développé une bibliothèque d'opérateurs en virgule fixe. Une présentation détaillée de cette bibliothèque de virgule fixe est en Annexe A.

Dans les paragraphes suivants on va détailler les solutions utilisées pour aboutir à une synthèse de l'architecture de commande dédiée l'implémentation sur FPGA.

#### 4.2.8.1 Initialisation des variables et des signaux

Tout objet VHDL possède une valeur initiale, soit par défaut héritée de la définition de type ou de sous-type auquel l'objet appartient, soit explicite au moyen d'une expression spécifiée au moment de la déclaration. Aucune de ces deux méthodes n'est supportée pour la synthèse. Il s'agit ainsi de prévoir dans le modèle VHDL à synthétiser un comportement de type *set/reset* explicite.

### 4.2.8.2 Résultats de synthèse

La figure 4.11 présente le résultat de synthèse de l'algorithme de commande à implémenter sur FPGA. Le fichier VHDL synthétisé dans cette figure est le *Top Level*, qui fait appel à tous les modules qui forment l'algorithme de commande DTC. La figure 4.12 présente le résultat de synthèse de la table de sélection. L'outil de synthèse utilisé est XST de l'environnement Xilinx ISE 12.1.

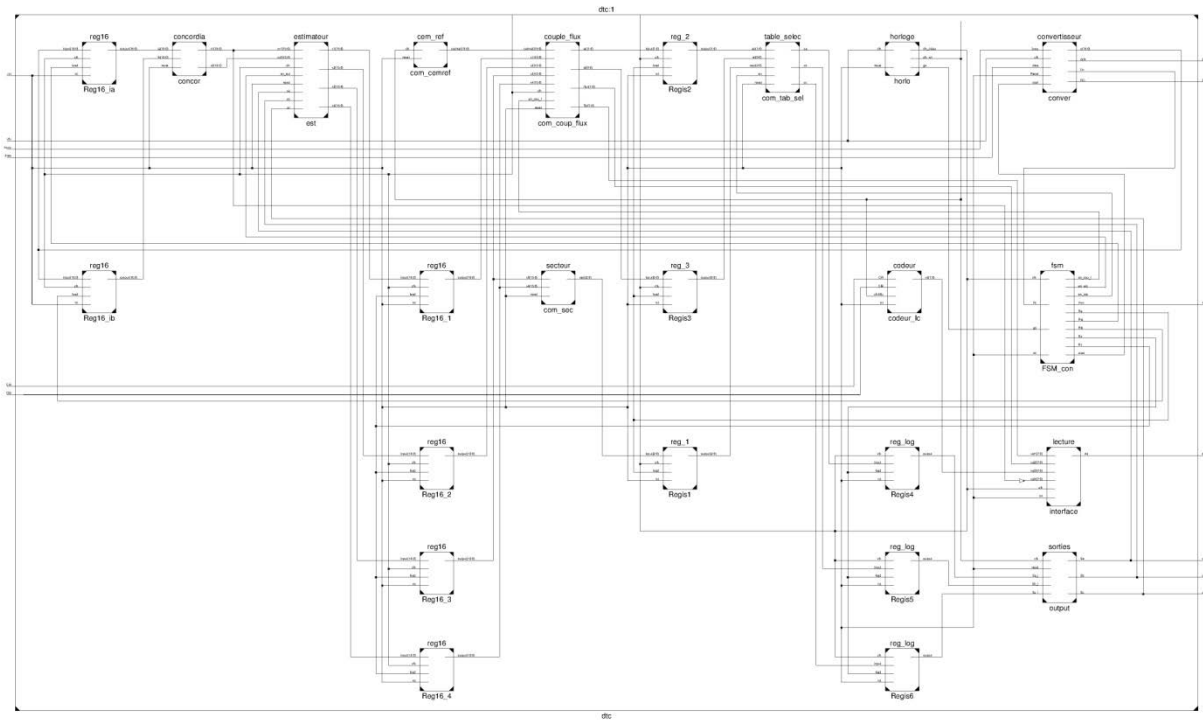


Figure 4.11. Schéma RTL de l'algorithme de commande

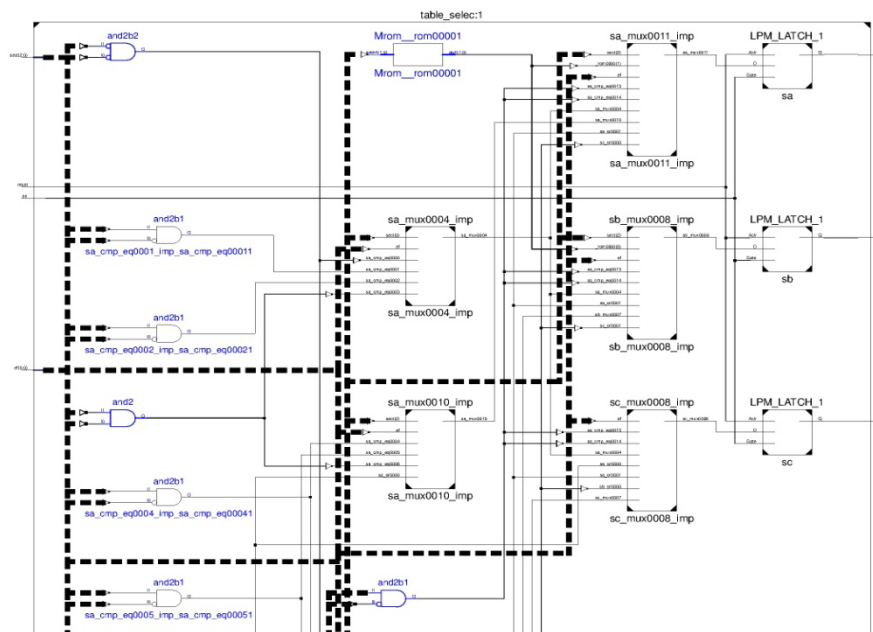


Figure 4.12. Le schéma RTL du module table de sélection

Le tableau 4.1 suivant présente les performances en termes de consommation de ressources, obtenues lors de l'implémentation de l'architecture de la commande DTC conventionnelle sur le FPGA Virtex 5 donnée par la figure 4.8. Les ressources consommées sont obtenues pour un format en virgule fixe de 16 bits.

Tableau 4.1. Ressources utilisées sur le FPGA par l'algorithme de commande.

Ressources	utilisées	Accessibles
Les pins E/S	13	640
Nombre de LUTs	1601	44800
Nombre de Slices	478	44800
Nombre des Latches	342	1737
Nombre de BUFG/BUFGCTRLs	4	32
Nombre de bloc multiplieur	16	128

La fréquence maximale de l'horloge est fixée par l'outil de synthèse égale à 34,12 MHz, qui correspond à une période minimale de 29,3 ns. La figure 4.13 présente le résultat de routage de l'architecture de la commande DTC de la machine asynchrone sur FPGA Virtex 5 développée avec l'outil *FPGA editor*.

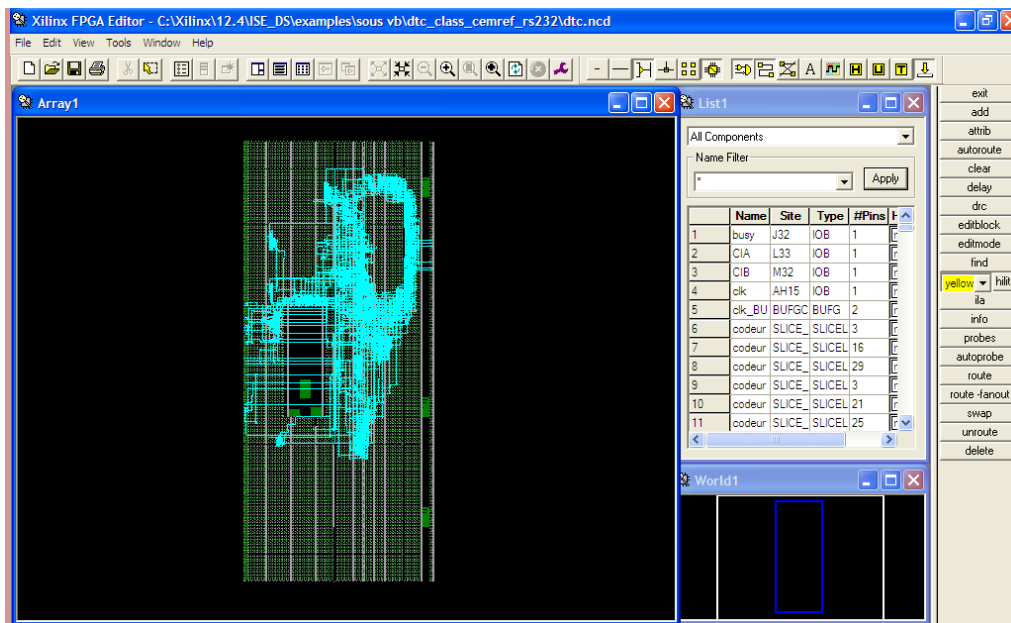


Figure 4.13 Résultat de routage de la commande DTC de la MA

#### 4.2.9 Simulation logique

L'outil de synthèse est capable de transformer une description RTL en un circuit optimisé à base de portes logiques. L'optimisation est gérée par les contraintes fournies à l'outil de synthèse sur la surface et/ou les délais que doit satisfaire le circuit. Le résultat de la synthèse est un circuit de portes logiques qui peut être décrit en VHDL. La simulation logique après la synthèse ne prend en compte que les délais des portes et une estimation des délais d'interconnexion.

On présente dans les figures 4.14 et 4.15 une comparaison entre le modèle de simulation de la commande DTC du moteur asynchrone déjà présenté par la figure 4.10, qui n'est pas

synthétisable et le même modèle de simulation mais en utilisant la représentation en virgule fixe de la bibliothèque développée ainsi que les solutions adoptées.

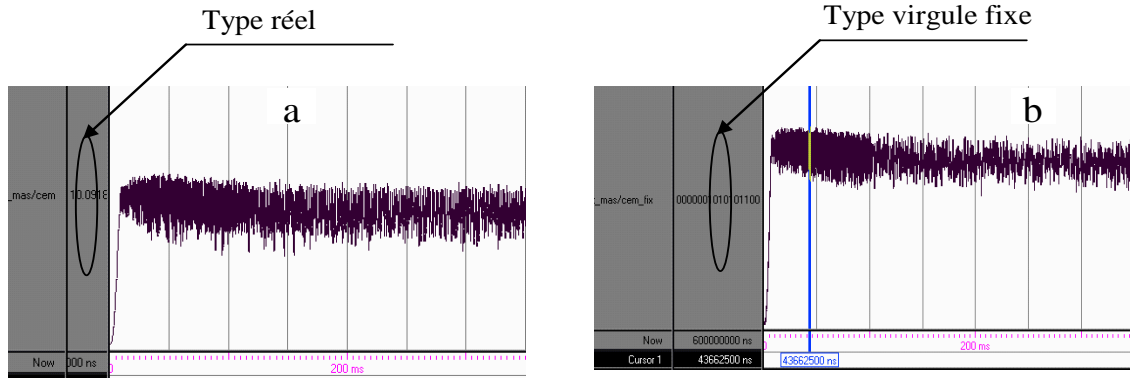


Figure 4.14. Le couple électromagnétique (N.m). (a). Le modèle de simulation non synthétisable. (b). Le modèle de simulation synthétisable utilisant la bibliothèque de virgule fixe.

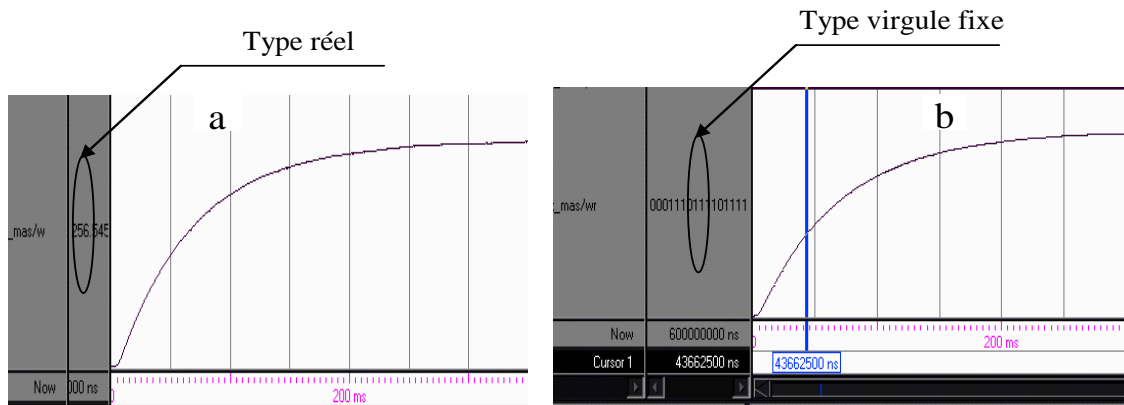


Figure 4.15. La vitesse rotorique (rad/sec). (a). Le modèle de simulation non synthétisable. (b). Le modèle de simulation synthétisable utilisant la bibliothèque de virgule fixe.

On peut remarquer, que le modèle non synthétisable décrit par les figures 4.14.a et 4.15.a représente l'évolution temporelle de couple et de la vitesse de rotation pour une durée de temps égale à 350 ms sous Modelsim, ne diffère pas du modèle synthétisable décrit par les figures 4.14.b et 4.15.b. Ce modèle de simulation synthétisable est obtenu en utilisant la représentation des réels en virgule fixe, l'algorithme de CORDIC pour les fonctions mathématiques et les initialisations des variables et des signaux. Par contre la légère différence est due aux approximations faites lors de la description du modèle de simulation synthétisable. Parmi ces approximations on cite [SoGd08] :

- Représentation des grandeurs réelles en virgule fixe sous les formats de 16 bits
- Introduire le temps mort entre deux interrupteurs sur un même bras de l'onduleur
- temps de retard des portes logiques qu'on tient compte à partir de cette étape.

Les résultats obtenus avec Modelsim en terme de comparaison avec les deux versions du modèle de simulation synthétisable et non synthétisable montrent bien la justesse des solutions adoptées et la bonne conformité des résultats.

### 4.2.10 Conclusion

Dans cette première partie, on a traité l'environnement et la méthodologie de conception utilisée pour implémenter la commande DTC d'une machine asynchrone. On a présenté en premier lieu le flot de conception adopté. Ensuite la modélisation et la spécification fonctionnelle de haut niveau de la chaîne de commande. En deuxième lieu, on a donné la méthodologie suivie et les outils utilisés lors de l'implémentation de la commande DTC d'une machine asynchrone à base d'un module matériel sur une cible FPGA. En outre, une description en VHDL de tout le système de puissance est abordée avec une étude comparative entre les résultats de simulation obtenus avec ceux établis dans le deuxième chapitre avec Matlab. Enfin, via l'outil ISE Xilinx, on a appliqué les techniques de synthèse incrémentale. Cette étude va nous ouvrir des horizons pour étudier les performances de contrôle direct intelligent de couple de la machine asynchrone.

## 4.3 Implémentation de la Commande Directe Floue de Couple

Aussi bien dans la littérature que dans le secteur industriel, de nombreuses études ont été effectuées autour de l'implémentation de la logique floue sur des solutions matérielles, essentiellement les FPGAs [ABar06] [KmDe06]. En particulier dans le domaine industriel, l'implémentation de la logique floue sur les FPGAs a été utilisée dans l'exploitation de l'énergie solaire [AMel10] [Ames11] [FChe11], la commande des robots [TzLI03] [ShIs08], des machines synchrones [YiKu09] [YiKu11] et des machines asynchrones [MoEl08] [RAru10] [RpDh12].

Cette partie décrit l'implémentation matérielle de la logique floue dédiée à la commande DTC de la machine asynchrone. Cette implémentation vise l'exploitation des nouvelles solutions matérielles pour la commande des machines asynchrones, ainsi que l'amélioration des performances de la Commande DTC de la MAS afin de remédier aux inconvénients de cette commande cités dans le deuxième chapitre. Ce module va être divisé en plusieurs entités décrites dans les paragraphes suivants.

### 4.3.1 Description du module Fuzzification

La fuzzification est le processus de conversion des données d'entrée en valeurs linguistiques floues. Dans la littérature, Il y a deux solutions matérielles pour déterminer le degré d'appartenance à un ensemble flou à partir d'une fonction d'appartenance. La première solution est l'approche orientée mémoire, comme son nom l'indique, pour chaque nombre fini d'entrées les valeurs de sortie sont calculées en off-line puis elles seront sauvegardées dans une mémoire RAM. L'avantage de cette solution est qu'elle est simple de changer une fonction d'appartenance. Cependant, la majeure limitation pour cette solution est qu'elle demande un grand espace mémoire qui augmente à la puissance de 2 chaque fois qu'on augmente le nombre d'entrée/sortie de la fonction d'appartenance.

La deuxième solution est l'approche orientée calcul, seulement les caractéristiques des fonctions d'appartenances sont sauvegardées dans une mémoire RAM afin de simplifier le calcul en on-line des valeurs de sortie de chaque fonction d'appartenance. Pour le cas des fonctions d'appartenances triangulaires, leurs caractéristiques sont le centre du triangle "*c*" et la pente "*a*". L'implémentation matérielle de cette solution est un circuit combinatoire qui



peut comporter des additionneurs, soustracteurs, multiplexeurs, multiplieurs et la plupart du temps une unité de contrôle. Dans cette étude, on adopte l'approche orientée calcul. On présente la description matérielle du module fuzzification. Les entrées/sorties sont définies sous 16 bits.

- Les fonctions d'appartenances de l'erreur flux illustrées par la figure 3.6 sont caractérisées par une pente  $a=1$ , donc seulement leurs centres  $c=-1, 0$  et  $1$  sont sauvegardés dans une mémoire RAM. Les valeurs de sortie pour une entrée "x" sont calculées en utilisant l'expression (4.1) suivante :

$$FA1(x, c) = \begin{cases} 1 - |x - c| & \text{Si } FA1 \succ 0 \\ \text{Sinon} & 0 \end{cases} \quad (4.1)$$

L'implémentation matérielle de cette expression est présentée par le circuit combinatoire de la figure 4.16. Ce circuit est composé de deux soustracteurs, un multiplexeur et une unité de contrôle.

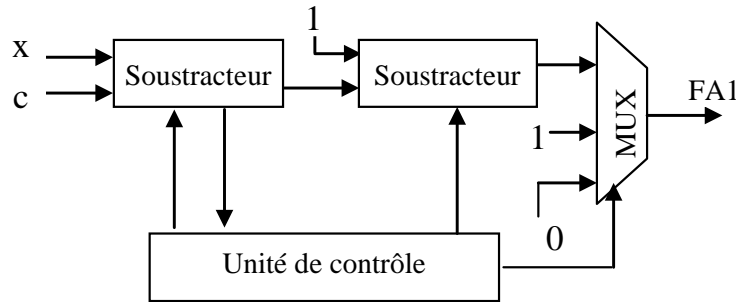
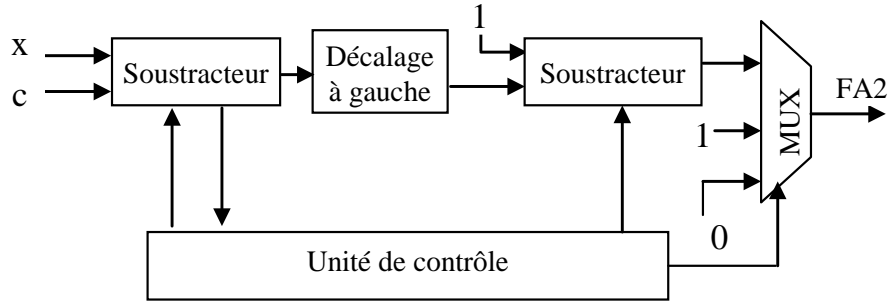


Figure 4.16. Le circuit combinatoire de  $e_\phi$

- Les fonctions d'appartenances de l'erreur couple illustrées par la figure 3.7 sont caractérisées par une pente  $a=1/2$ , donc seulement leurs centres  $c=-1, -0.5, 0, 0.5$  et  $1$  sont sauvegardés dans une mémoire RAM. Les valeurs de sortie pour une entrée "x" sont calculées en utilisant l'expression (4.2) suivante :

$$FA2(x, c) = \begin{cases} 1 - \left( \frac{1}{2} * |x - c| \right) & \text{Si } FA2 \succ 0 \\ \text{Sinon} & 0 \end{cases} \quad (4.2)$$

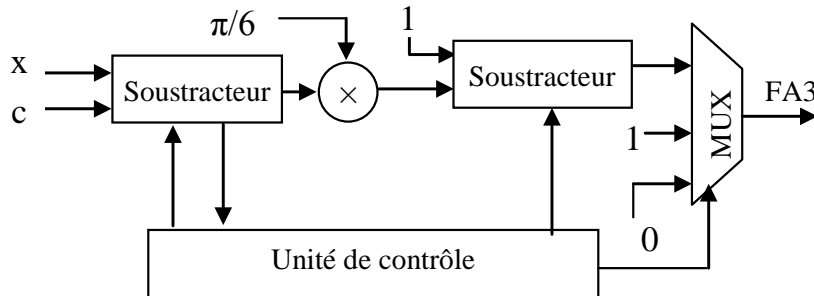
L'implémentation matérielle de cette expression est présentée par le circuit combinatoire de la figure 4.17. Ce circuit est composé de deux soustracteurs, un multiplexeur, un registre à décalage gauche et une unité de contrôle.

Figure 4.17. Le circuit combinatoire de  $e_c$ .

- les fonctions d'appartenance de l'angle du vecteur flux statorique illustrées par la figure 3.8 sont caractérisées par une pente  $a=\pi/6$ , donc seulement leurs centres  $c= \pi/12, 3*\pi/12, 5*\pi/12, 7*\pi/12, 9*\pi/12, 11*\pi/12, 13*\pi/12, 15*\pi/12, 17*\pi/12, 19*\pi/12, 21*\pi/12$  et  $23*\pi/12$  sont sauvegardés dans une mémoire RAM. Les valeurs de sortie pour une entrée "x" sont calculées en utilisant l'expression (4.3) suivante :

$$FA3(x, c) = \begin{cases} 1 - \left( \frac{\pi}{6} * |x - c| \right) & \text{Si } FA3 > 0 \\ \text{Sinon} & 0 \end{cases} \quad (4.3)$$

L'implémentation matérielle de cette expression est présentée par le circuit combinatoire de la figure 4.18. Ce circuit est composé de deux soustracteurs, un multiplexeur, un multiplieur par  $\pi/6$  et une unité de contrôle.

Figure 4.18. Le circuit combinatoire de l'angle du vecteur flux statorique  $\theta$ 

Le processus de fuzzification est exécuté en utilisant trois blocs de fuzzification, un pour chaque entrée, connecté comme le montre la figure 4.19. Ces blocs sont : bloc de fuzzification de l'erreur flux, bloc de fuzzification de l'erreur couple et bloc de fuzzification de l'angle du vecteur statorique  $\theta_s$ . Le bloc de registre 1 est composé de trois registres qui permettent de sauvegarder les trois valeurs de sortie des trois fonctions d'appartenance pour une entrée de l'erreur flux. De même le bloc de registre 2 est composé de cinq registres qui permettent de sauvegarder les cinq valeurs de sortie des cinq fonctions d'appartenance pour une entrée de l'erreur couple. Les douze registres qui forment le bloc de registre 3, permettent de sauvegarder les douze valeurs de sortie des douze fonctions d'appartenance pour une entrée de l'angle du vecteur statorique  $\theta_s$ .

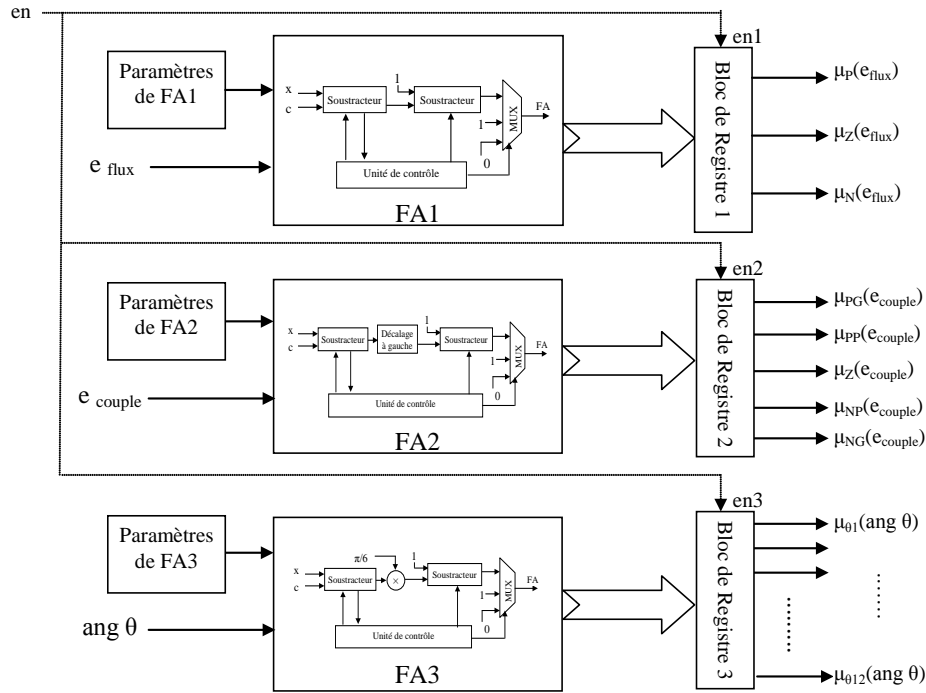


Figure 4.19. Architecture de la fuzzification

#### 4.3.2 Description du module moteur d'inférence

La description matérielle du module moteur d'inférence est illustrée par la figure 4.20. Ce module accepte comme entrée les trois blocs de registres issus du module fuzzification, le bloc sélecteur de règles permet de construire la base de règles formée par les 180 règles. Cette base est obtenue en réalisant toutes les combinaisons possibles entre les trois valeurs floues de l'erreur flux, les cinq valeurs floues de l'erreur couple et les douze valeurs floues de l'angle du vecteur statorique. Dans cette phase, afin de profiter des avantages des solutions matérielles, le traitement de ces 180 règles se fait en parallèle ; c'est-à-dire les 180 opérateurs  $Min i$  ( $i=1..180$ ) produisent une sortie disponible sur le même front d'horloge. Ensuite, les sorties de ces 180 opérateurs  $Min$  vont passer par l'étage des opérateurs  $Max$ . Les sorties de cet étage sont désignées  $m_1, m_2, \dots$  jusqu'à  $m_7$ . Pour chaque sortie  $m_i$  lui correspond la plus grande valeur parmi les sorties des opérateurs MIN dont leurs règles active la sortie "i". Par exemple, les blocs des opérateurs de MIN1 à MIN22 correspondent aux règles qui activent la sortie 1, c'est-à-dire active la tension  $V_1$  parmi les huit tensions de la figure 2.4 qu'on peut les appliquer aux bornes de la machine asynchrone à travers l'onduleur de tension. Le format de données durant cette phase est de 16 bits.

Cette solution présente un inconvénient majeur qui est le grand nombre de règles à utiliser. C'est vrai que le fait de les activer en parallèle permet d'un côté de diminuer le temps de calcul, mais d'un autre côté, entraîne l'augmentation énorme des besoins en ressources matérielles lors de la phase de l'implémentation. Le paragraphe suivant apporte une amélioration à cette solution caractérisée par l'ajout d'un bloc de sélecteur de règles actives.

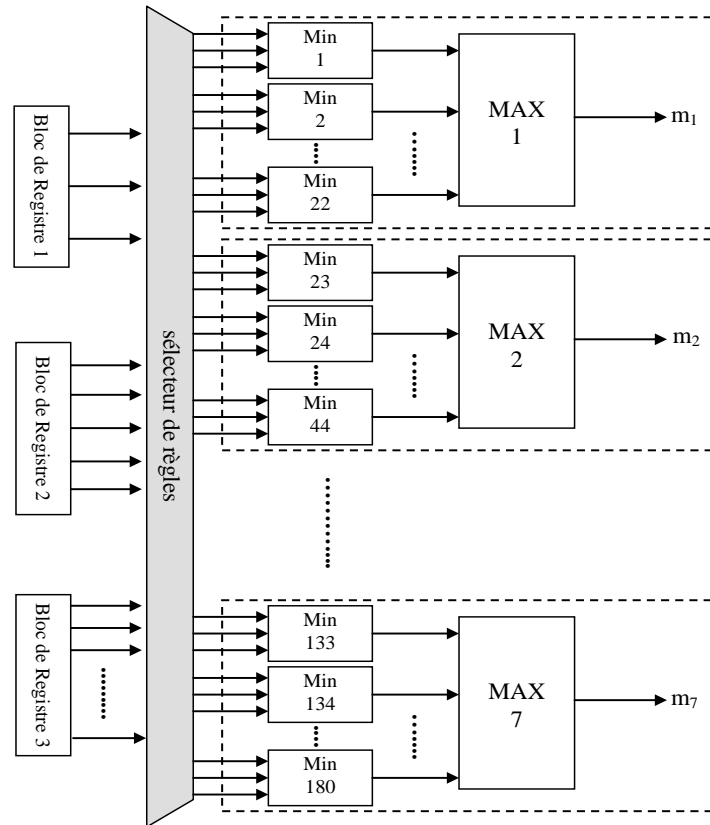


Figure 4.20. Architecture du module moteur d'inférence

### 4.3.3 Amélioration du module moteur d'inférence

Au lieu de traiter pour chaque nouvelle donnée toutes les 180 combinaisons de règles dans la base de règles, on peut améliorer ce module d'inférence en appliquant à chaque fois seulement les règles qui sont actives. Le sélecteur de règles actives, en fonction des degrés d'appartenances des erreurs de couple et de flux et de l'angle  $\theta_s$  fournies à partir du bloc de fuzzification, permet de déterminer les règles actives parmi les 180 prédéfinies. Ces règles actives ont des degrés d'activation ( $\alpha_i$ ) non nulle. Le nombre de règles actives à appliquer sera réduit et déterminé à partir de l'équation (4.4) suivante :

$$\text{Nombre de règles actives} = m^n \quad (4.4)$$

Où  $n$  représente le nombre d'entrée et  $m$  est le nombre maximal de chevauchement dans les ensembles flous. Dans notre cas, les figures 3.6 à 3.8 fixent  $n=3$  et  $m=2$ . Par conséquent, le nombre de règles actives à chaque fois est :  $m^n = 2^3 = 8$  règles. Le sélecteur de règles actives est utilisé pour minimiser le temps du calcul et les ressources matérielles réservées au contrôleur flou. La figure 4.21 illustre l'architecture du module moteur d'inférence avec sélecteur de règles actives.

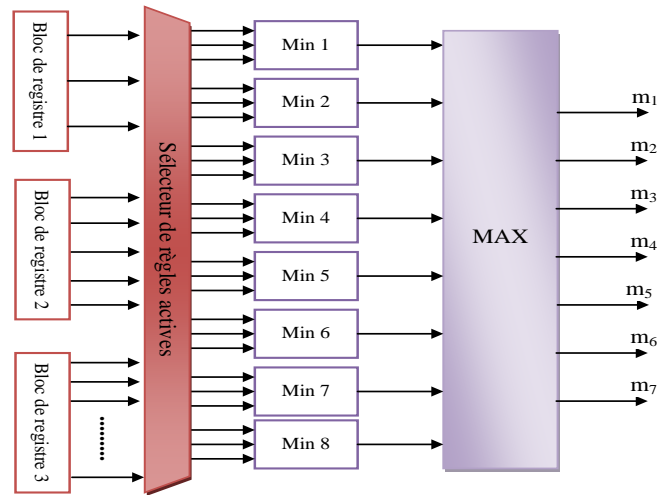


Figure 4.21. Architecture du module moteur d'inférence avec sélecteur de règles actives.

#### 4.3.4 Description du module defuzzification

La description matérielle du module defuzzification est réalisée par un opérateur MAX comme le montre la figure 4.22. Les entrées de ce module sont les sorties du module moteur d'inférence. Comme il a été précisé dans le paragraphe précédent, les  $m_i$  ( $i=1..7$ ) correspondent au degré d'activation des tensions  $V_i$ .

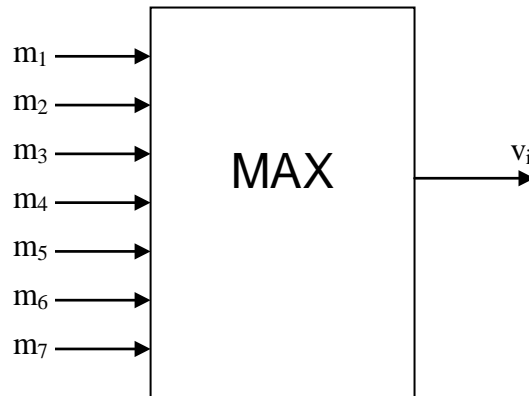


Figure 4.22. Architecture de la defuzzification

La sortie de ce bloc représentant la sortie de tout le bloc flou, correspond à la tension qu'on doit appliquer aux bornes de la machine à travers l'onduleur.

#### 4.3.5 Architecture de la Commande DTC floue

Cette phase consiste à l'intégration d'un système d'inférence flou dans l'algorithme de la commande DTC. Durant ce paragraphe et similairement à la simulation avec MATLAB, on va simuler à l'aide du langage VHDL la commande DTC conventionnelle et la commande DTC floue.

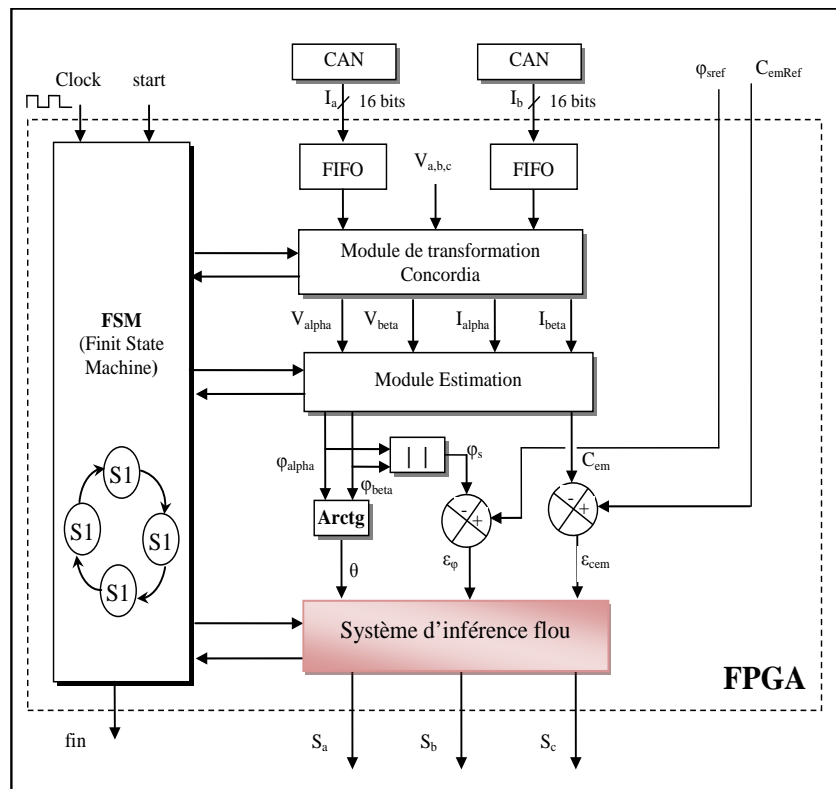


Figure 4.23. Architecture de la commande DTC floue

La structure de la commande DTC conventionnelle est illustrée par la Figure 4.8. Concernant la commande floue présentée par la figure 4.23, on va remplacer les blocs module comparateurs à hystérésis et la Table de sélection de la commande DTC par un système d'inférence flou qu'on a construit et testé. Pour comparer les deux approches de commande, on a simulé ces deux modules avec ModelSim. Les résultats sont illustrés par la figure 4.24. On remarque qu'il y a une amélioration au niveau du couple électromagnétique obtenu par la commande DTC floue par rapport à celle obtenue avec la commande DTC conventionnelle avec une importante diminution des ondulations.

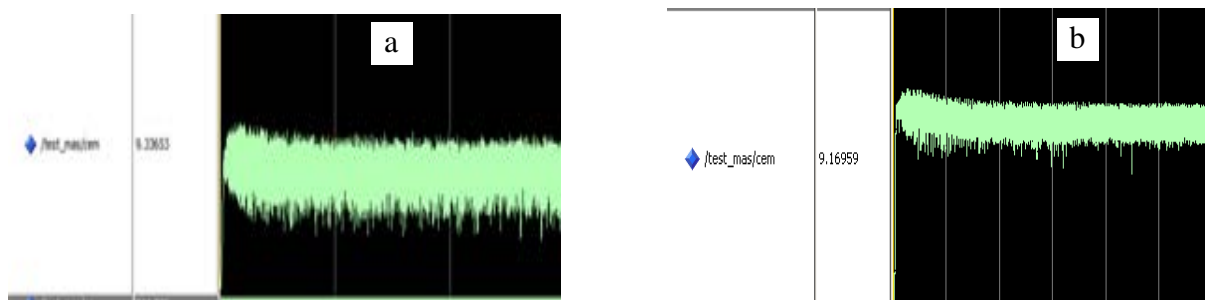


Figure 4.24: Couple électromagnétique obtenu par simulation VHDL. (a). Commande DTC conventionnelle. (b). Commande DTC Floue

La figure 4.25 illustre le schéma RTL du résultat de synthèse de la Commande Directe Floue de Couple de la machine asynchrone. La figure 4.26 présente le schéma RTL du résultat de synthèse du module Couple-Flux. Ce schéma est issu de l'outil *ISE Xilinx Design 12*, est utilisé pour implémenter l'algorithme de commande sur FPGA.

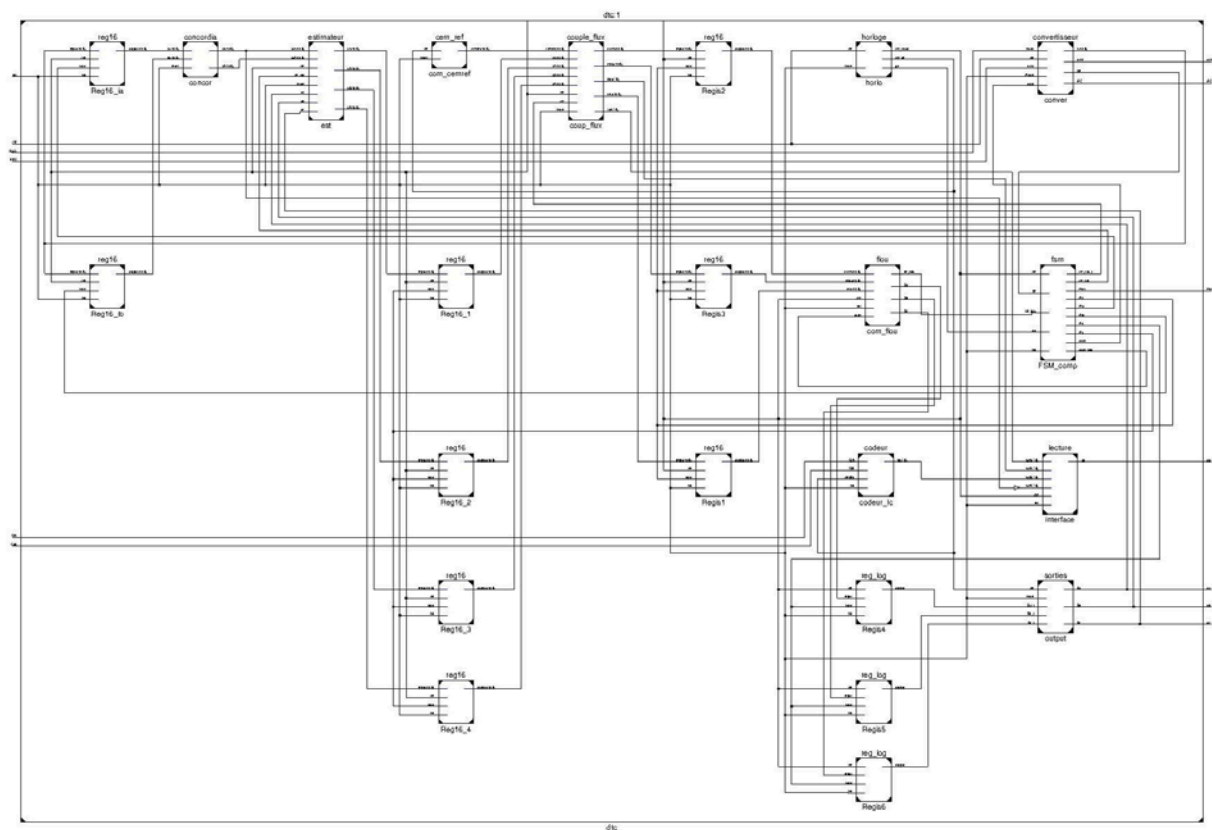


Figure 4.25. Schéma RTL de la DTC floue (Top Level)

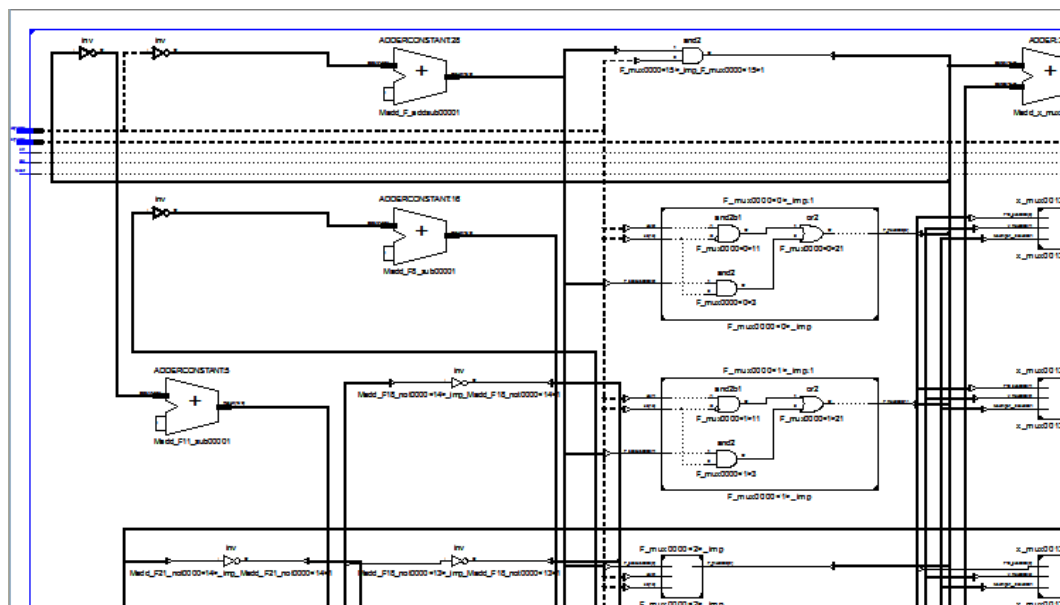


Figure 4.26. Portion du schéma RTL du module Couple-Flux

Le tableau 4.2 présente les performances en termes de consommation de ressources, obtenues lors de l'implémentation de l'architecture de la commande DTFC sur l'FPGA VIRTEX 5 donnée par la figure 4.23. Les ressources consommées sont élaborées pour un format en virgule fixe de 16 bits.

Tableau 4.2 : Ressources utilisées sur le FPGA par l'algorithme de commande DTFC

Ressources	utilisées	Accessibles
Les pins E/S	13	640
Nombre de buffer global	4	32
Nombre de slices LUT	4655	44800
Nombre de slices registres	966	44800
Nombre de Latch	678	4943
Nombre de DSP48Es	43	128

La fréquence maximale de l'horloge est fixée par l'outil de synthèse égale à 36.137MHz, qui correspond à une période minimale de 27.672ns. La figure 4.27 présente le résultat de routage de l'architecture de la commande floue de la machine asynchrone sur FPGA Virtex 5.

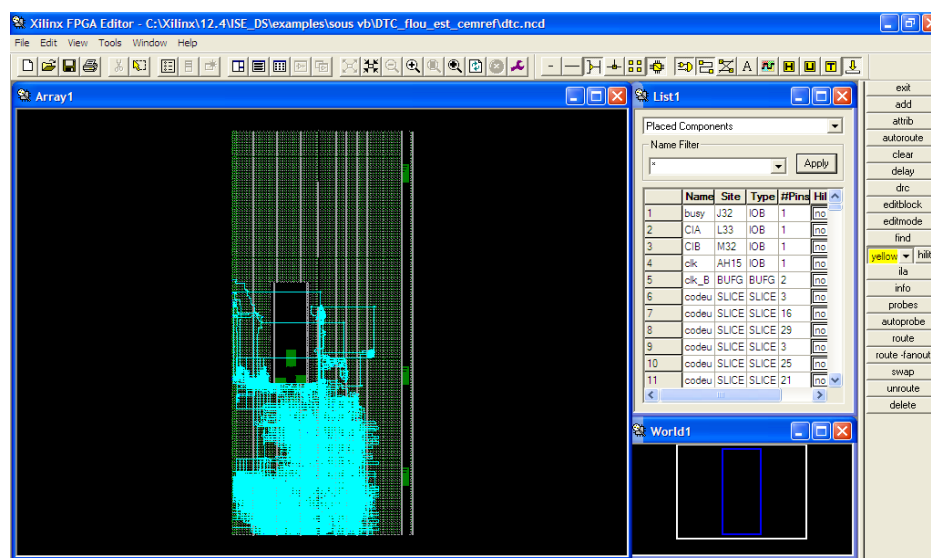


Figure 4.27. Résultat de routage de l'architecture de la commande floue

Le paragraphe suivant s'intéresse à l'implémentation de la Commande Directe Neuronale de Couple.

#### 4.4 Implémentation de la Commande Directe Neuronale de Couple

Les réseaux de neurones confirment leur efficacité dans plusieurs domaines, tels que la reconnaissance des formes, classification et le contrôle de processus industriels. Les travaux de [AdMe07], [ChFo09] et [NgNg10] présentent un état de l'art sur l'utilisation des réseaux de neurones. Cependant, les techniques de réseau de neurones trouvent des difficultés pour s'intégrer dans l'environnement industriel vue la grande capacité de calcul exigée. Ainsi est née l'idée de l'implémentation matérielle des réseaux de neurones. Dans la littérature, plusieurs travaux traitent de la conception et de l'implémentation des applications à base de réseaux de neurones sur des circuits intégrés matériels tels que les FPGAs. On cite, parmi ces travaux qui nous intéressent dans notre domaine de la commande des machines asynchrones [BiBo07] [FjLi08] [DaZa08] [MaPe09] [FaAr10] [DvAn11].

Cette partie décrit l'implémentation matérielle du réseau de neurones dédié à la commande DTC de la machine asynchrone. L'apprentissage de cette commande est effectué avec



l'algorithme de rétropropagation du gradient. Ce module va être divisé en plusieurs entités qui sont décrites dans les paragraphes suivants :

#### 4.4.1 Description de l'unité neurone :

La description sous VHDL d'un neurone illustrée par la figure 4.28 est formée par une RAM, deux multiplexeurs et un MAC (Multiplieur accumulateur). C'est l'unité de base dans la réalisation du réseau de neurones. La figure 4.29 montre l'entité du code VHDL de la description matérielle d'un neurone.

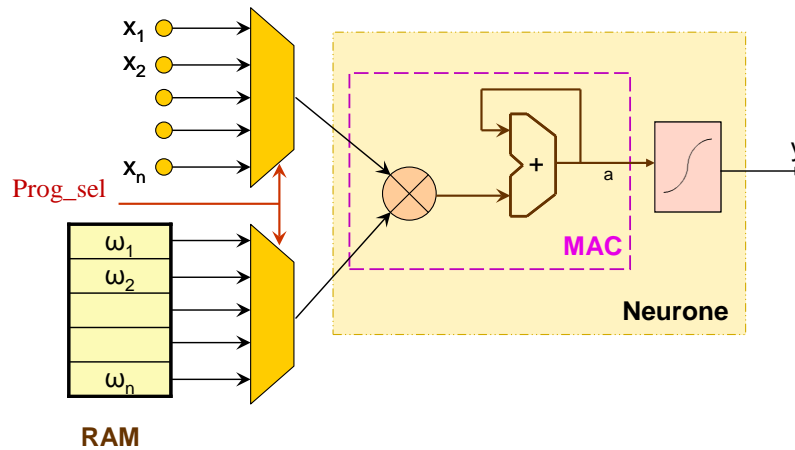


Figure 4.28. Implémentation matérielle d'un neurone

Le choix de la fonction d'activation est très important dans le bon fonctionnement d'un réseau de neurones qui peut exister suivant deux types : linéaire, ou non-linéaire, comme tangente hyperbolique, sigmoïde, etc. [HayS99]. La réalisation de la fonction d'activation linéaire est relativement simple, par contre, la réalisation des fonctions d'activation non-linéaires est beaucoup plus complexe. Précisément, la méthode la plus rapide d'implémenter ces fonctions d'activation est de remplir des tables de valeurs (look up table) prè-calculées de la fonction d'activation choisi [AmOm06]. Ces tables seront vues comme des ROMs lors de la synthèse. L'inconvénient majeur de cette solution est le besoin d'utiliser beaucoup d'espaces mémoire. Une autre solution plus complexe mais demande moins d'espace mémoire est d'approximer la fonction d'activation à l'aide d'une fonction linéaire par morceaux. Dans cette dernière solution, l'espace mémoire se réduit et de même pour la précision. Un compromis entre la nécessité en espace mémoire et précision des fonctions d'activation doit être alors étudié. Dans notre cas, on a opté pour la première solution, on a utilisé une table de valeurs mémorisée dans une ROM dans laquelle, on a choisi 255 valeurs équidistantes dans le domaine de définition de la fonction d'activation, et à chaque valeur correspond son image par la tangente hyperbolique.

Ce module réalise le fonctionnement d'un seul neurone, il est activé par le signal *prog\_sel*. Le neurone effectue d'abord le produit des données en entrée avec les poids synaptiques correspondants mémorisés dans une RAM, puis il les additionne. Ce calcul est réalisé avec l'opérateur MAC. Le résultat de ces opérations est soumis à la fonction d'activation.

```

Entity neurone is
  Generic(n:Integer:=3);
  Port( rst : in std_logic;
        prog_sel: in std_logic;
        x : in Tab_fix(1 to n);
        w : in Tab_fix(1 to n);
        b : in SFix(MID_index downto LSB_index);
        y : out SFix(MID_index downto LSB_index) );
End neurone:

```

Figure 4.29: Modèle VHDL de l'entité neurone

La description structurelle de l'entité neurone, présentée par la figure 4.29, permet de spécifier lors de la simulation le nombre d'entrées de façon générique. En fait, pour notre structure on a besoin de neurones avec trois entrées et d'autres avec 10.

Afin de vérifier le fonctionnement de ce module, on a développé un fichier test\_bench et un fichier de configuration permettant de simuler le fonctionnement de notre module. L'exemple qu'on a testé concerne un neurone formé de trois entrées ayant les valeurs (1, 1, 1) et les poids correspondants ( $w_1=0.4$ ,  $w_2=0.5$ ,  $w_3=0.25$ ,  $b_1=1$ ) présenté sous un format de virgule fixe de 16 bits ; 1 bit de signe, 5 bits pour la partie entière et 10 bits pour la partie fractionnaire. Le résultat obtenu est comparé avec celui donné par le module neurone programmé avec MATLAB. La figure 4.30 présente le résultat de simulation d'un neurone sous Modelsim 6.5.

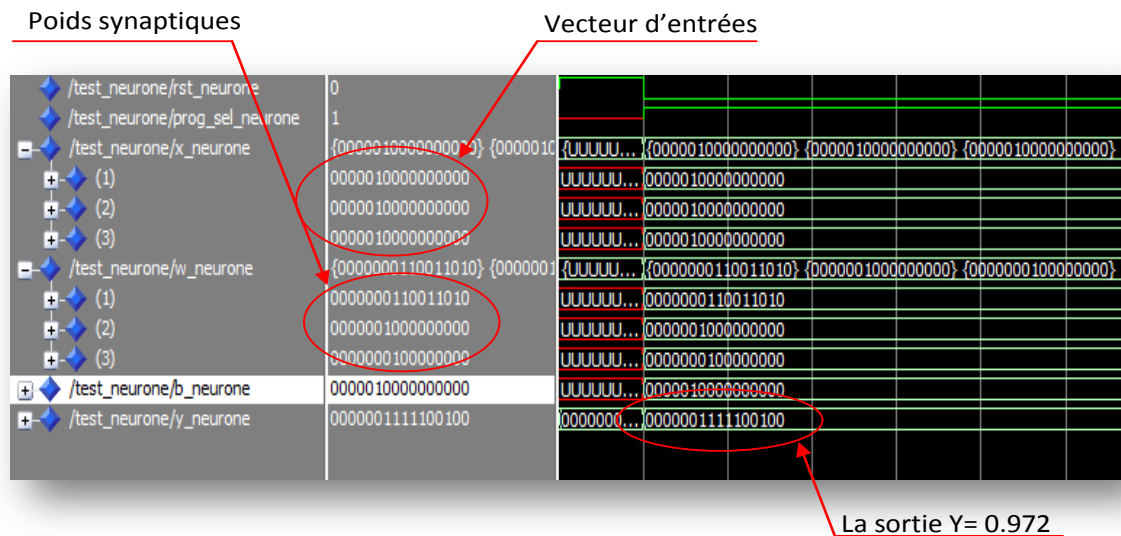


Figure 4.30. Simulation de la séquence d'exécution du module neurone

On obtient comme valeur de sortie  $y = 0.972$ , avec Matlab le résultat de sortie est égale à 0.973 qui correspond à un taux d'erreur de  $10^{-3}$  près. Ce module a subi plusieurs tests afin de valider son architecture développée sous VHDL. La figure 4.31 présente la schématisation de haut niveau du module neurone.

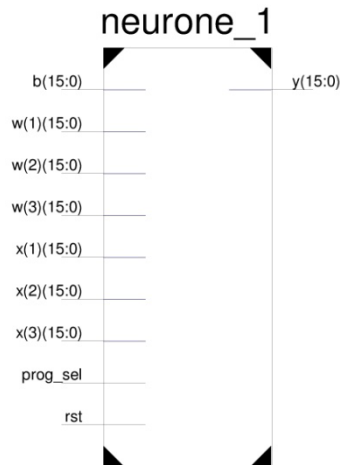


Figure 4.31. Schéma haut niveau du module Neurone.

#### 4.4.2 Description d'une couche

Le module couche est réalisé à base du module neurone. En effet, chaque couche est composée de plusieurs neurones fonctionnant en parallèle. C'est pour cela, notre approche de description du module neurone est réalisée en respectant la contrainte de réutilisabilité. De cette façon, on fait appel au composant neurone tant qu'il est nécessaire, tout en fixant à chaque fois la généricité du composant neurone, c'est-à-dire fixer d'avance le nombre d'entrée de chaque neurone. La figure 4.32 illustre un exemple de couche de trois entrées, un biais et de 10 neurones.

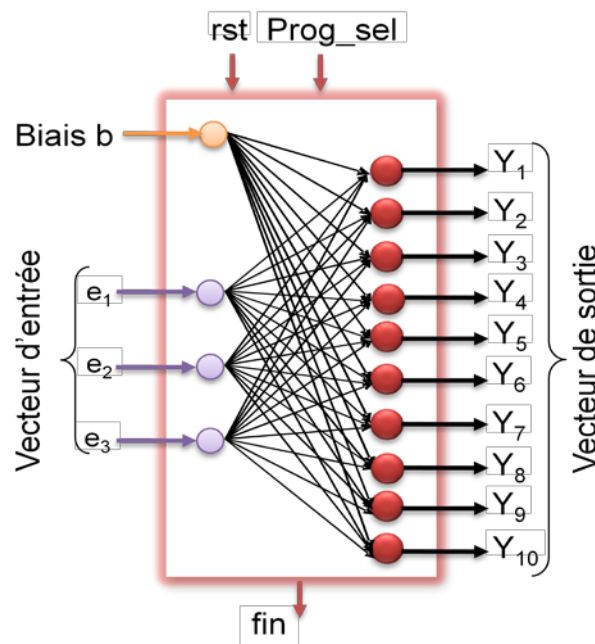


Figure 4.32 : Structure d'une couche

Dans le cas de ce travail on a développé trois couches formant notre réseau de neurones : la première avec trois entrées et 10 sorties, la deuxième avec dix entrées et dix sorties et la troisième avec dix entrées et trois sorties. La figure 4.33 illustre le style de description du module couche1 sous VHDL. Avec ce style de description, instantiation des composants neurones dans l'architecture, les neurones vont fonctionner en parallèle. On peut remarquer

d'après la description, que cette couche corresponde à la première couche, puisque les dix neurones instanciés dans cette architecture présentent 3 entrées chacun.

```

Entity couche1 is
  Port( rst   : in std_logic;
        prog_sel : in std_logic;
        e     : in Tab_fix(1 to 3);
        y     : out Tab_fix(1 to 10);
        fin   : out std_logic );
end couche;

Architecture arch_couche of couche is
  .....
Begin
  n1 : neurone generic map (3) port map (rst,prog_sel,e,w1(1),b1(1),y(1));
  n2 : neurone generic map (3) port map (rst,prog_sel,e,w1(2),b1(2),y(2));
  .....
  n10: neurone generic map (3) port map (rst,prog_sel,e,w1(10),b1(10),y(10));
End arch_couche;

```

Figure 4.33: Modèle VHDL de la couche1

Cette entité est activée par le signal *prog\_sel* qui va être associé aux "n" neurones composant la couche. En effet, la première couche présente trois entrées et elle est composée de dix neurones. Chaque neurone détermine la sortie associée aux entrées, on aura ainsi un vecteur de sortie de taille dix. Ce module est simulé avec un vecteur d'entrée de trois données (e(1), e(2) et e(3)) et de la matrice poids w1 (1...10) qui sont chargés de la base d'apprentissage mémorisée dans une RAM. Pour chaque entrée, on associe un vecteur de poids de trois éléments et un biais. Le résultat de simulation du module *couche1* avec l'outil *Modelsim* est illustré par la figure 4.34.

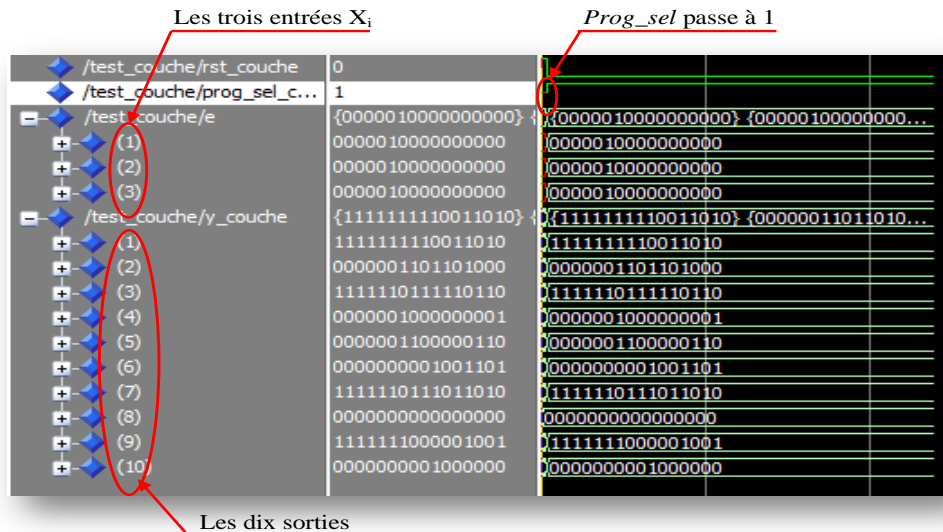


Figure 4.34 : Résultat de simulation d'une couche

On remarque que le module ne sera activé que lorsque *prog\_sel* passe à 1. On obtient à la sortie le vecteur y de dix éléments. Les valeurs de sortie sont comparées à celles obtenues

avec MATLAB. On obtient des valeurs avec une erreur de  $10^{-3}$  près. Plusieurs tests ont été réalisés afin de vérifier le bon fonctionnement de cette architecture.

#### 4.4.3 Description de la propagation

Le choix du nombre de couches et de neurones par couche est fixé par les résultats de simulations et de tests qui mènent aux performances maximales de la commande. Le réseau de neurones qu'on a développé pour la commande de la machine asynchrone contient trois couches. La sortie de chaque couche correspond à l'entrée de la couche suivante. Le vecteur d'entrée va être propagé à travers les neurones des différentes couches vers la couche de sortie. Chaque neurone  $N_i$  de la couche  $i$  calcule sa sortie  $Y_i$  à partir d'une somme pondérée des sorties  $Y_{i-1}$  des neurones de la couche  $i-1$  qui lui sont connectés. Les vecteurs d'entrées, les vecteurs de poids et les vecteurs de biais associés sont chargés directement de la base d'apprentissage. L'architecture de ce module est illustrée par la figure 4.35.

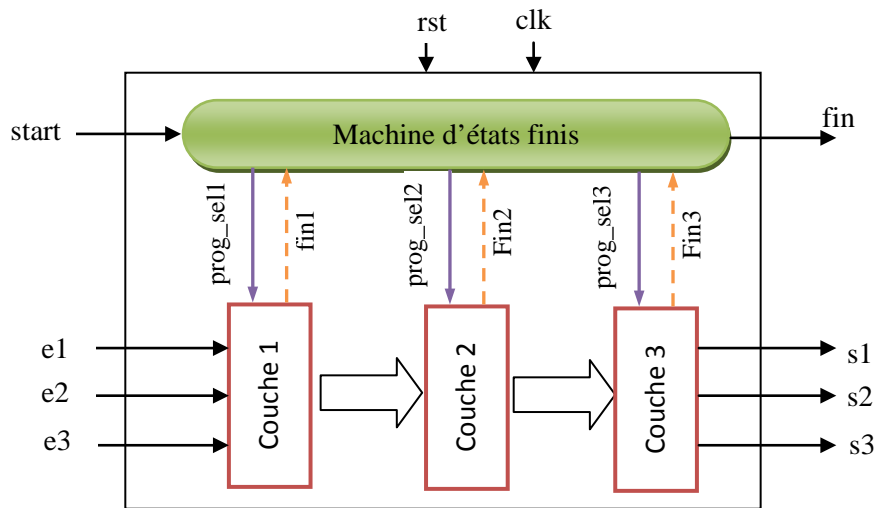


Figure 4.35 : Structure du réseau de neurones.

L'entité propagation réalisant le fonctionnement du réseau de neurones est décrite sous VHDL comme indiqué par la figure 4.36. Le mode propagation est activé en mettant à 1 l'entrée *start*. Un cycle de propagation se déclenche dès le prochain front montant de l'horloge. Par la mise à 1 du signal de sortie *fin*, il annonce la fin du cycle de propagation. Ce signal *fin* est remis à 0 à chaque début de cycle et maintenu à cette valeur jusqu'à la fin. L'activation et la désactivation des couches sont contrôlées par une machine d'états finis. Les signaux *prog\_sel1*, *prog\_sel2* et *prog\_sel3* vont activer successivement les différentes couches du réseau.

```

Entity prop is
  Port (rst : in std_logic;
        clk : in std_logic;
        start: in std_logic;
        fin: out std_logic
        );
End prop;

Architecture arch_prop of prop is
  .....
Begin
  f_prop: fsm_prop port map (rst,clk,start,prog_sel1,prog_sel2,prog_sel3,fin);

  c1: couche port map (rst, prog_sel1, e(compt), s1(compt));
  c2: couche2 port map (rst, prog_sel2, s1(compt), s2(compt));
  c3: couche3 port map (rst, prog_sel3, s2(compt), ss(compt));
End arch_prop;

```

Figure 4.36 : Modèle VHDL de la propagation

Le réseau de neurones doit apprendre les 36 vecteurs d'entrées de la base d'apprentissage. Les sorties de tous les exemples doivent être calculées avant de passer à la phase de correction de poids. C'est-à-dire, il faut utiliser 36 cycles de propagation avant d'entamer un cycle de rétropropagation afin d'ajuster les poids. Pour réussir cette tâche, la machine d'états finis est dotée d'un compteur qui contient le nombre d'exemples calculés. Le tableau 4.3 et la figure 4.37 résument le fonctionnement de la machine à états finis.

Tableau 4.3 : Fonctionnement de la machine à états finis de la propagation

Etat courant	Condition	Etat suivant	Sortie de fsm
Init_prop : initialisation de la propagation	Start=1	St0_prop	prog_sel1=0 prog_sel2=0 prog_sel3=0 compt=0
St0_prop : passage à la première couche		St1_prop	prog_sel1=1 prog_sel2=0 prog_sel3=0 fin_prop =0
St1_prop : passage à la deuxième couche		St2_prop	prog_sel1=0 prog_sel2=1 prog_sel3=0 fin_prop =0
St2_prop : passage à la troisième couche et un cycle de propagation est complété		St3_prop	prog_sel1=0 prog_sel2=0 prog_sel3=1 fin_prop =0
St3_prop : s'il reste encore des exemples alors commencer un autre cycle de propagation si non revenir à l'état initial	compt < 36	St0_prop	Compt =compt+1 fin_prop=0
	compt =36	Init_prop	fin_prop=1

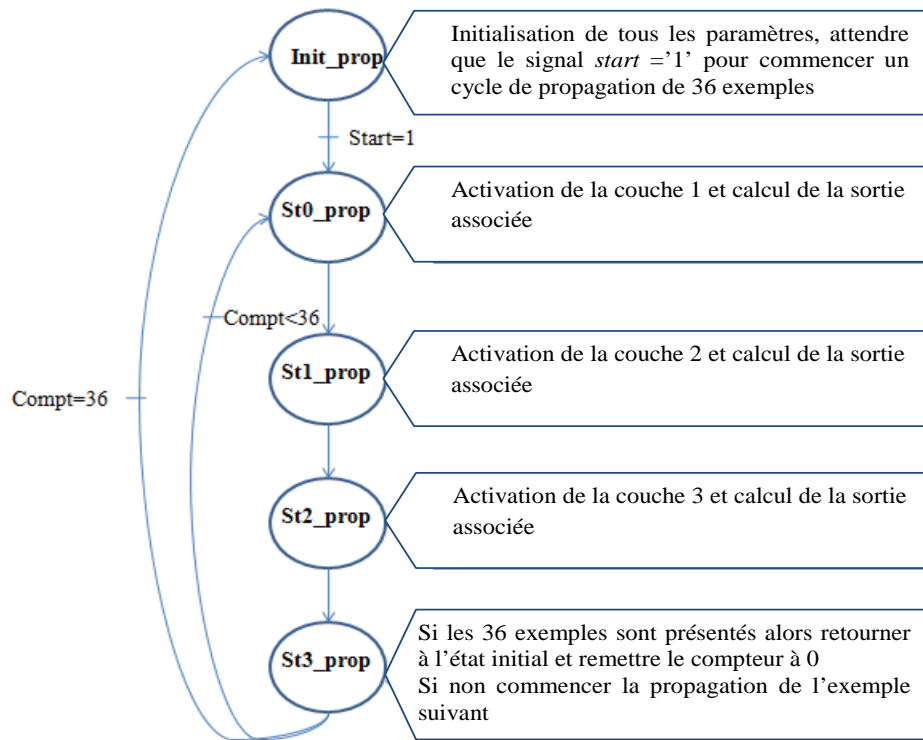


Figure 4.37. Les états de fsm\_prop

La simulation du réseau avec les 36 exemples de la base d'apprentissage avec des poids initiaux permet d'avoir les résultats décrits par la Figure 4.38.

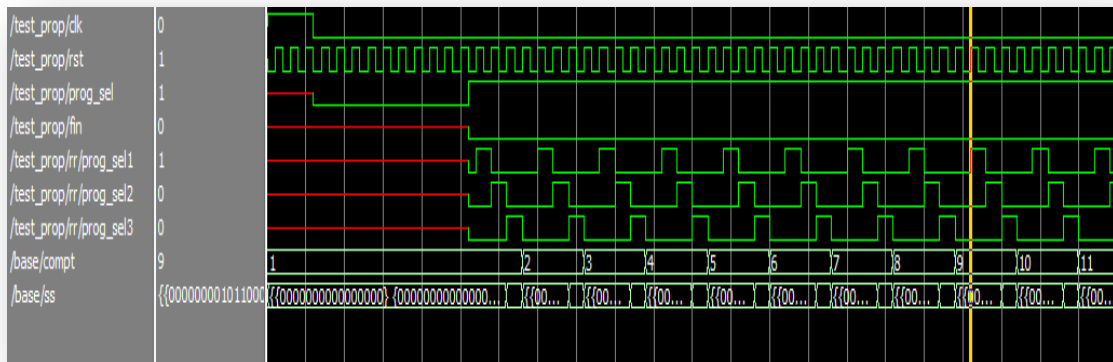


Figure 4.38 : Résultat de simulation de l'entité prop

On remarque que la phase d'apprentissage commence lorsque le signal *start*=1 et à chaque fois une seule couche est activée. On obtient ainsi les sorties associées aux 36 exemples. En comparaison avec les valeurs de sorties de 36 vecteurs d'entrées avec MATLAB on obtient une erreur de l'ordre  $10^{-3}$ .

#### 4.4.4 Description de l'algorithme de rétropropagation

Cet algorithme a pour objectif de mettre à jour les poids synaptiques d'un réseau de neurones afin qu'il réalise la tâche qui lui est demandée. Pour notre architecture, les poids sont choisis aléatoirement et à l'aide de cet algorithme on va les ajuster pour réussir la Commande Directe Neuronale de Couple des machines asynchrones.

Le module *retro* va mettre à jour les poids de réseau construit et basé sur le résultat du module *prop*. Ce module calcule les sorties associées aux entrées stockées dans la base d'apprentissage qui contient aussi le sorties désirées. En fonction de l'erreur entre les sorties calculées et désirées, les poids sont ajustés à partir de la couche de sortie vers la couche d'entrée.

```

Entity retro is
  Port (rst : in std_logic;
        Retro_sel : in std_logic;
        Fin_retro : out std_logic
        );
End retro;

```

Figure 4.39 : Modèle VHDL de l'entité retro

Ce mode est sélectionné en mettant à 1 l'entrée *retro\_sel*. Un cycle complet de rétropropagation afin d'ajuster les poids se déclenche dès le prochain front d'horloge. Il s'agit de calculer d'abord les erreurs pour tous les exemples sans mettre à jour les poids. Lorsque tous les exemples de la base d'apprentissage se présentent au réseau, on applique la rétropropagation en utilisant l'erreur totale. Cette méthode est préférée pour des raisons de rapidité et de convergence. La fin du cycle de rétropropagation est annoncée par la mise à 1 de la sortie *fin\_retro*. L'entité retro de la description de l'architecture du module de rétropropagation est illustrée par la figure 4.39.

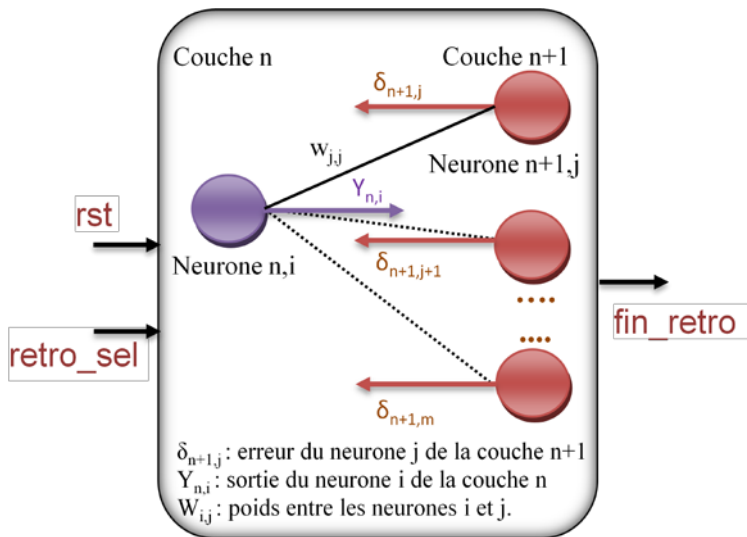


Figure 4.40 : Mise à jour de poids d'un neurone

L'erreur  $\delta_{n+1,j}$  est calculée en fonction de l'erreur accumulée de tous les exemples de la base d'apprentissage de  $\delta_{n+2,j}$  de la couche  $n+1$  et de  $Y_{n,i}$  de la sortie de la couche  $n$ . Le terme de correction des poids  $\Delta w_{ij}$  est calculé en fonction de l'erreur  $\delta_{n+1,j}$ , le pas d'apprentissage, et la sortie  $Y_{n,i}$  de la couche  $n$ . Ce terme sert à ajuster et à corriger le poids de neurone "i" de la couche  $n$ . Durant cette phase, on a présenté 36 vecteurs d'entrées calculés par le module "prop" et qui sont différents des sorties désirées. Il s'agit de calculer l'erreur quadratique



moyenne  $mse$  et de mettre à jour les poids en fonction de cette erreur. L'objectif de cet algorithme est de minimiser l'erreur quadratique  $mse$  qui est initialisée à une grande valeur.

#### 4.4.5 Réalisation de l'apprentissage

L'algorithme le plus répandu pour l'apprentissage des réseaux de neurones est l'algorithme de rétropropagation. Cet algorithme ajuste les poids d'un réseau dont l'architecture est fixée par le concepteur à chaque fois qu'un exemple est présenté. Cet ajustement est fait de telle sorte à minimiser l'erreur entre la réponse du réseau à une entrée et la sortie désirée. Ceci est réalisé en utilisant la méthode de descente de gradient.

La phase d'apprentissage consiste en premier lieu à présenter les exemples à faire apprendre au réseau et calculer la somme des erreurs  $\sum \delta_i$  entre la valeur désirée  $D$  et la valeur calculée par le réseau. En deuxième lieu, on active le module de rétropropagation pour répercuter l'erreur  $\delta_i$  de chaque neurone "i" de la couche  $n$  sur les neurones de la couche  $n-1$  et ceci afin de corriger les poids synaptiques de chaque neurone.

Les entités « prop » et « retro » réalisent un seul cycle de propagation et de rétropropagation à la fois. L'apprentissage du réseau consiste à répéter ce cycle tant de fois jusqu'à obtenir une erreur quadratique prédéfinie ou atteindre le nombre d'itérations. La figure 4.41 représente la structure du module apprentissage.

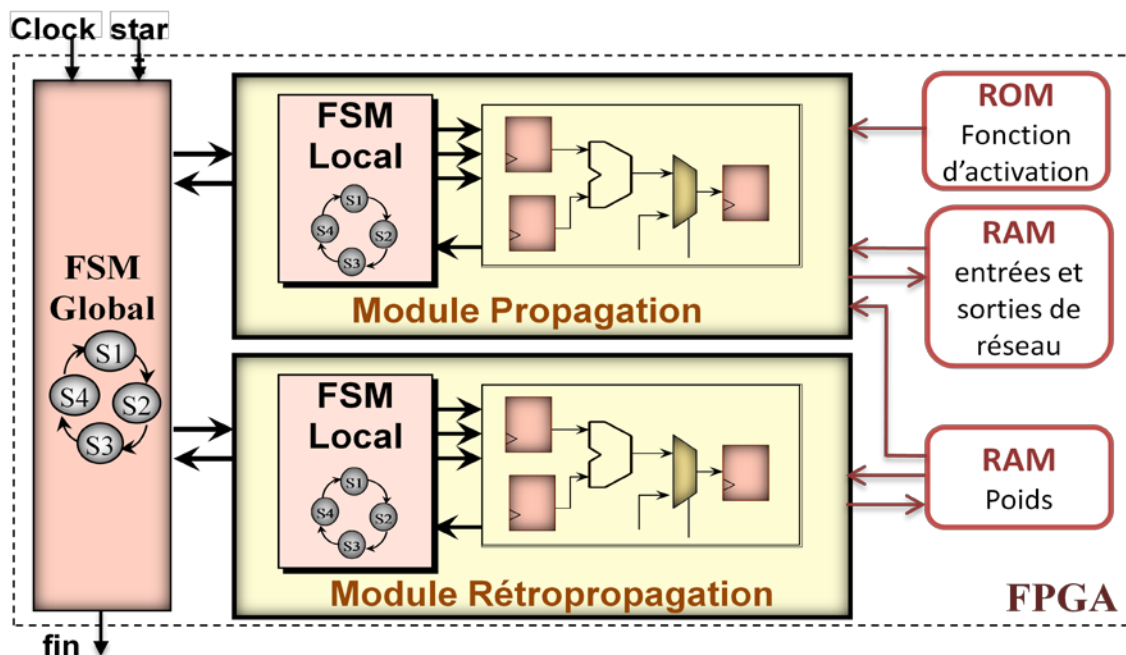


Figure 4.41 : Structure du module apprentissage.

La réalisation de l'apprentissage nécessite l'utilisation d'une machine à états finis pour commander le fonctionnement des composants « prop » et « retro » qui sont instanciés à l'intérieur de l'architecture de l'entité « apprentissage ». A chaque fois un cycle de propagation qui consiste à calculer les sorties associées aux données de la base d'apprentissage avec les poids courants suivi d'un cycle de rétropropagation qui va modifier les poids synaptiques en fonction de l'erreur entre ces sorties et celles désirées.

```

Entity apprentissage is
  Port (rst : in std_logic;
        clk : in std_logic
        );
End apprentissage;

Architecture arch_app of apprentissage is
  .....
Begin
  f_app: fsm port map (rst, clk, fin_prop, fin_retro, start, retro_sel);
  U1: prop port map (rst, clk, start, fin_prop);
  U2: retro port map (rst, clk, retro_sel, fin_retro);
End arch_app;

```

Figure 4.42 : Modèle VHDL de l'entité apprentissage

Le fonctionnement de la machine à états finis globale de l'apprentissage est résumé dans le tableau 4.4

Tableau 4.4 : Fonctionnement de la fsm liée à l'apprentissage

Etat courant	Condition	Etat suivant	Sortie de fsm
Init : initialisation des paramètres d'apprentissage.		St0	start<='0'; retro_sel<='0'; epochs <= 0;
St0 : passage à un cycle de propagation		St1	start<='1'; retro_sel<='0';
St1 : si la propagation de données à travers les différentes couches est terminée alors passer à la rétropropagation si non rester à cet état jusqu'à fin de la propagation.	fin_prop= 1	St2	start<='0'; retro_sel<='0';
	fin_prop= 0	St1	start<='0'; retro_sel<='0';
St2 : début d'un cycle de rétropropagation		St3	retro_sel<='1';
St3 : si la correction de poids est effectuée alors un cycle d'apprentissage est terminé et on va incrémenter le compteur de nombre d'itérations	fin_retro='1'	St4	epochs <= epochs+1; retro_sel<='0';
	fin_retro='0'	St3	retro_sel<='0';
St4 : si on obtient une erreur quadratique acceptable ou qu'on atteint le nombre maximal d'itération alors l'apprentissage est arrêté sinon on commence un autre cycle d'apprentissage.	mse>0.01 et epochs<500	St0	
	mse<0.01 et epochs<500	St5	
St5 : fin de l'apprentissage de réseau de neurones		Init	start <='0'; retro_sel<='0';

Comme l'explique la figure 4.34 et le tableau 4.4, le cycle d'apprentissage est répété tant de fois jusqu'à obtenir une erreur quadratique acceptable ou atteindre le nombre maximal d'itérations. On a fixé l'erreur à la valeur mse=0.01 et le nombre maximal d'itérations à epochs=500. Le fonctionnement de ce module d'apprentissage consiste à vérifier et à simuler en présence des entrées de la base d'apprentissage, les sorties désirées et les poids synaptiques initiaux enregistrés dans une RAM.

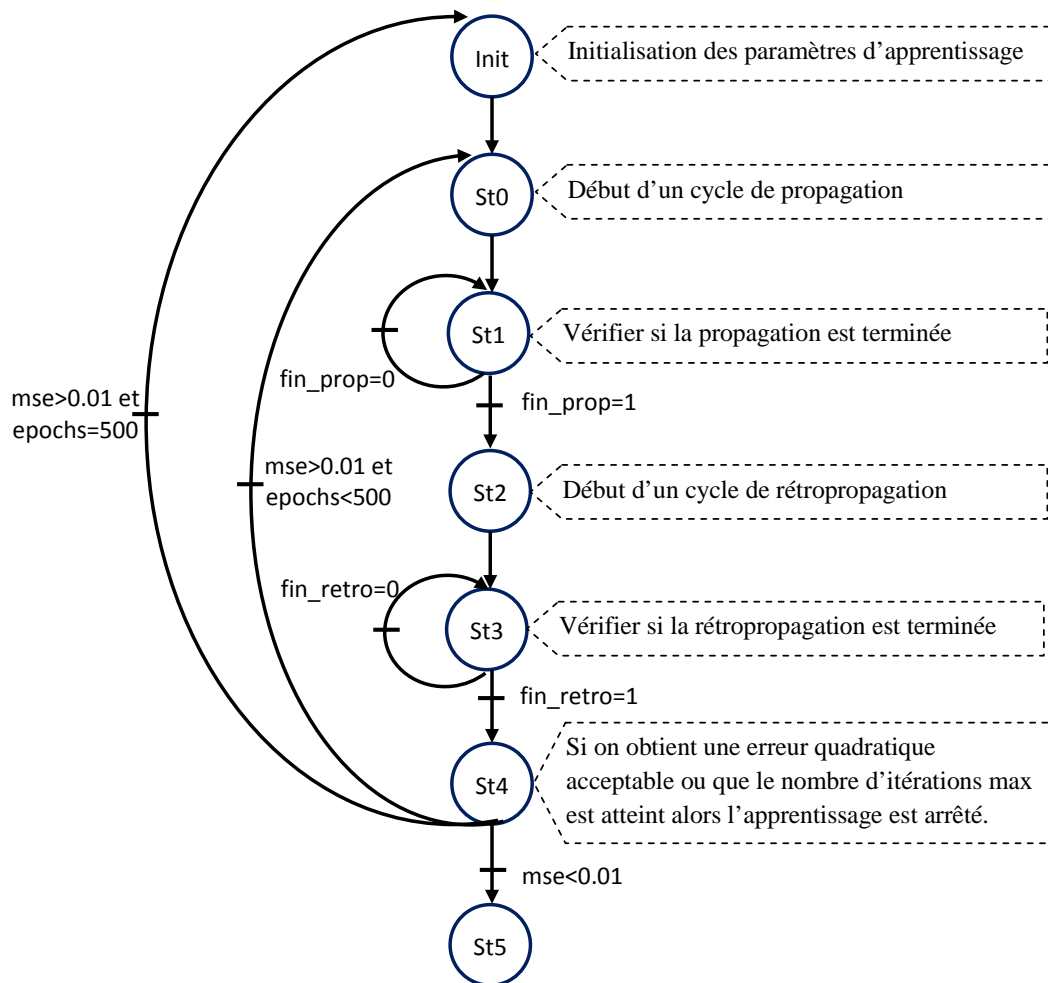


Figure 4.43. Les différents états de la machine d'états finis de l'apprentissage

Pour valider le réseau avec les nouveaux poids obtenus après apprentissage, on a simulé le réseau avec des exemples de la base d'apprentissage et on a vérifié bien qu'il réalise la tâche demandée. Le résultat de simulation du réseau avec un exemple de la base d'apprentissage est représenté par figure 4.44. Le vecteur d'entrée (1, -1, 1) correspondant à  $E_\phi$ ,  $E_c$  et secteur N se présente au réseau de neurones, la sortie tension calculée est (1, 0, 1) qui correspond à  $S_a$ ,  $S_b$  et  $S_c$  l'état des interrupteurs du l'onduleur. Plusieurs essais ont été effectués sur ce réseau de neurones, en présentant à chaque fois une entrée de la base de connaissance à l'entrée du réseau. En comparant la sortie calculée et désirée, on peut conclure que le réseau de neurones ainsi conçu s'adapte bien à la commande neuronale.

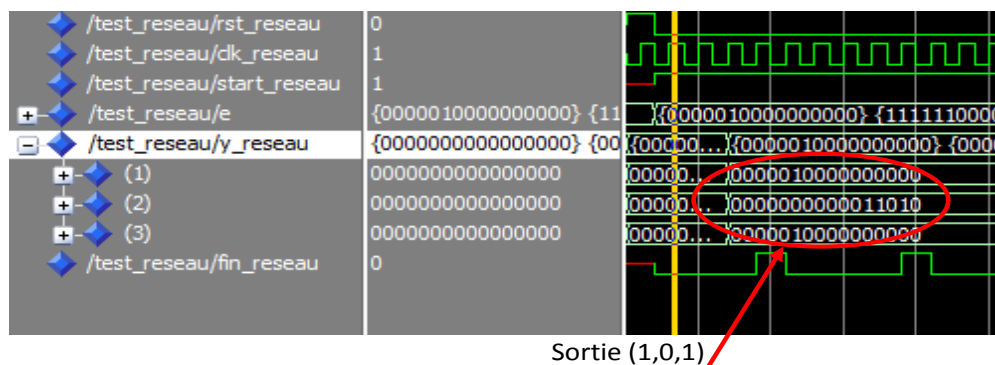


Figure 4.44. Résultat de simulation du réseau de neurones adapté à la commande.

Une fois la phase d'apprentissage est terminée, on obtient les nouveaux poids qui vont être utilisés dans la commande de la machine asynchrone.

#### 4.4.6 La commande DTC neuronale

Cette phase consiste à l'intégration du réseau de neurones développé dans les paragraphes précédents avec les poids finals dans l'algorithme de la commande DTC. Durant ce paragraphe et d'une façon similaire que les résultats obtenus avec MATLAB, on va simuler à l'aide du langage VHDL la commande DTC conventionnelle et DTC neuronale pour comparer les performances de la nouvelle technique de commande.

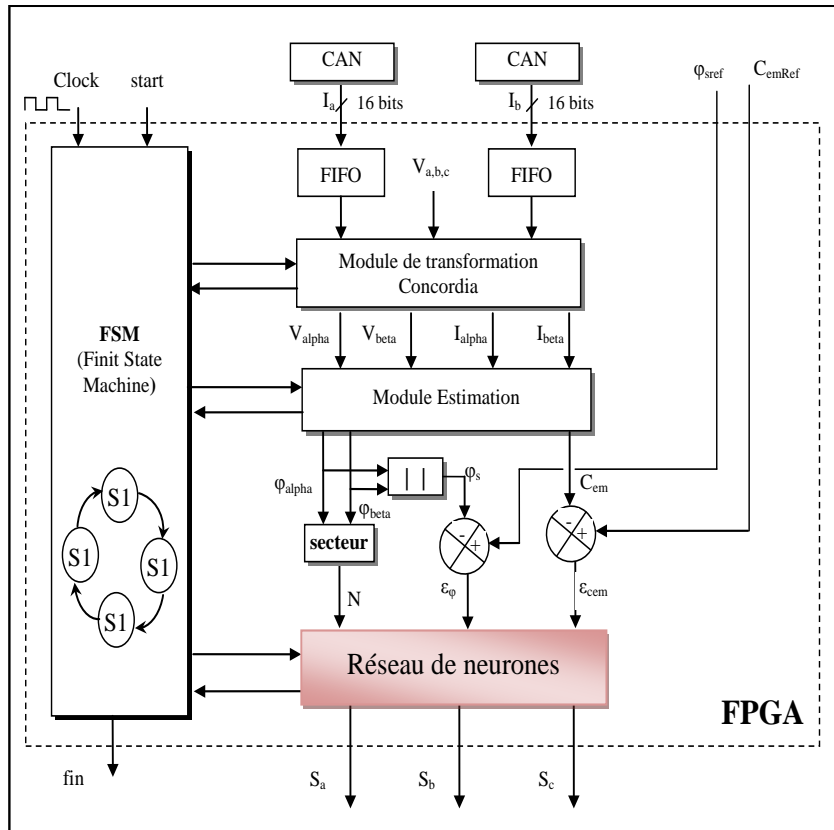


Figure 4.45 : Structure générale de la commande neuronale

La structure de la commande DTC conventionnelle est illustrée par la figure 4.8. La commande neuronale représentée par la figure 4.45, substitue les blocs module hystérésis et Table de sélection de la commande DTC par le réseau de neurones qu'on a construit et testé.

Pour comparer les deux approches de commande, on a simulé les deux modules avec ModelSim. Les résultats obtenus sont représentés par la figure 4.46.

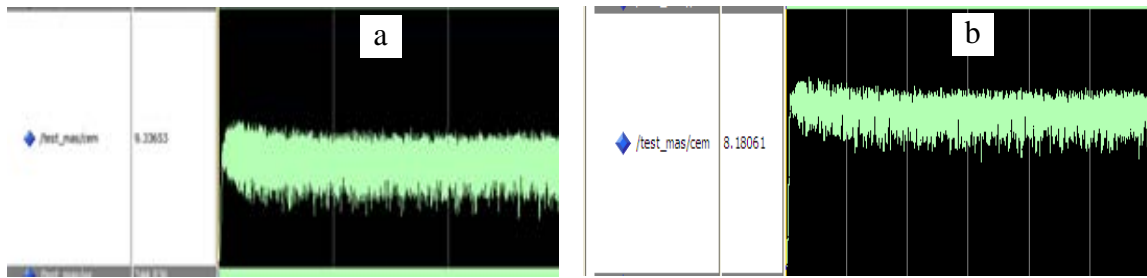


Figure 4.46. Couple électromagnétique obtenu par simulation VHDL. (a) Commande classique. (b).Commande neuronale.

On remarque qu'il y a une amélioration au niveau du couple électromagnétique obtenu par le réseau de neurones par rapport à celui obtenu avec la commande DTC conventionnelle marquée par une diminution légère des ondulations dues aux approximations faites lors la description du système.

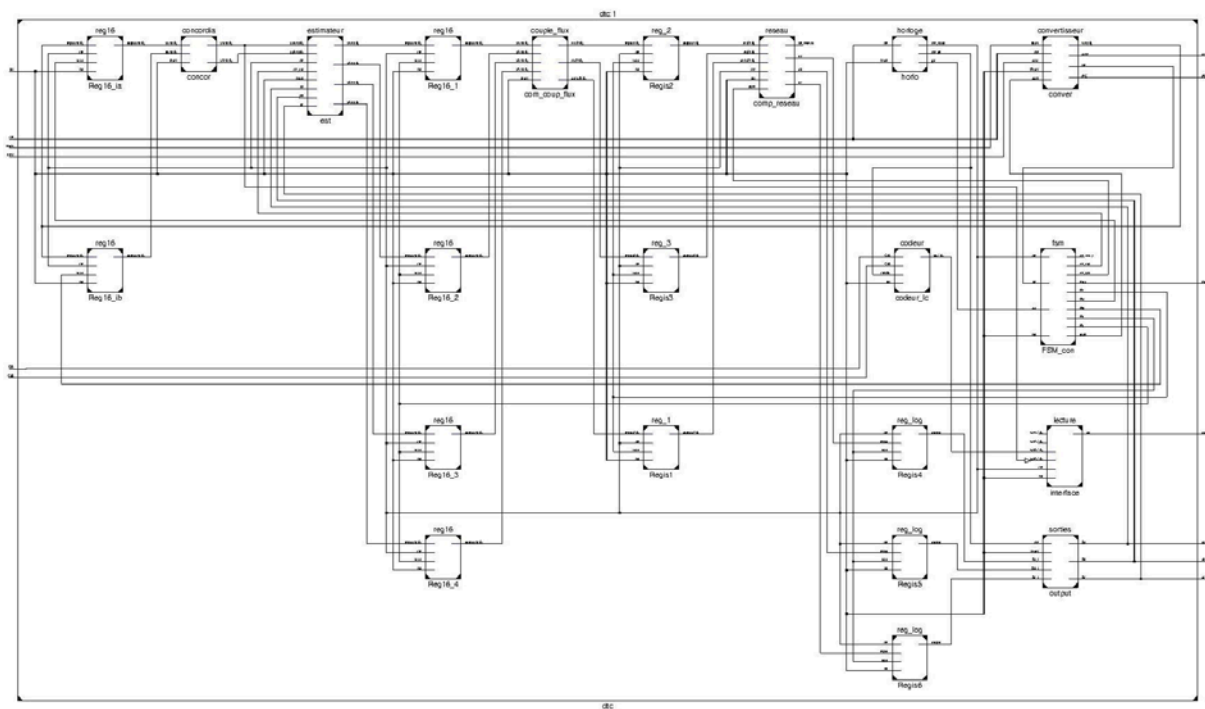


Figure 4.47 : Résultat de synthèse de la DTC neuronale (Top Level)

Parmi ces approximations on cite :

- Représentation des grandeurs réelles en virgule fixe sous le format de 16 bits,
- Introduire le temps mort entre deux interrupteurs sur un même bras de l'onduleur,
- Temps de retard des portes logique qu'on tient compte à partir de cette étape,
- Logiciel ModelSim n'est pas conçu pour la représentation analogique.

La figure 4.47 illustre le schéma RTL du résultat de synthèse de la Commande Directe Neuronale de Couple de la machine asynchrone, cependant, la figure 4.48 présente le schéma RTL du résultat de synthèse du module réseau de neurones. Il est obtenu à partir de l'outil *ISE Xilinx Design 12*, dans le but d'implémenter l'algorithme de commande Neuronale sur FPGA.

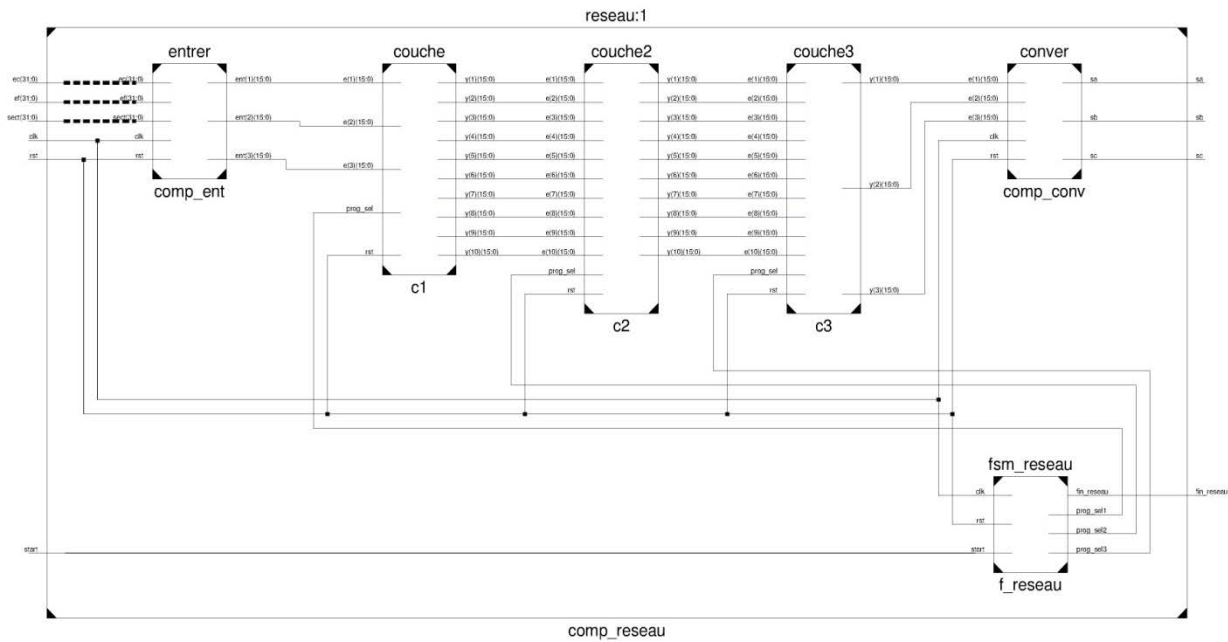


Figure 4.48: Résultat de synthèse de la DTC neuronale du module réseau de neurones

Le tableau 4.5 présente les performances en termes de consommation de ressources, obtenues lors de l'implémentation de l'architecture de la Commande Directe Neuronale de Couple sur l'FPGA VIRTEX 5 donnée par la figure 4.45. Les ressources consommées sont obtenues pour un format en virgule fixe de 16 bits.

Tableau 4.5 : Ressources utilisées sur le FPGA par l'algorithme de commande Neuronale.

Ressources	utilisées	Accessibles
Les pins E/S	13	640
Nombre de buffer global	7	32
Nombre de slices LUT	13939	44800
Nombre de slices registres	805	44800
Nombre de Latch	665	14079
Nombre de DSP48Es	118	128

La fréquence maximale de l'horloge est fixée par l'outil de synthèse égale à 35.539MHz, qui correspond à une période minimale de 28.138ns. La figure 4.49 présente le résultat de routage de l'architecture de la commande neuronale de la machine asynchrone sur FPGA Virtex 5.

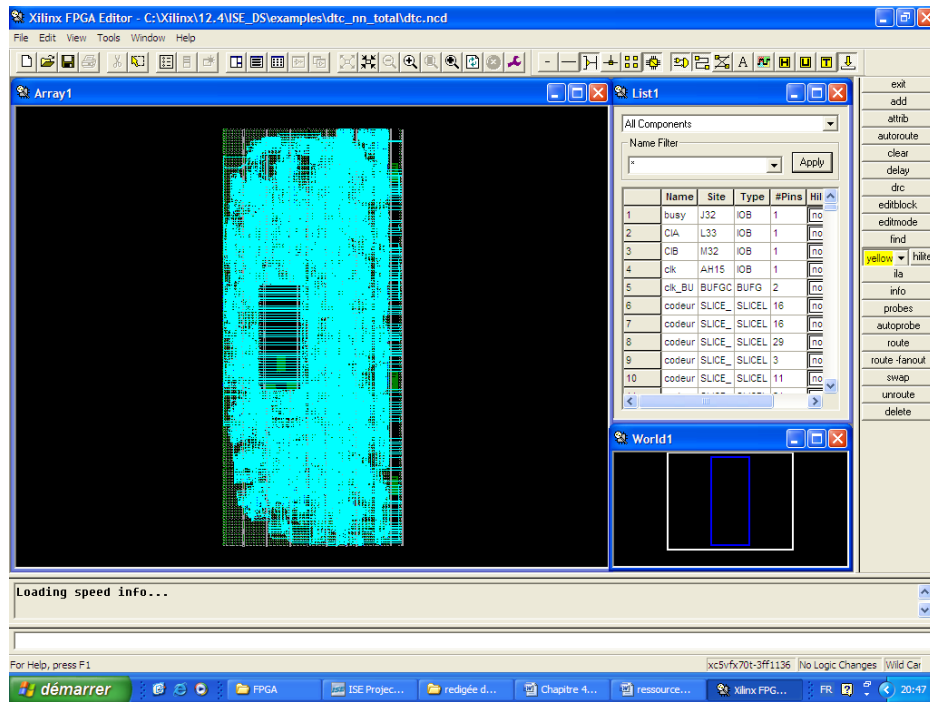


Figure 4.49. Le résultat de routage de l'architecture de la commande Neuronale

## 4.5 Conclusion

Dans la première partie de ce chapitre on a traité l'environnement et la méthodologie de conception utilisée pour implémenter la Commande DTC conventionnelle d'une machine asynchrone. On a présenté en premier lieu le flot de conception adopté, ensuite la modélisation et la spécification fonctionnelle de haut niveau de la chaîne de commande. En deuxième lieu, on a présenté la méthodologie suivie et les outils utilisés lors de l'implémentation de la commande DTC d'une machine asynchrone à base d'un module matériel pour cible FPGA. En outre, une description en VHDL de tous les modules a été présentée. On a comparé les résultats de simulation obtenus sous VHDL avec ceux de la commande DTC conventionnelle. Enfin avec l'outil XSE de ISE Xilinx, on a appliqué tout le flot de conception de la phase de synthèse jusqu'à la phase de génération du *bitstream*. La deuxième partie concerne l'implémentation de toute la commande directe floue de couple de la machine asynchrone sur FPGA. Une étude détaillée sur la description matérielle sous VHDL des modules qui composent le système d'inférence flou a été présentée et discutée. Lors de la dernière partie, on a présenté la description en VHDL de tous les modules, ensuite l'implémentation de la Commande Directe Neuronale de Couple de la machine asynchrone sur une plate-forme à base d'un module matériel FPGA. Enfin avec l'outil XSE de ISE Xilinx, on a présenté les résultats de synthèse.

Le dernier chapitre est consacré à la validation expérimentale des lois de commandes théoriques élaborées dans les chapitres précédents.

## **Chapitre 5 : Résultats expérimentaux de la commande DTC intelligente**



## 5.1 Introduction

La validation par simulation d'un algorithme permet de donner une idée sur les performances attendues. Dans des fois, ces simulations ne peuvent pas reproduire tous les phénomènes physiques agissant sur la chaîne de commande, car il est difficile de les modéliser et de les contrôler. De plus, les contraintes technologiques ne sont pas prises en considération lors de la simulation comme le temps de traitement de données, les retards, les erreurs de mesures dues aux capteurs, les bruits de mesure, etc.

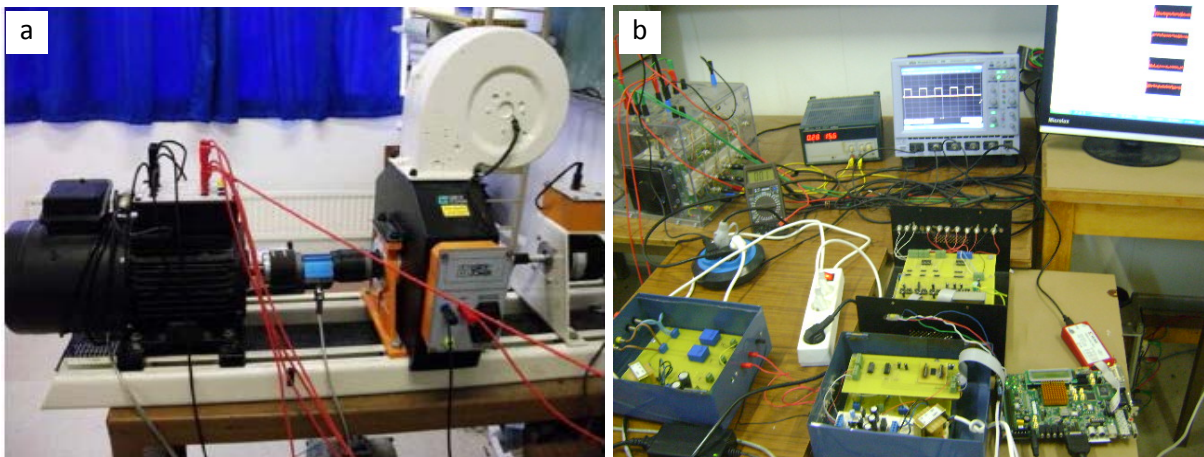
Dans ce contexte, viennent les travaux développés dans ce chapitre qui porte sur l'implémentation expérimentale sur FPGA des stratégies de commande et d'observation développées dans les chapitres précédents. Ce chapitre se divise en deux parties : dans la première on va décrire le banc d'essai expérimental mis en place dans l'unité de recherche ESIER (Etude des Systèmes Industriels et Energies Renouvelables) à l'ENIM. Ce banc d'essai sert à la validation expérimentale des stratégies de commande de la machine asynchrone sur FPGA. Dans la deuxième partie, on va présenter et discuter les résultats expérimentaux relatifs à l'implémentation des stratégies de commande suivantes :

- La Commande DTC conventionnelle utilisant un estimateur du flux statorique,
- La Commande DTC conventionnelle utilisant un observateur à mode glissant du flux statorique,
- La Commande DTFC utilisant un estimateur du flux statorique.

## 5.2 Présentation du banc d'essai expérimental

Afin de valider expérimentalement les stratégies de commande mises en œuvre, on a réalisé un banc d'essais à base de FPGA. Une photo du banc expérimental est donnée par la figure 5.1. Le schéma synoptique de ce banc est illustré par la figure 5.2. Il est constitué par les circuits suivants :

- Le circuit de puissance,
- Un système de mesure et d'acquisition,
- Le circuit de commande,
- Un système interface de commande de l'onduleur.



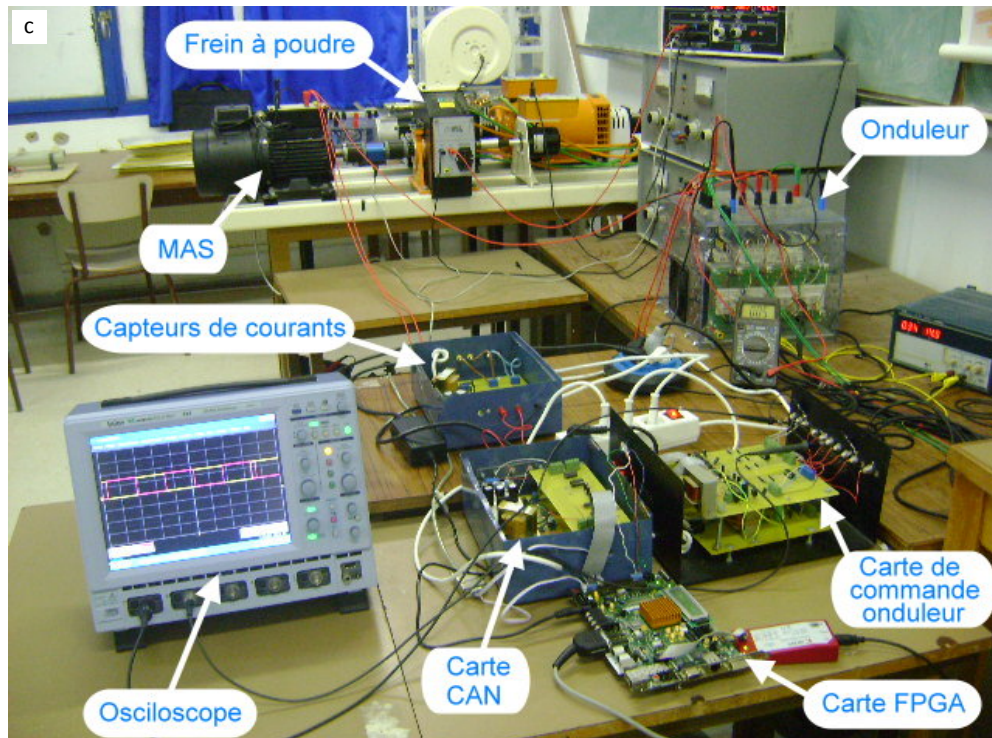


Figure 5.1 Photo du banc d'essai expérimental

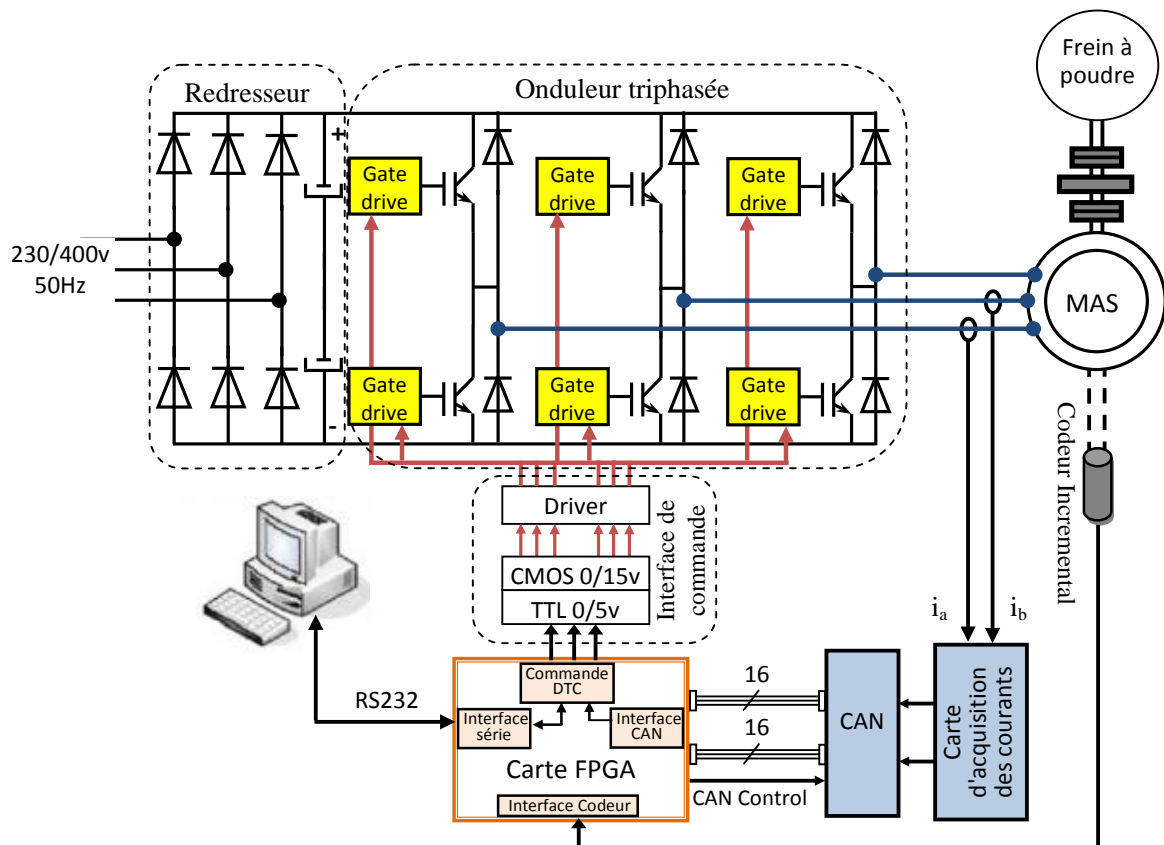


Figure 5.2. Schéma synoptique du banc d'essai expérimental

### 5.2.1 Le circuit de puissance

Le circuit de puissance de ce banc d'essai est constitué des éléments suivants :

#### 5.2.1.1 Source électrique

Cette source comporte un réseau triphasé (380V/50Hz) et un autotransformateur triphasé. La présence de ce dernier permet de générer une source de tension triphasée d'amplitude réglable.

#### 5.2.1.2 Convertisseur commandé

Le convertisseur utilisé est de type SEMIKRON. Il est composé d'un redresseur et d'un onduleur de tension triphasé. Le rôle du redresseur consiste à transformer une tension alternative triphasée de fréquence et d'amplitude constante, en une tension continue constante. Ce redresseur est compatible avec le réseau triphasé 380V et délivre une tension continue allant jusqu'à 537V. L'onduleur de tension triphasé permet de transformer une tension continue en une tension alternative de fréquence et d'amplitude variables.

L'architecture de l'onduleur triphasé illustrée par la figure 2.2 du deuxième chapitre, se compose de trois bras. Chaque bras comporte deux interrupteurs de puissance IGBT de la firme SIEMENS, de courant nominal 50A et d'une puissance nominale apparente de l'ordre de 20KVA.

La fréquence de commutation de l'onduleur est déterminée par le temps d'ouverture et de fermeture des interrupteurs et par le temps mort. Ce dernier est le temps introduit entre la fermeture d'un interrupteur et l'ouverture du même bras, afin de prévenir les risques de court-circuit. Pour ce type d'onduleur, la fréquence maximale de commutation est fixée à 20 KHz et le temps mort est de 3  $\mu$ s. Une photo de cet onduleur est illustrée dans la figure 5.1.c.

#### 5.2.1.3 Machine asynchrone

La machine asynchrone du banc d'essai est de puissance nominale 1.5 KW. Les caractéristiques et les paramètres électriques et mécaniques de cette machine sont donnés dans le tableau 2.2. La figure 5.1.a présente une photo de la machine asynchrone couplée à un frein à poudre qui permet de générer un couple charge.

### 5.2.2 Système de mesure et d'acquisition

La commande DTC de la machine asynchrone nécessite l'acquisition des grandeurs électriques alimentant la machine. L'acquisition de  $i_A$  et  $i_B$  est suffisante pour réussir la commande. La chaîne de mesure et d'acquisition numérique utilisée est illustrée par la figure 5.3.



Figure 5.3. Chaîne de mesure et d'acquisition

La chaîne de mesure et d'acquisition permet d'acquérir les deux courants de ligne ( $i_A$  et  $i_B$ ) et de les envoyer après traitement sous un format binaire vers le système de commande. Cette chaîne est constituée de cinq modules :

- Capteur de courants : c'est un dispositif permettant de générer en sortie des images analogiques instantanées des courants de la machine asynchrone. On a choisi d'utiliser les capteurs de courant à effet hall de type LEM : LA25-NP, vu que leur mise en œuvre est aisée, avec une bonne précision et une très faible sensibilité aux perturbations extérieures,
- Un conditionneur : c'est un montage électronique permettant de convertir la sortie du capteur en une tension,
- Un amplificateur : c'est un montage électronique à base d'un amplificateur opérationnel permettant d'amplifier le signal,
- Un filtre analogique : c'est un quadripôle qui réalise une fonction de filtrage en transmettant de manière sélectif une bande de fréquence,
- Un convertisseur analogique numérique à pour objectif de convertir les courant statoriques analogiques issus de la carte d'acquisition en des grandeurs numériques. La conversion analogique numérique est assurée par un seul convertisseur et un multiplexeur analogique afin de sélectionner une entrée parmi les deux courants statoriques. Le convertisseur est ADS8509 caractérisé par une résolution de 16 bits séries et un temps de conversion de 2.2  $\mu$ s. L'exploitation de ce convertisseur nécessite la génération des signaux de contrôle. Pour générer ces signaux, on a utilisé une description VHDL implémentée sur la carte de commande à base d'un FPGA.

Le principe de fonctionnement du convertisseur analogique numérique ainsi que les montages électroniques de tous ces modules sont présentés en annexe B.

### 5.2.3 Partie commande

#### 5.2.3.1 La carte de commande

La carte de commande à base de FPGA utilisée dans ce banc d'essai est la plateforme ML507 de la firme Xilinx. Cette carte est utilisée comme support physique pour l'implémentation des différentes stratégies de commande dédiées à l'entraînement électrique. La figure 5.4 présente la plateforme cible ML507 à base de l'FPGA Virtex 5 XC5VFX70T. Ce dernier est composé de 11200 slices liées entre elles par des connexions programmables et des blocs RAM internes de taille totale 5328 Kb. Il contient aussi 128 blocs DSP48E. Chacun de ces blocs comporte un multiplieur, un accumulateur et un additionneur/soustracteur. La carte ML507 contient un oscillateur interne qui permet de générer une horloge de fréquence 100 MHz. Elle possède aussi plusieurs supports de communication avec des dispositifs externes à travers le port USB ou par une liaison série RS232. Les ports d'entrées/sorties de cette plateforme possèdent un niveau logique haut de 3.3V.



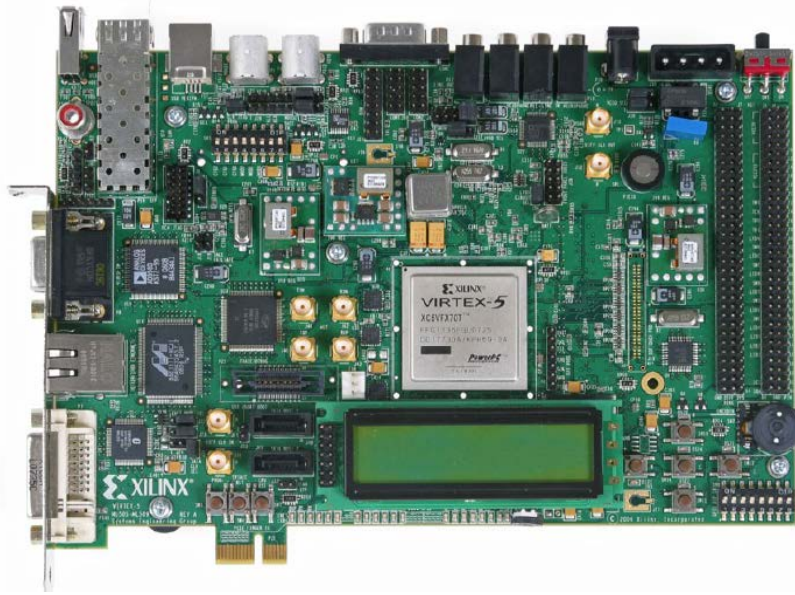


Figure 5.4. La carte de commande à base de FPGA : plateforme ML507.

### 5.2.3.2 L'architecture de commande

L'architecture de commande implémentée sur cible FPGA, est présentée par la figure 5.2. Cette architecture contient les quatre modules suivants :

- **Interface CAN :** Ce module a pour fonction de contrôler la conversion analogique numérique. Ce contrôle est assuré par l'envoi d'une horloge de lecture de données, d'un signal de début de conversion et la réception d'un signal de fin de conversion. Les données sous format binaire seront accessibles en série sur chaque front montant de l'horloge. Ce module permet ensuite de stocker les données reçues dans des registres à décalages et de les transmettre au module commande DTC. Le détail des opérations de contrôle et de lecture du processus de conversion analogique et numérique est expliqué en annexe B. La figure 5.5 illustre les différents signaux de communication et du contrôle du processus de conversion.

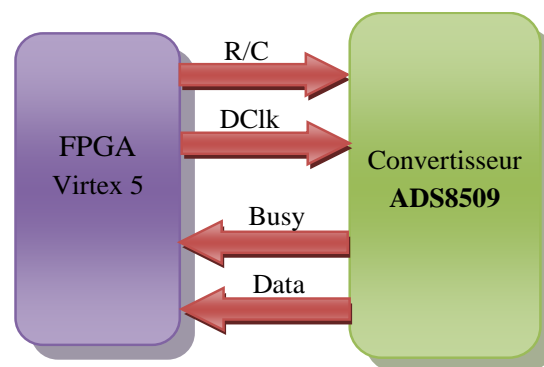


Figure 5.5. Contrôle et acquisition de données du convertisseur

- **Interface série :** ce module assure la communication série entre l'ordinateur et la carte FPGA à travers le port RS232. Au niveau de l'FPGA, les valeurs instantanées de couple électromagnétique, le module du flux statorique, la vitesse rotorique et la composante du courant  $i_{sq}$  seront envoyées à l'ordinateur à travers le protocole de communication série RS232. Au niveau de l'ordinateur, une interface graphique de visualisation en temps réel de

données est développée sous Visual-Basic, assure la lecture de ces données issues du port série. Cette interface présentée par la figure 5.6, permet une visualisation graphique des données reçues ainsi que la sauvegarde sur le disque dur pour une utilisation ultérieure.

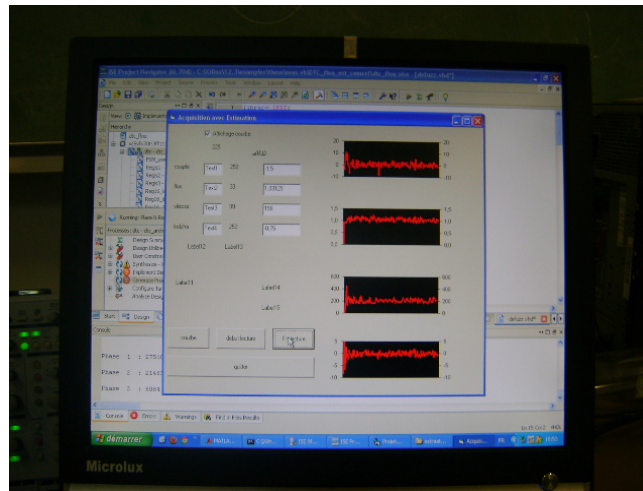


Figure 5.6. Interface graphique de visualisation de données

- **Interface codeur** : ce module assure la lecture de données issues du codeur incrémental monté sur l'arbre de la machine asynchrone, afin de déterminer la vitesse rotorique. Ce codeur incrémental est équipé de 1024 fentes et génère à sa sortie trois signaux :  $U_A$  et  $U_B$  qui sont en quadrature de phase et  $U_{A0}$  qui fournit une impulsion à chaque tour mécanique. Le déphasage de  $90^\circ$  électrique des signaux  $U_A$  et  $U_B$  permet de déterminer le sens de rotation.
- **Commande DTC** : c'est le module principal de la commande. Ce module reçoit en entrée les mesures instantanées de deux courants  $i_A$  et  $i_B$  sous format binaire. Ensuite, il détermine les signaux de commande de l'onduleur  $S_a$ ,  $S_b$  et  $S_c$  afin d'appliquer une tension aux bornes de la machine asynchrone. Cette stratégie permet de commander la machine asynchrone en couple. Le principe de cette stratégie ainsi que les équations de commande ont été déjà développé dans le deuxième chapitre.

#### 5.2.4 Interface de commande

Cette partie nécessite une maquette de commande de l'onduleur qui doit assurer :

- L'isolation galvanique entre les signaux de commandes générées par la carte de commande à base de FPGA et l'onduleur,
- Amplifier les signaux issus de la carte de génération des signaux de commande, car les signaux issus de la carte FPGA sont de niveau 3.3V, alors que les signaux de commande des interrupteurs de l'onduleur doivent être de 15V,
- A partir des trois signaux, on crée des temps morts fixes entre les commandes complémentaires d'un même bras d'onduleur,
- L'isolation optique de l'onduleur et la carte de commande rapprochée.

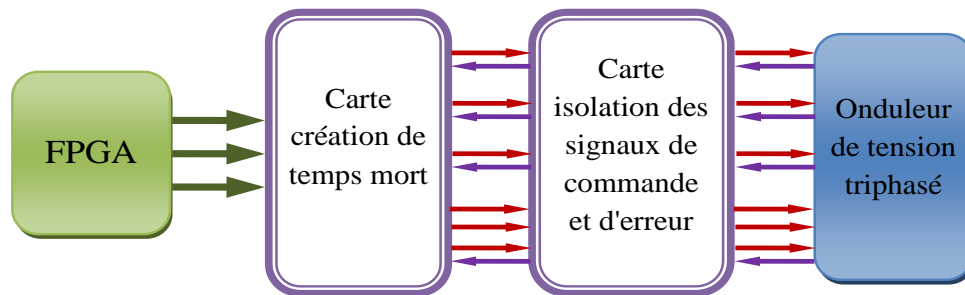


Figure 5.7 : Schéma synoptique de la carte de commande rapprochée.

Cette interface de commande de l'onduleur illustrée par la figure 5.7 est constituée de deux cartes : la carte de création de temps mort et la carte d'amplification et d'isolation des signaux de commande et d'erreurs.

#### 5.2.4.1 Carte création de temps mort

Dans le cas réel, le transistor IGBT possède un temps de retard à la fermeture et un temps de retard à l'ouverture. Pour éviter la possibilité de la conduction simultanée des deux interrupteurs d'un même bras d'onduleur, on doit retarder la conduction des deux interrupteurs par l'insertion d'un temps mort entre chaque signal de commande et son complémentaire. Ce retard doit être supérieur au temps de retard à la fermeture et à l'ouverture de l'IGBT. La carte qu'on a conçue permet de générer à partir d'un signal, deux signaux décalés d'un temps mort de 3  $\mu$ s. Sur la figure 5.7, les signaux en vert sont les trois signaux logiques  $S_a$ ,  $S_b$  et  $S_c$  générés par l'FPGA, alors que les signaux en rouge sont les trois signaux  $S_a$ ,  $S_b$  et  $S_c$  et leurs complémentaires  $S'_a$ ,  $S'_b$  et  $S'_c$  décalés d'un temps mort. Une étude détaillée sur le développement de la carte électronique de création de temps mort est présenté dans l'annexe C.

#### 5.2.4.2 Carte d'amplification et d'isolation des signaux de commande et d'erreur

Les impulsions de commande générées par l'FPGA sont de niveau logique 0-3.3V, alors que la commande des interrupteurs de l'onduleur nécessite des signaux de niveau logique 0-15V. D'où cette carte doit assurer deux tâches :

- Une isolation galvanique entre la carte de commande FPGA et la carte de commande rapprochée de l'onduleur,
- Une amplification des signaux de commande de l'onduleur.

Ces deux tâches peuvent être réalisées par l'utilisation des optocoupleurs. Si les signaux logiques issus de la carte FPGA se connectent à l'entrée des optocoupleurs de type 6N136, et si ces optocoupleurs seront alimentés à 15V, on aura une amplification des signaux de commande ainsi que l'isolation optique entre la carte de commande FPGA et la carte de commande rapprochée de l'onduleur. La figure 5.8 illustre le principe de fonctionnement d'un optocoupleur.

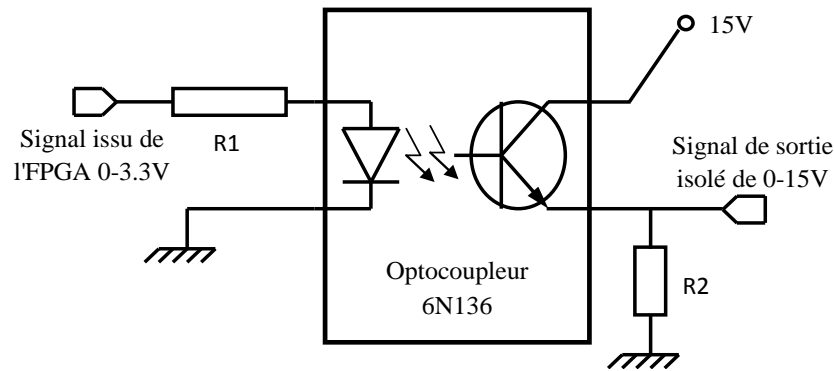


Figure 5.8. Schéma de principe de l'optocoupleur

L'onduleur possède un système de protection, il génère un signal de défaut dans le cas d'anomalie (surintensité, surtension, court-circuit ou bien une augmentation de température). Ces signaux ont été exploités dans la réalisation de cette carte, afin de désactiver immédiatement les signaux de commandes si un défaut est détecté. Sur la figure 5.7, les signaux en bleu représentent les signaux d'erreurs issus de l'onduleur. Le montage électronique de cette carte ainsi que son principe de fonctionnement sont présentés en détaille dans l'annexe C.

## 5.3 Etude expérimentale

### 5.3.1 Vérification des signaux de commande

Avant d'entamer la phase d'implémentation des stratégies de commande de la machine asynchrone, on procède à une phase de vérification des signaux de commande de l'onduleur issus de la carte de commande. Cette phase consiste à implémenter la Commande DTC sur FPGA, sans alimenter l'onduleur. D'où on peut voir le comportement de la partie commande sans la partie puissance. La figure 5.9 présente les deux signaux de commande des IGBT respectivement haut et bas d'un même bras d'onduleur. Le temps mort entre les deux signaux de commande est de  $3\mu\text{s}$ , ainsi que leurs amplitudes est de 15V.

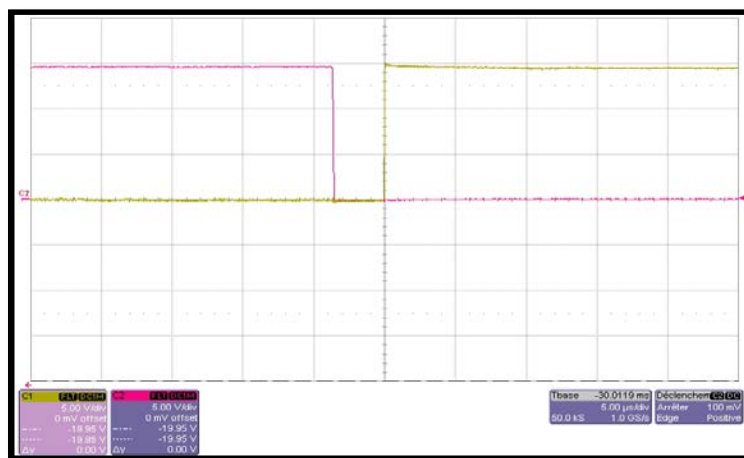


Figure 5.9. Les signaux de commande d'un même bras d'onduleur



La figure 5.10 présente les signaux de commande de deux bras d'onduleur. Chaque cadrant de cette figure montre les signaux de commande des interrupteurs IGBT respectivement haut et bas d'un même bras.

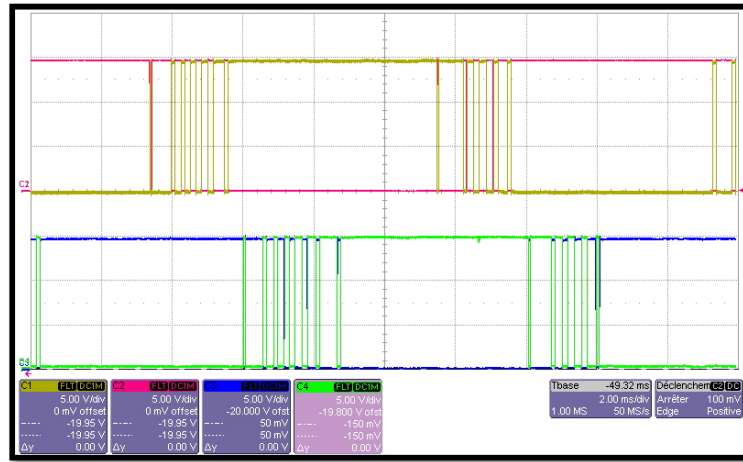


Figure 5.10. Les signaux de commande de deux bras d'onduleur

L'objectif de cette phase est de vérifier et valider le comportement de la partie commande. D'après les figures 5.9 et 5.10, on peut constater que les signaux de commande sont conformes à l'étude théorique.

### 5.3.2 Implémentation de la DTC

#### 5.3.2.1 Implémentation sur FPGA

La figure 5.11 présente le diagramme temporel de l'architecture de commande DTC qui décrit les différentes opérations nécessaires pour la génération des signaux de commande. A l'instant  $t_k$ , le début de la  $k^{\text{ème}}$  période d'échantillonnage, les deux courants statoriques  $i_{sa}$  et  $i_{sb}$  sont échantillonnés. Au même temps, le processus de conversion analogique numérique est activé. La commande DTC sera activée après un temps  $2 \cdot t_{ADC}$ , qui correspond au temps de conversion. Ces modules sont contrôlés par la machine d'états finis de la figure 4.8 du quatrième chapitre. Après un temps de calcul  $t_{DTC}$ , le module table de sélection de l'architecture de commande DTC, permet de générer les signaux de commande  $S_a$ ,  $S_b$  et  $S_c$  correspondants à la  $k^{\text{ème}}$  période d'échantillonnage. Ces signaux de commande de l'onduleur sont générés durant chaque période d'échantillonnage après un temps d'exécution  $T_{ex} = 2 \cdot t_{ADC} + t_{DTC}$ . Après une période d'échantillonnage  $T_e$ , un nouveau cycle de calcul correspondant à la  $(k+1)^{\text{ème}}$  période d'échantillonnage est activé.

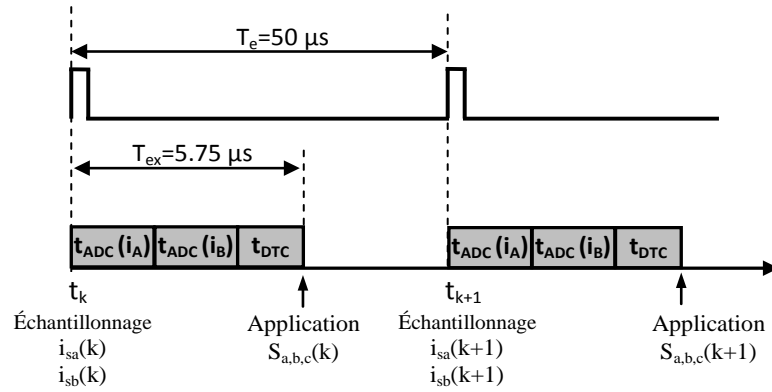


Figure 5.11. Diagramme temporel de l'architecture de commande DTC

Le tableau 5.1 présente les performances temporelles du temps de calcul obtenu lors de l'implémentation de l'architecture de commande de la DTC. Le temps de calcul de la commande DTC est de l'ordre de  $t_{DTC} = 1.35 \mu s$ . En ajoutant le temps de conversion  $2 * t_{ADC} = 4.4 \mu s$ , on obtient le temps d'exécution de l'architecture de commande DTC estimé à  $T_{ex} = 5.75 \mu s$ .

Tableau 5.1. Performances temporelles de l'architecture DTC

Module	cycles	Temps d'exécution
<b>Temps de conversion AN</b>	110	2.2 $\mu s$
<b>Concordia</b>	6	0.18 $\mu s$
<b>Estimateur Flux &amp; Couple</b>	15	0.45 $\mu s$
<b>Secteur et Hystérésis</b>	16	0.48 $\mu s$
<b>Table de sélection</b>	8	0.24 $\mu s$
<b><math>t_{DTC} = 0.18 + 0.45 + 0.48 + 0.24</math></b>		1.35 $\mu s$
<b><math>T_{ex} = t_{DTC} + 2 * t_{ADC} = 1.35 + 2 * 2.2</math></b>		5.75 $\mu s$

### 5.3.2.2 Résultats expérimentaux de la DTC à base d'un estimateur de flux

Les figures 5.12 et 5.13 montrent les résultats expérimentaux obtenus lors de l'implémentation de la stratégie de Commande DTC à base d'un estimateur de flux. Ces résultats sont obtenus pour une fréquence d'échantillonnage de 20 KHz. La figure 5.12 présente l'allure obtenue des courants statoriques  $i_A$  et  $i_B$ . Cette dernière figure, illustre l'apparition des ondulations du courant.

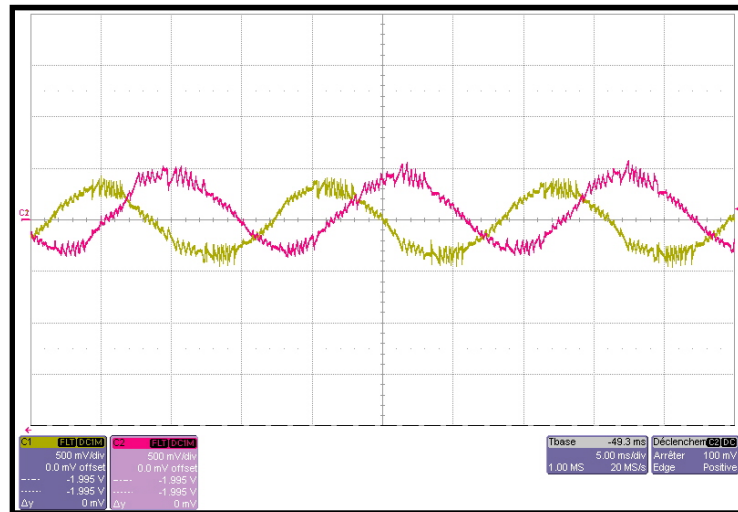


Figure 5.12. Les courants statoriques de la MAS

La figure 5.13.a illustre l'évolution du couple électromagnétique, suite à un couple de charge appliqué en régime permanent. Au démarrage, le couple est assez rapide, il atteint la valeur 9N.m. Ensuite, il se stabilise autour de la valeur faible puisque la machine fonctionne à vide. Au moment de l'application du couple de charge, le couple électromagnétique atteint sa valeur de référence 2.7 N.m. puis revient à la valeur initial du régime permanent une fois le couple de charge est nul. On peut observer sur cette figure des oscillations importantes au niveau de son amplitude. Ceci constitue l'inconvénient majeur de la commande DTC.

Le flux statorique est présenté par la figure 5.13.b. On peut remarquer que le flux statorique atteint immédiatement sa valeur de référence de 0.91 Wb. On observe aussi une légère ondulation du flux statorique autour de sa valeur de référence.

La figure 5.13.c présente l'évolution temporelle de la vitesse de rotation. La montée en vitesse est rapide au démarrage, la vitesse atteinte est proche de 300 rd/s. Au moment de l'application du couple de charge, on observe une diminution de la vitesse allant jusqu'à 250 rd/s, ceci est dû au fait qu'il n'y pas de régulation.

Le courant statorique  $i_{\text{salpha}}$  présenté par la figure 5.13.d est caractérisé par un fort courant de démarrage allant jusqu'à 10A, puis il se stabilise à sa valeur nominale de magnétisation de 1.2A, comme illustrée par la figure 5.13.e. Au moment de l'application du couple de charge, le courant  $i_{\text{salpha}}$  atteint la valeur 3A en module qui correspond à la valeur du courant de charge. Ceci est illustré par la figure 5.13.f.

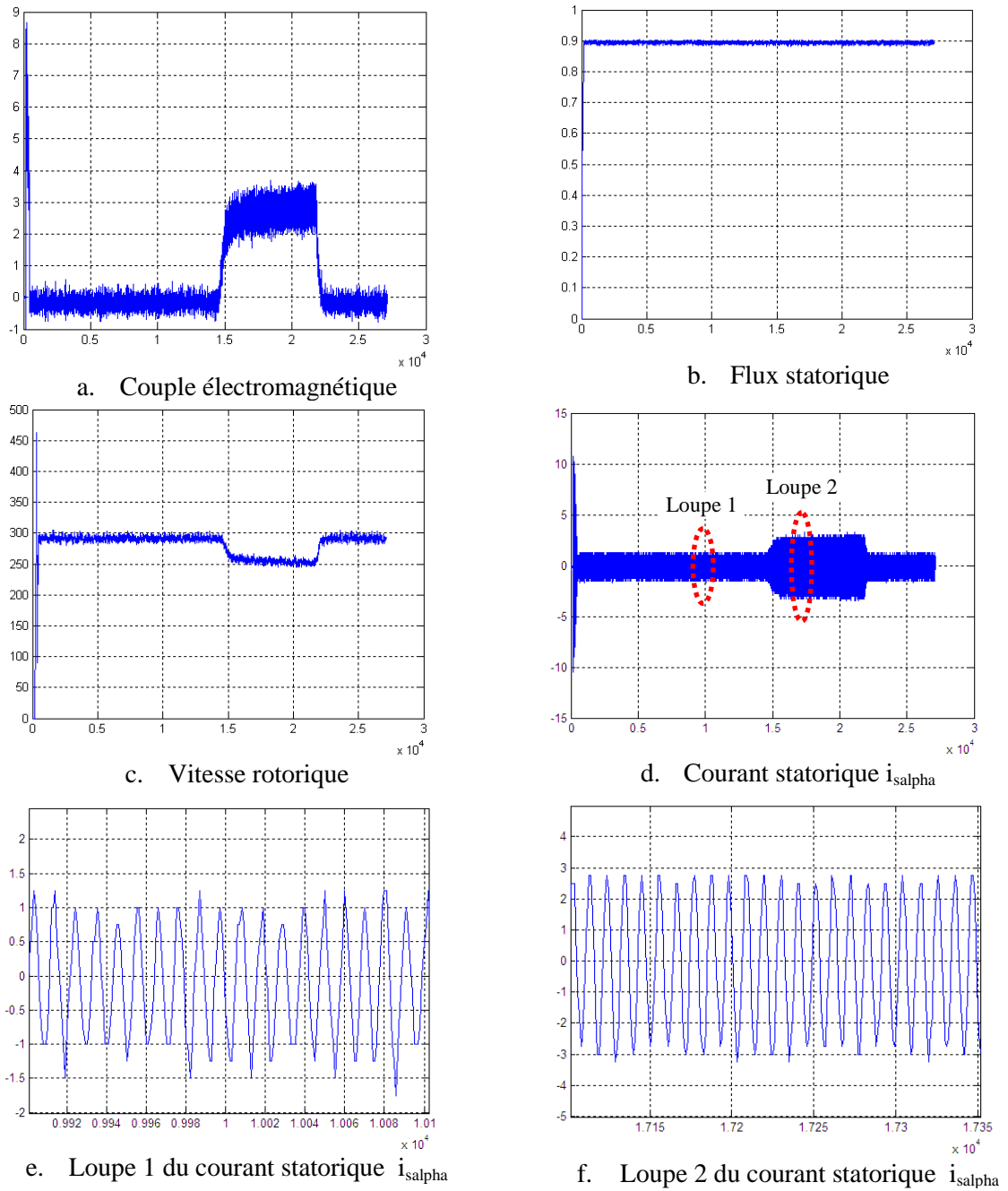


Figure 5.13 Résultats expérimentaux de la commande DTC à base d'un estimateur

Les résultats expérimentaux ont révélé une dynamique acceptable du flux statorique, du couple électromagnétique, de la vitesse rotorique et du courant statorique, avec l'apparition des ondulations en régime permanent. En comparant les résultats expérimentaux de la figure 5.13 et les résultats de simulations déjà développés dans le chapitre deux, on peut remarquer une conformité satisfaisante de résultats, ce qui prouve la validité du modèle proposé.

### 5.3.2.3 Résultats expérimentaux de la DTC à base d'un observateur de flux

La figure 5.14 donne les résultats expérimentaux obtenus lors de l'implémentation de la stratégie de Commande DTC à base d'un observateur à mode glissant. Ces résultats expérimentaux sont obtenus pour une fréquence d'échantillonnage de 20 KHz.

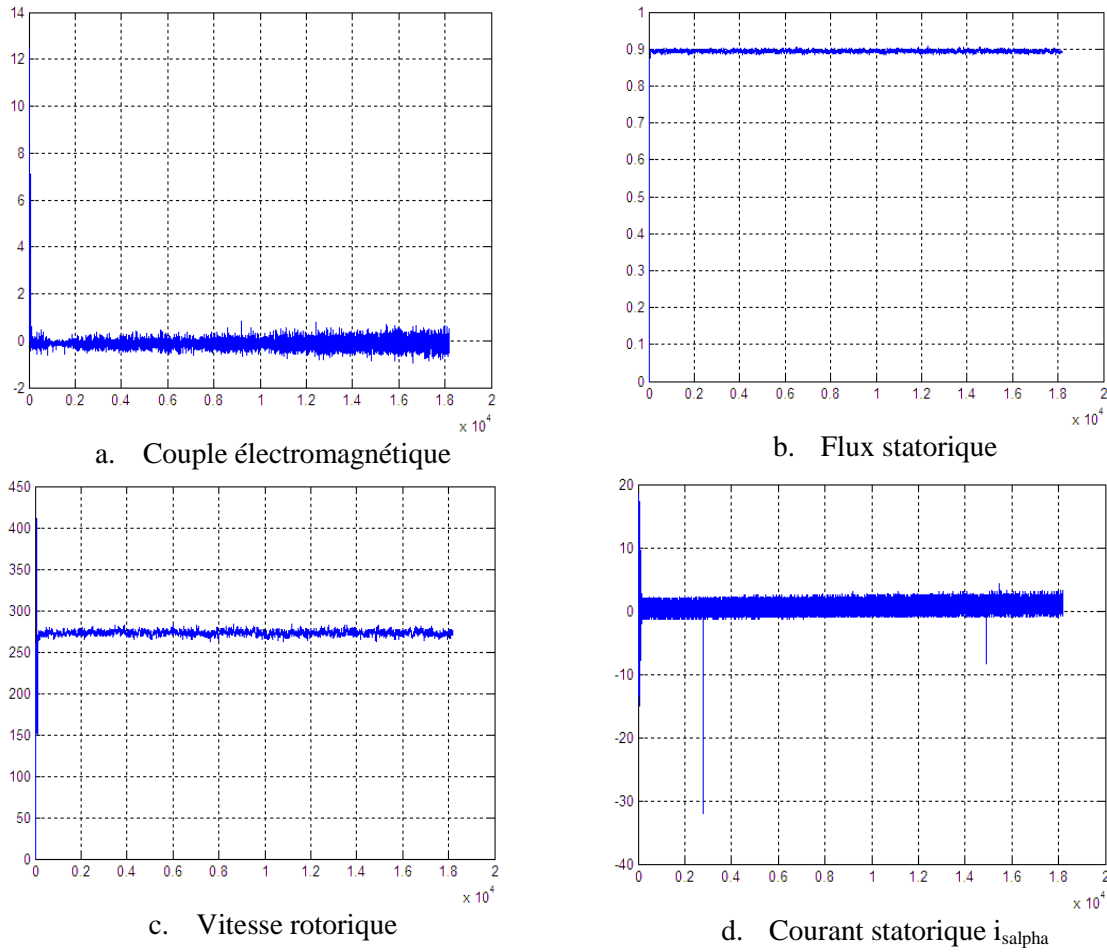


Figure 5.14 Résultats expérimentaux de la commande DTC à base d'un observateur de flux

La figure 5.14.a illustre l'évolution du couple électromagnétique. Au démarrage, le couple est assez rapide, il atteint la valeur 12 N.m. Ensuite, il se stabilise autour de la valeur nulle puisque la machine fonctionne à vide. On peut observer aussi la présence des oscillations importantes au niveau de son amplitude. Le flux statorique présenté sur la figure 5.14.b, atteint immédiatement sa valeur de référence de 0.91 Wb. Sur cette figure, on enregistre une légère ondulation du flux statorique autour de sa valeur de référence. La figure 5.14.c présente l'évolution temporelle de la vitesse de rotation. La montée en vitesse est rapide au démarrage, la vitesse atteinte est proche de 275 rd/s. Quand à la figure 5.14.d, elle illustre l'évolution du courant statorique  $i_{salpna}$ . Sur cette figure, on peut observer le fort courant de démarrage, ensuite il se stabilise à sa valeur nominale de magnétisation de 1.5 A.

### 5.3.3 Implémentation de la DTFC

#### 5.3.3.1 Implémentation sur FPGA

Dans le troisième chapitre de ce rapport, on a présenté l'apport des techniques de l'intelligence artificielle sur la commande DTC. C'est pour cela on choisit dans ce paragraphe d'implémenter la DTFC sur FPGA [SoGd13a].

La figure 5.15 présente le diagramme temporel de l'architecture de Commande Direct Floue de Couple qui décrit les différentes opérations nécessaires pour la génération des signaux de commande. A l'instant  $t_k$ , le début de la  $k^{\text{ème}}$  période d'échantillonnage, les deux courants

statoriques  $i_{sa}$  et  $i_{sb}$  sont échantillonnés. Au même temps, le processus de conversion analogique numérique est activé. La commande DTFC sera activée après un temps  $2*t_{ADC}$ , qui correspond au temps de conversion. Ces modules sont contrôlés par la machine d'états finis de la figure 4.23 du quatrième chapitre. Après un temps de calcul  $t_{DTFC}$ , le module système d'inférence flou de l'architecture de commande DTFC, permet de générer les signaux de commande  $S_a$ ,  $S_b$  et  $S_c$  correspondants à la  $k^{ème}$  période d'échantillonnage. Ces signaux de commande de l'onduleur sont générés dans chaque période d'échantillonnage après un temps d'exécution  $T_{ex} = 2*t_{ADC} + t_{DTFC}$ . Après une période d'échantillonnage  $T_e$ , un nouveau cycle de calcul correspondant à la  $(k+1)^{ème}$  période est activé.

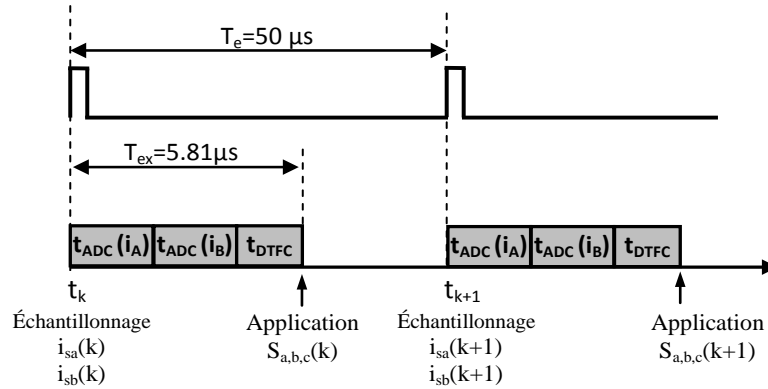


Figure 5.15. Diagramme temporel de l'architecture de commande DTFC

Le tableau 5.2 présente les performances temporelles du temps de calcul obtenu lors de l'implémentation de l'architecture de commande de la DTFC. Le temps de calcul de la commande DTFC est de l'ordre de  $t_{DTFC} = 1.41 \mu s$ . En lui ajoutant le temps de conversion  $2*t_{ADC} = 4.4 \mu s$ , on obtient le temps d'exécution de l'architecture de commande DTFC estimé à  $T_{ex} = 5.81 \mu s$ .

Tableau 5.2. Performances temporelles de l'architecture DTFC

Module	cycles	Temps d'exécution
Temps de conversion AN	110	2.2 us
Concordia	6	0.18 us
Estimation du Flux & Couple	15	0.45 us
Secteur	16	0.48 us
Système d'inférence flou	20	0.6 us
$t_{DTFC} = 0.18 + 0.45 + 0.48 + 0.6$		1.41 us
$T_{ex} = t_{DTFC} + 2*t_{ADC} = 1.41 + 2* 2.2$		5.81 us

### 5.3.3.2 Résultats expérimentaux de la DTFC

La figure 5.16 présente les résultats expérimentaux obtenus lors de l'implémentation de la stratégie de Commande Directe Floue de Couple.

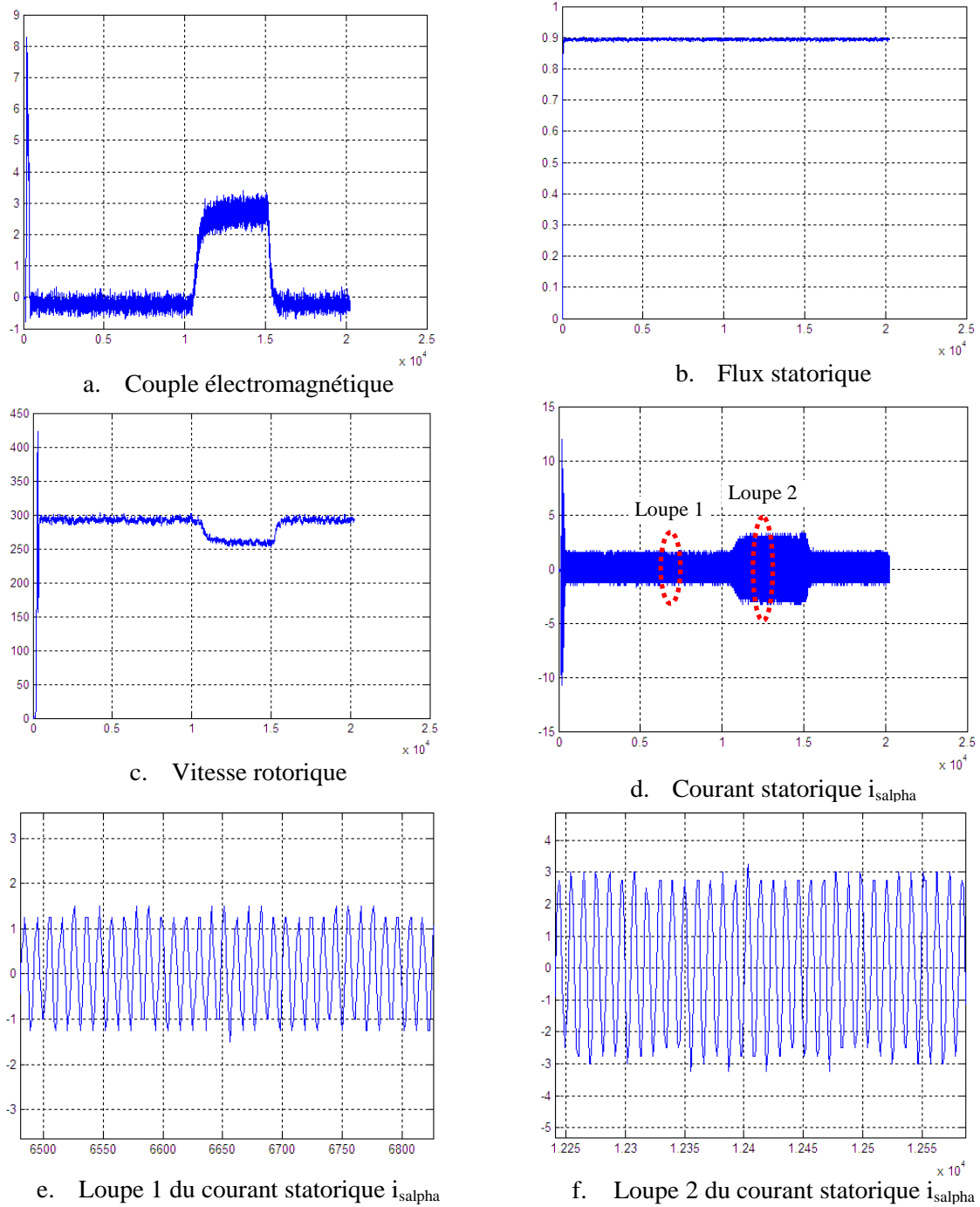


Figure 5.16. Résultats expérimentaux de la commande DTFC

Ces résultats sont obtenus pour une fréquence d'échantillonnage de 20 KHz. Afin de les comparer avec ceux obtenus dans la figure 5.13, on va choisir le même profil de fonctionnement de la MAS. C'est-à-dire, on va démarrer la MAS à vide, ensuite on va appliquer à partir du régime permanent un couple de charge. La figure 5.16.a présente l'évolution du couple électromagnétique suite à un couple de charge. Au démarrage, le couple atteint la valeur 8.8N.m. Ensuite, il devient proche de zéro correspond au fonctionnement à vide. Lorsqu'on applique un couple de charge, le couple atteint la valeur de référence 2.7 N.m puis il revient à la valeur initiale du régime permanent. Par comparaison avec la figure 5.13.a, on peut observer une réduction importante des ondulations du couple de la machine

asynchrone avec la commande DTFC. Comme on peut observer aussi qu'au démarrage, le couple de la DTFC atteint la valeur maximale avant celui de la DTC.

Le flux statorique est illustré par la figure 5.16.b, où il atteint au démarrage de la MAS sa valeur de référence de 0.91 Wb. En termes de comparaison avec la figure 5.13.b, on remarque une diminution des ondulations du flux statorique avec la commande DTFC et un établissement plus rapide au démarrage que celui de la DTC conventionnelle.

La figure 5.16.c présente l'évolution temporelle de la vitesse de rotation. La montée en vitesse est rapide au démarrage, la vitesse atteinte est proche de 300 rd/s. Au moment de l'application du couple de charge, on observe une diminution de la vitesse allant jusqu'à 250 rd/s.

Le courant statorique  $i_{s\alpha}$  présenté par la figure 5.16.d est caractérisé par un fort courant de démarrage allant jusqu'à 12A, puis il se stabilise à sa valeur nominale de magnétisation de 1.5A, comme illustrée par la figure 5.16.e. Au moment de l'application du couple de charge, le courant  $i_{s\alpha}$  atteint la valeur 3A en module qui correspond à la valeur du courant de charge.

Les résultats expérimentaux de la commande DTFC montrent de meilleures performances que celles obtenues par la Commande DTC. Il est intéressant de remarquer que la commande floue se distingue par une importante réduction des ondulations de couple et du flux par rapport à la commande conventionnelle. On enregistre aussi, une bonne réponse dynamique du couple comme la DTC conventionnelle. Ceci prouve la justesse des résultats théoriques introduits dans le troisième chapitre.

## 5.4 Conclusion

Ce chapitre a été consacré essentiellement à l'implémentation expérimentale des stratégies de commande de la machine asynchrone développées dans cette thèse. En premier lieu, on a décrit les différentes parties du banc d'essai réalisé dans l'unité de recherche ESIER à l'ENIM. Ce banc d'essai est réalisé autour d'une plateforme ML507 à base d'un FPGA Virtex5. Une carte d'acquisition et de mesure pour la mesure des courants statoriques ainsi qu'une carte de commande des interrupteurs IGBTs de l'onduleur triphasé sont réalisés. Ces cartes constituant le banc d'essai sont protégées contre les surtensions et les surintensités. En deuxième lieu, on a présenté et discuté les résultats expérimentaux relatifs à l'implémentation des stratégies de commande conventionnelle et intelligente. Les résultats expérimentaux de la commande DTC floue montrent bien des améliorations remarquables au niveau des ondulations du couple et du flux.

Enfin, disons en général que les premiers résultats expérimentaux sont intéressants et méritent d'être mieux investigué dans le futur en abordant d'autres régimes de fonctionnement à basses vitesses et un régime de défluxage.



## **Conclusion générale**

## **Conclusion générale**

Dans ce travail, on a instauré de nouvelles solutions technologiques dans la commande des machines électriques. Plus particulièrement, on a intégré les techniques de l'intelligence artificielle, tels que la logique floue et le réseau de neurones dans l'élaboration de la commande DTC de la machine asynchrone. Ensuite, on a présenté une démarche méthodologique pour l'implémentation matérielle de cette commande sur une cible FPGA.

Les principaux volets de ce travail sont résumés comme suit :

Dans une première phase, on a présenté en premier lieu, un état de l'art sur l'ensemble des travaux de recherche menés autour de la thématique afférente à la commande DTC à fréquence contrôlée et non contrôlée. En deuxième lieu, on a présenté un état de l'art sur l'implémentation matérielle des algorithmes de commande des machines asynchrones. Ensuite, on a discuté et présenté l'environnement et la démarche pour la description, simulation et la synthèse afin de réussir l'implémentation de la commande de machines asynchrones sur cible FPGA.

La deuxième phase de ce travail est composée de trois parties. Dans la première, on a commencé par présenter brièvement le modèle électrique en régime dynamique de la machine asynchrone. Ensuite, on a mis en œuvre une stratégie de commande DTC basée sur le principe de découplage entre le couple et le flux. Dans la deuxième partie de ce chapitre, deux approches d'observation du flux statorique de la machine asynchrone ont été développées à savoir : l'observateur à grand gain et l'observateur par mode glissant. Ces deux approches répondent principalement aux besoins les plus critiques des lois de commande de la MAS en matière de robustesse vis-à-vis les variations paramétriques de la machine et assurent un bon fonctionnement sur toute la plage de vitesse, particulièrement à basses vitesses. Dans la troisième partie, une étude comparative des résultats est menée, afin d'explorer les performances de la commande DTC avec observateur et estimateur du flux statorique d'une machine asynchrone alimentée via un onduleur de tension. On retient que l'observateur par mode glissant a donné les meilleures performances en termes d'observation de flux statorique et de robustesse. Ce dernier a été adopté dans la phase d'implémentation de la commande DTC sur FPGA. Il faut signaler que malgré les performances dynamiques enregistrées au niveau des grandeurs couple électromagnétique et flux statorique, la commande DTC conventionnelle reste marquée par deux inconvénients majeurs liés à l'existence des ondulations importantes ainsi qu'à la variation de la fréquence de commutation introduite par l'utilisation des comparateurs à hystérésis.

Dans la troisième phase, on a commencé par montrer l'apport de l'intelligence artificielle dans la Commande DTC de la machine asynchrone. Dans ce cadre, on a présenté en premier lieu les principes de base de la logique floue et la manière de sa mise en œuvre. Ensuite on a développé la commande directe floue de couple (DTFC) de la machine asynchrone. En deuxième lieu, on a présenté une description théorique de réseaux de neurones artificiels et son utilisation dans la mise en place de la commande directe neuronale de couple (DTNC) de

la machine asynchrone. Cette phase est clôturée par une étude comparative entre la commande DTC, la commande DTFC et la commande DTNC. Les résultats de la commande DTC intelligente montrent des améliorations remarquables au niveau des ondulations de couple et de flux. Ces améliorations sont marquées par une diminution des ondulations et une rapidité au niveau de la dynamique de couple et de flux pendant la phase de démarrage de la machine. Il faut retenir que la commande floue a apporté plus d'amélioration que la commande neuronale. Cela est confirmé par la diminution des harmoniques de couple électromagnétique, de flux statorique et du courant statorique. Ainsi que par la diminution des pertes de commutation des interrupteurs, ce qui permet d'aboutir à une fréquence de commutation presque constante aux alentours de 5 kHz. En outre, la commande neuronale est considérée comme étant une commande plus performante que la commande conventionnelle.

La quatrième phase est subdivisée en trois parties. Durant la première partie, on a traité l'environnement et la méthodologie de conception utilisée pour implémenter la commande DTC d'une machine asynchrone. La méthodologie de conception s'appuie sur le concept d'adéquation algorithme architecture, qui permet de minimiser la consommation des ressources matérielles tout en respectant les contraintes temporelles de l'algorithme. On a présenté tout d'abord le flot de conception adopté, la modélisation et la spécification fonctionnelle de haut niveau de la chaîne de commande. Ensuite, on a décrit la méthodologie suivie et les outils utilisés lors de l'implémentation de la commande DTC à base d'un module matériel pour cible FPGA. La description sous VHDL de tous les modules a été présentée, suivi d'une comparaison des résultats obtenus sous VHDL avec ceux de la commande DTC conventionnelle sous Matlab. Finalement, avec l'outil XSE de ISE Xilinx, on a appliqué tout le flot de conception de la phase de synthèse jusqu'à la phase de génération du *bitstream*. La deuxième partie concerne l'implémentation de la commande DTFC sur FPGA. Une étude détaillée sur la description matérielle sous VHDL des modules qui composent le système d'inférence flou a été présentée et discutée. Lors de la dernière partie, on a présenté la description sous VHDL de tous les modules de la Commande directe neuronale de couple de la machine asynchrone pour une implémentation sur une plate-forme à base d'un module matériel FPGA.

La dernière phase de cette thèse, a été consacrée à l'implémentation expérimentale des stratégies de commande DTC de la machine asynchrone. En premier lieu, on a décrit les différents constituants du banc d'essai réalisé dans l'unité de recherche ESIER (Etude des Systèmes Industriels et Energies Renouvelables) à l'ENIM. Ce banc d'essai est réalisé autour d'une plateforme ML507 à base d'un FPGA Virtex5. Une carte d'acquisition et de mesure conçue pour la mesure des courants statoriques, ainsi qu'une carte de commande rapprochée des interrupteurs IGBTs de l'onduleur triphasé. Ces cartes sont protégées contre les surtensions et les surintensités. En deuxième lieu, on a présenté et discuté les résultats expérimentaux relatifs à l'implémentation des stratégies de commande choisies à savoir : (i) la stratégie de Commande DTC conventionnelle basée sur un estimateur du flux statorique, (ii) la stratégie de Takahashi basée sur un observateur à mode glissant de flux statorique et (iii) la stratégie DTFC. Les résultats expérimentaux de la commande floue et la commande conventionnelle montrent une certaine conformité avec les résultats théoriques trouvés

notamment au niveau de la réduction des ondulations et la rapidité au niveau de la dynamique de couple et de flux pendant la phase de démarrage de la machine.

Les travaux de recherche présentés dans ce mémoire de thèse ouvrent un certain nombre de perspectives dans des voies différentes :

A court terme on peut :

- Développer une base de données des IPs dédiée à la commande des systèmes électriques. Cette base contient des modules de code écrit en VHDL des fonctions de services (processeur flou, processeur neuronal, module Concordia, estimateur de couple, régulateur PID ...),
- Améliorer le banc d'essai expérimental, en ajoutant un autre convertisseur analogique numérique à la carte d'acquisition, afin d'assurer la lecture des deux courants statoriques  $i_A$  et  $i_B$  en même temps. Cela permet de réduire le temps d'exécution  $T_{ex}$  de  $4.8\mu s$  à  $2.6\mu s$ ,
- Implémenter le mécanisme de création du temps mort sur FPGA. De cette manière la carte de commande FPGA génère directement les trois signaux de commande  $S_a$ ,  $S_b$  et  $S_c$ , et leurs compléments avec un temps mort entre les signaux d'un même bras de l'onduleur.

A long terme on compte :

- Intégrer la commande Neuro-Floue dans la commande DTC de la machine asynchrone, en profitant des avantages de la logique floue et les avantages des réseaux de neurones,
- Utiliser la technique de reconfiguration dynamique partielle et/ou totale entre deux algorithmes de commande ou plus selon les critères,
- Utiliser la reconfiguration dynamique des algorithmes pour assurer une continuité de fonctionnement en cas de présence d'un défaut de fonctionnement (d'interrupteurs de puissance, de capteurs électriques, mécaniques, ...),
- Aborder la thématique de recherche afférente au domaine touchant le pilotage des véhicules électriques.

# **Bibliographie**

- [AbAb03] A.B Abdelali, A. Mtibaa, L. Boussaid, M. Abid, Li. Chen, "Transformations de haut niveau dans le processus de conception des circuits intégrés dédiés pour des applications de TSI ". GEI.2003
- [ABar06] A. Barriga, S. Sanchez-Solano, P. Brox, A. Cabrera, I. Baturone, "Modelling and implementation of fuzzy systems based on VHDL", *International Journal of Approximate Reasoning*, 164–178. Elsevier (2006).
- [AbbA09] A. Abbou, H. Mahmoudi. "Performance of a sensorless speed control for induction motor using DTFC strategy and intelligent techniques". *Journal of Electrical Systems*. Volume 5, Issue 3, September 2009.
- [AbEl12] A. Elbacha, Z. Boulghasoul, E. Elwarraki, "A Comparative Study of Rotor Time Constant Online Identification of an Induction Motor Using High Gain Observer and Fuzzy Compensator", *WSEAS Transactions on Systems and Control*, Issue 2, Volume 7, April 2012.
- [AbHa12] A. Hammoumi, A. Massoum, A. Meroufel, P. Wira, "Application des Réseaux de Neurones pour la Commande de la Machine Asynchrone sans capteur mécanique", *ACTA ELECTROTEHNICA*, Volume 53, Number 2, 2012
- [AbHf08] H.F. Abdul Waheb, H. Sanusi, "Simulink Model of Direct Torque Control of Induction Machine". *American Journal of Applied Sciences*, 2008. Science Publications.
- [AbKh10] A. Khoudir, "Techniques De Commande Avancées Appliquées Aux Machine De Type Asynchrone". Thèse de Doctorat, Electrotechnique Option Machines électriques. Université Mentouri de Constantine. République Algérienne Démocratique et Populaire. Juin 2010.
- [AbYa12] A ; Yangui, "Stratégie DTC Pour Les Variations De Vitesse Et De Position : Implantation Sur Différentes Associations Moteur Asynchrone Onduleur". Mémoire D'habilitation, Ecole Nationale d'Ingénieurs de Sfax, 12 mai 2012.
- [AdMe07] A. Merabet, "Commande non linéaire à modèle prédictif pour une machine asynchrone", Thèse De Doctorat En Ingénierie. L'université Du Québec. Mai 2007.
- [AlIm04] Imad AL-ROUH, "Contribution à la commande sans capteur de la machine asynchrone". Thèse de Doctorat, l'Université Henri Poincaré, Nancy-I, Juillet 2004.
- [AMel10] A. Mellit, H. Mekki, A. Messai, H. Salhi, "FPGA-based implementation of an intelligent simulator for stand-alone photovoltaic system", *Expert Systems with Applications* 6036–6051. Elsevier 2010.
- [Ames11] A. Messai, A. Mellit, A. Pavan, A. Guessoum, H. Mekki, "FPGA-based implementation of a fuzzy controller (MPPT) for photovoltaic module", *Energy Conversion and Management*, 2695–2704. Elsevier 2011.
- [AmOm06] A. R. Omondi, J. C. Rajapakse, "FPGA Implementations of Neural Networks", Livre publié par Springer 2006.
- [AnMa04] A. Maria, "Commande directe de couple à fréquence de modulation constante des moteurs synchrones à aimants permanents". Thèse de Doctorat en Génie Electrique, Institut National des Sciences Appliquées de Lyon, France, Novembre 2004.

- [AnTr01] A. Trzynadlowski, "Control of Induction Motors", l'Université de Nevada, Reno. Edition Academic Press 2001.
- [AriA00] A. Aria, J. Romeral, E. Aldabas, "Fuzzy logic Direct Torque Control". ISIE'2000, cholula, Mexico. 2000 IEEE.
- [BeSe11a] B. Sebti, "Contribution A La Commande Directe Du Couple De La Machine À Induction". Thèse de Doctorat, Faculté des Sciences de l'Ingénieur, Université De Batna, Mars 2011.
- [BeSe11b] B. Sebti, F. Naceri and R. Abdessemed, "Improvement in DTC-SVM of AC Drives Using a New Robust Adaptive Control Algorithm". International Journal of Control Automation and Systems, IJCAS, vol. 9, no. 2, 2011.
- [BhiS06] Bhim S, Pradeep J, Mittal A, Gupta J, "Neural Network Based DTC IM Drive for electric vehicle propulsion system". 0-7803-9794-0/06. 2006 IEEE
- [BiBo06] B. Bose, "Power Electronics and Motor Drives, Advances and Trends", Academic Press is an imprint of Elsevier, published 2006, 935 pages.
- [BiBo07] B. K. Bose, "Neural Network Applications in Power Electronics and Motor Drives, an Introduction and Perspective". IEEE Transactions on Industrial Electronics, Vol. 54, No. 1, February 2007 IEEE.
- [BibP07] Bibhu P, Dinkar P, Sabyasachi S. "A Simple hardware realization of switching table based direct torque control of induction motor". Electric Power Systems research 2007. Elsevier Publisher.
- [BoAl12] B. Alecsa, M. N. Cirstea, and A. Onea, "Simulink Modeling and Design of an Efficient Hardware-Constrained FPGA-Based PMSM Speed Controller", IEEE Transactions on Industrial Informatics, VOL. 8, NO. 3, AUGUST 2012.
- [CaMa00] C. D. A. MARTINS, "Contrôle Direct du Couple d'une Machine Asynchrone alimentée par Convertisseur Multiniveaux à Fréquence Imposée", Thèse de Doctorat, A l'Institut National Polytechnique de Toulouse. Décembre 2000.
- [CaMa02] C. A Martins, X Roboam, T.A Meynard, A.S Carvalho, "Switching frequency imposition and ripple reduction in DTC drives by using a multilevel converter", IEEE Transactions on Power Electronics, vol. 17, no. 2, pp. 286 – 297, March 2002.
- [CaWe00] C. De Wit, A. Youssef, J.P. Barbot, PH. Martins, F. Malrait, "Observability Conditions of Induction Motors at Low Frequencies", Proceedings of the 39th IEEE Conference on Decision and Control Sydney, Australia December, 2000.
- [CCar00] C. Carlos, "Modélisation Contrôle Vectoriel et DTC, Commande des Moteurs Asynchrone 1", Edition Hermes Science Europe, 2000.
- [Celo05] Handel-C Language Reference Manual, For DK version 4, 2005.
- [ChFo09] C. Forgez, "Contribution à la modélisation et à l'identification temps réel pour les systèmes à énergie électrique embarquée", Habilitation à Diriger des Recherches de l'Université de Technologie de Compiègne. mars 2009

- [ChKh06] C. Kheireddine, "Structures D'identification Et De Commande Des Systemes Non Lineaires Basees Sur Les Techniques Floues", Thèse de Doctorat en Electronique, Université de Batna. Juillet 2006.
- [ChuC90] Chuen C. "Fuzzy Logic in control system: fuzzy logic controller, Part II". IEEE Transaction On Systems Man and cybernetics, vol 20 no 2 march/april 1990.
- [CirM02] M. Cirstea, A. Dinu, J. Khor, M. McCormick, "Neural and Fuzzy Logic Control of Drives and Power Systems", Newnes, An imprint of Elsevier Science First published 2002. 412 pages.
- [CrLa06] C. Lascu, G. Andreescu, "Sliding-Mode Observer and Improved Integrator With DC-Offset Compensation for Flux Estimation in Sensorless-Controlled Induction Motors". IEEE Transactions On Industrial Electronics, VOL. 53, NO. 3, JUNE 2006.
- [DaZa08] D. Zhang, H. Li, "A Stochastic-Based FPGA Controller for an Induction Motor Drive with Integrated Neural Network Algorithms", IEEE Transactions on Industrial Electronics, Vol. 55, No. 2, February 2008 IEEE.
- [DeJi06] D. Jinlian, T. Li, "Improvement of Direct Torque Control Low-speed Performance by Using Fuzzy Logic Technique". Proceedings of the 2006 IEEE, International Conference on Mechatronics and Automation, June 25 - 28, 2006, Luoyang, China.
- [DePi02] D. A. Pizzo, P. Marino, N. Visciano, "Harmonic and interharmonic impact of DTC-based induction motor drives on 3-wire network". Proceedings of the 2002 IEEE, International Symposium on Industrial Electronics vol.4, 2002 Page(s): 1201-1206.
- [DepM88] M. Depenbrock, "Direct Self Control of Inverter-Fed Induction Machines", IEEE Transaction on Power Electronics, vol. PE-3, no 4, Oct 1988, pp 420-429.
- [DeSa11] D. Saptano, "Conception d'un outil de prototypage rapide sur le FPGA pour des applications de traitement d'images", thèse de doctorat, Université De Bourgogne, Novembre 2011
- [DoCa00] D. Casadei, G. Serra, A. Tani, "Implementation of a Direct Control Algorithm for Induction Motors Based on Discrete Space Vector Modulation", IEEE Transactions on Power Electronics, Vol. 15, No. 4, on page(s): 769-777, July 2000.
- [DrTr08] Dramane TRAORE, "Commande non Linéaire sans Capteur de la Machine Asynchrone". Thèse de Doctorat en Automatique et Informatique Appliquée, ÉCOLE CENTRALE DE NANTES. Novembre 2008.
- [DsFr03] D. Seyoum, F. Rahman, C. Grantham, "An Improved Flux Estimation In Induction Machine For Control Application", Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International.
- [DvAn11] D.V.Naga Ananth, "Artificial Neural Network Based Direct Torque Control for Variable Speed Wind Turbine Driven Induction Generator", International Journal of Computer and Electrical Engineering, Vol. 3, No. 6, December 2011.
- [ErMo07] E. Monmasson, and M. Cirstea, "FPGA Design Methodology for Industrial Control Systems – A Review" IEEE Trans. Ind. Electron., vol.54, no.4, pp.1824-1842, August. 2007.



- [ErMo11] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGA in Industrial Control Applications". IEEE Transactions on Industrial Informatics, vol.7, no.2, May 2011.
- [FaAr10] F. G. Areed, A. Y. Haikal, R. H. Mohammed, "Adaptive neuro-fuzzy control of an induction motor", Ain Shams Engineering Journal, 71–78. Elsevier 2010.
- [FaKh11] F. Khoucha, M. S. Lagoun, A. Kheloui, M. H. Benbouzid, "A Comparison of Symmetrical and Asymmetrical Three-Phase H-Bridge Multilevel Inverter for DTC Induction Motor Drives". IEEE Transactions on Energy Conversion, Vol. 26, No. 1, March 2011
- [FaMe10] F. Mehazzem, "Contribution à la Commande d'un Moteur Asynchrone destiné à la Traction électrique". Thèse de Doctorat en Electronique, optronique et systèmes, Université Paris-EST, ESIEE, decembre 2010.
- [FaPi11] F. Piltan, A. Gavahian, N. Sulaiman, M.H. Marhaban, "Novel Sliding Mode Controller for Robot Manipulator using FPGA", Journal of Advanced Science and Engineering Research 1 (2011) 1-22
- [FChe11] F. Chekireda, C. Larbesa, D. Rekiouab, F. Haddad, "Implementation of a MPPT fuzzy controller for photovoltaic systems on FPGA circuit", Energy Procedia, 541–549. Elsevier 2011.
- [FjLi08] F. Lin, L.T. Teng, C.Y. Chen, Y.C. Hung, "FPGA-based adaptive backstepping control system using RBFN for linear induction motor drive", IET Electric Power Applications. IET 2008.
- [FrMo05] F. Morand, "Technique D'observation Sans Capteur De Vitesse En Vue De La Commande Des Machines Asynchrones". Thèse de Doctorat en Génie électrique. L'institut National des Sciences Appliquées de Lyon. Janvier 2005.
- [FrRi03] F. Ricci, H. Le-Huy, "Modeling and simulation of FPGA-based variable-speed drives using Simulink", Mathematics and Computers in Simulation, 183–195, Elsevier 2003.
- [GirC99] C. Girerd, F. Aubebart, P. Poure, J.P.Blondé, Y.A. Chapuis, F. Braun, "Modélisation VHDL/SpectreHDL et simulation mixte sous CADENCE: conception d'un ASIC de commande de moteur asynchrone". Proceedings des Journées Thématiques Universités/Industrie sur l'Adéquation Algorithme Architecture pour les Applications Temps Réel Industrielles Complexes". 23-23 mars 1999, Lille (France), p 64-71
- [GoeA06] A. Goedel, I. N. Silva, P. J. Serni, "Load torque identification in induction motor using neural networks technique", Electric Power System Research, Science Direct 2006.
- [HaHa12] H. Hasnaoui, "Contrôle De Vitesse Par DTC : Mise En Evidence Des Manquements De La Stratégie de TAKAHASHI". Thèse de Doctorat en Génie Electrique. L'école Nationale d'Ingénieurs de Sfax. Mars 2012.
- [HaLi10] H. Li, Q. Mo, Z. Zhao, "Research on Direct Torque Control of Induction Motor Based on Genetic Algorithm and Fuzzy Adaptive PI Controller", International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), vol. 3, pp. 46-49, 2010.

- [HaRe02] H. Rehman, A. Derdiyok, M. Guven, L. Xu, "A New Current Model Flux Observer for Wide Speed Range Sensorless Control of an Induction Machine". IEEE Transactions On Power Electronics, VOL. 17, NO. 6, NOVEMBER 2002
- [HayS99] S. Haykin, "Neural Network – A comprehensive Foundation", Prentice Hall 1999. 897 pages.
- [HoFg11] H. Ouadi, F. Giri, L. Dugard, Ph-Dorléans, A. Elfadili, J.F.Massieu, "High Gain Observer for Induction Motor in Presence of Magnetic Hysteresis", The 18th International Federation of Automatic Control (IFAC) World Congress, Milano (Italy) August 28 - September 2, 2011.
- [HrKe04] H. R. Keyhani, M. R. Zolghadri, A. Homaifar, "An extended and improved discrete space vector modulation direct torque control for induction motors", Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual, on page(s): 3414- 3420 Vol. 5. 20-25 June 2004
- [HusF08] Hussein F, Malik E. "Improving the torque ripple in DTC of PMSM using Fuzzy Logic". 978-1-4244-2279-1/08. 2008 IEEE
- [IbOk10] I. H. OKUMUS, M. AKTAS, "Adaptive hysteresis band control for constant switching frequency in DTC induction machine drives". Turkish journal of Electrical Engineering and Computer sciences, Vol.18, No.1, 2010.
- [ImBa11] I. BAHRI, "Contribution of FPGA-based System-on-Chip controllers for embedded AC drive applications", Thèse de Doctorat, Université De Cergy Pontoise. 2011
- [JaLi08] J. Lis, C. T. Kowalski, T. Orlowska-Kowalska, "Sensorless DTC Control of the Induction Motor using FPGA", International Symposium on Industrial Electronics, June 30 2008-July 2 2008. IEEE 2008.
- [JaPu11] J. H. Pujar, S. F. Kodad, "Robust Sensorless Speed Control of Induction Motor with DTFC and Fuzzy Speed Regulator". International Journal of Electrical and Electronics Engineering 5:1, 2011
- [JeDe03] J DETRY, "Bibliothèque d'opérateurs paramétrables pour l'arithmétique réelle sur FPGA". Rapport de DEA, Ecole Normal Supérieur de Lyon. 07-2003.
- [JeMa98] J. Maes, J. Melkebeek, "Discrete Time Direct Torque Control of Induction Motors using Back-EMF measurement". Thirty- third IAS annual meeting. The 1998 IEEE Industry applications conference 1998. vol1:407-414. 1998 IEEE.
- [JpCa95] J.P.Caran, J.P.Hautier : "Modelisation et commande de la machine Asynchrone". Collection dirigée par Pierre Borne. Editions Technip, 1995.
- [KaHa09] K. Hartani, M. Bourahla, Y. Miloud, M. Sekour, "Electronic Differential with Direct Torque Fuzzy Control for Vehicle Propulsion System", Turkish Journal of Electrical Engineering and Computer Science, Vol.17, No.1, 2009.
- [KeHa02] H. Kebbati, "Conception d'opérateur numérique réutilisable : application a une méthodologie d'implantation rapide et optimale d'algorithme de commande de systèmes électriques". Thèse de Doctorat, Université LOUIS PASTEUR Strasbourg. Décembre 2002

- [KeYo02] Y. Kebbati, "Développement d'une méthodologie de conception matériel à base de modules génériques VHDL/VHDL-AMS en vue d'une intégration de systèmes de commande électriques". Thèse de Doctorat, Université LOUIS PASTEUR Strasbourg. Décembre 2002
- [KhGh05] K. Ghali, "Méthodologie de conception système à base de plateformes reconfigurables et programmables". Thèse de Doctorat, L'université Paris Xi Orsay. Mars 2005.
- [KmDe06] K. M. Deliparaschos, F. I. Nenedakis And S. G. Tzafestas, "Design and Implementation of a Fast Digital Fuzzy Logic Controller Using FPGA Technology", Journal of Intelligent and Robotic Systems. Springer 2006
- [KoKa08] K. Katia, "Contribution des techniques de la logique floue pour la commande d'une machine à induction sans transducteur rotatif", Thèse de Doctorat en électrotechnique, Université de Batna, Algérie. Mai 2008.
- [Labview] <http://www.ni.com/labview/f/>
- [LinF01] F. Lin, J. Yu and M. Tzeng, "Sensorless Induction Spindle Motor Drive Using Fuzzy Neural Network Speed Controller", Electric Power Systems Research, vol. 58, 2001.
- [LinL06] Lin L, Songhua S, Shengping L, Qiang L, Wei L. "Stator Resistance Identification of Induction Motor in DTC System Based on Wavelet Network". Intelligent Control and Automation, 2006. WCICA 2006. 2006 IEEE.
- [LoBa99] L. BAGHLI, "contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques", Thèse de Doctorat, l'Université Henri Poincaré, Nancy-I, Janvier 1999
- [LoCh06] L. Charaabi, "Conception des architectures matérielles dédiées à la commande de systèmes électriques ". Thèse de Doctorat, ENIT Tunisie. décembre 2006.
- [LoZa65] L. A. Zadeh, "Fuzzy sets" Information and Control, vol. 8, pp. 338-353, 1965.
- [MaGh05] M. Ghanes, "Observation Et Commande De La Machine Asynchrone Sans Capteur Mécanique". Thèse de Doctorat en Automatique et Informatique Appliquée. Université De Nantes, École Centrale De Nantes. Novembre 2002.
- [MaGh06] M. Ghanes, J. De Leon and A. Glumineau. "Novel Controller for Induction motor without mechanical Sensor and Experimental Validation", IEEE Conference on Decision & Control, San Diego California USA, pp.4008-4013, December 13-15, 2006.
- [MaHa10] M. Hajian, J. Soltani, G. A. Markadeh, S. Hosseinnia, "Adaptive Nonlinear Direct Torque Control of Sensorless IM Drives With Efficiency Optimization". IEEE Transactions on Industrial Electronics, Vol. 57, No. 3, March 2010.
- [MaPe09] M. Perron, "Conception D'une Stratégie De Commande Vectorielle Énergétiquement Optimale De La Machine Asynchrone Basée Sur Un Modèle Des Pertes Neuronal Sur Fpga", Thèse de doctorat, Faculté des études supérieures de l'Université Laval Québec 2009.
- [Matlab] <http://www.mathworks.com/>

- [MaZe05] M. Zelechowski, "Space Vector Modulated – Direct Torque Controlled (DTC – SVM) Inverter – Fed Induction Motor Drive". Thèse de Doctorat, Faculté des ingénieurs électriques, Université technologique de Warsaw, Poland, 2005.
- [MCir02] M. Cirstea, A. Dinu, J. Khor, M. McCormick, "Neural and Fuzzy Logic Control of Drives and Power Systems", Newnes, An imprint of Elsevier Science First published 2002. 412 pages.
- [MfMi07] M. Faouzi Mimouni, "Modélisation Commande De Machines Electriques". Cours Mastère Génie Electrique. Ecole Nationale d'Ingénieurs de Monastir 2007.
- [MfMi08] M. Faouzi Mimouni, "Commande De Machines Electriques". Cours Mastère Génie Electrique. Ecole Nationale d'Ingénieurs de Monastir 2008.
- [MnUd12] M. N. Uddin, M. Hafeez, "FLC-Based DTC Scheme to Improve the Dynamic Performance of an IM Drive". IEEE Transactions On Industry Applications, Vol. 48, No. 2, March/April 2012.
- [MoBj07] M. Bendjedia "Synthèse D'algorithmes De Commande Sans Capteurs De Moteurs Pas A Pas Et Implantation Sur Architecture Programmable". Thèse de Doctorat en Electronique, Optronique et Systèmes, UNIVERSITE DE FRANCHE-COMTE. Novembre 2007.
- [MoEl08] M. N. ELMobaied, "Fuzzy Logic Speed Controllers Using FPGA Technique for Three-Phase Induction Motor Drives", thèse de doctorat, Université Islamique de Gaza. 2008.
- [MoHa09] M. O. Hamiti, "Réduction des Ondulations de Couple d'une Machine Synchrone à Réductance Variable. Approches par la Structure et par la Commande". Thèse de Doctorat en Génie Electrique de l'université Henri Poincaré, Nancy-I, 15 Juin 2009.
- [MoMo08] M. Mordjaoui, "Modelisation des effets electromagnetiques "Apport de la logique floue et neuro-floue»", Thèse de Doctorat, Electrotechnique Option Machines électriques Université De Batna. Juin 2008.
- [MoNa07a] M. W. Naouar, E. Monmasson, A. Naassani, I. Belkhodja, and N. Patin, "FPGA-Based Current Controllers for AC Machine Drives, A Review ". IEEE Transactions on Industrial Electronics, Vol. 54, No. 4, August 2007.
- [MoNa07b] M. W. Naouar, "Commande numérique à base de composants FPGA d'une machine synchrone ". Thèse de Doctorat, ENIT Tunisie. décembre 2007.
- [MtAb00] A. Mtibaa "Contribution à la conception de haut niveau des circuits Intégrés complexes : Application à un système de transmission de données". Thèse de Doctorat, École Nationale d'ingénieurs de Tunis, Février 2000.
- [MtAb07a] A. Mtibaa, "Langage de description des circuits numérique complexes". Notes de cours. Bibliothèque ENIM inventaire 1179. juin 2007
- [MtAb07b] A. Mtibaa, "les circuits programmables". Notes de cours et transparents. Bibliothèque ENIM inventaire 1180. juin 2007

- [MySh98] M. Shin, D. Hyun, S. Cho, S. Choe, "An Improved Stator Flux Estimation for Speed Sensorless Stator Flux Orientation Control of Induction Motors", Power Electronics Specialists Conference, 29th Annual IEEE 1998.
- [NaUd12] M. Nasir Uddin, M. Hafeez, "FLC-Based DTC Scheme to Improve the Dynamic Performance of an IM Drive". IEEE Transactions on Industry Applications, Vol. 48, No. 2, March/April 2012.
- [NgNg10] N. K. Nguyen, "Approche neuromimétique pour l'identification et la commande des systèmes électriques : application au filtrage actif et aux actionneurs synchrones", Thèse De Doctorat en Electronique, Electrotechnique et Automatique. Université De Haute-Alsace, Mulhouse. Décembre 2010.
- [NrId03] N. R. Idris, A. H. M. Yatim, N. D. Muhamad, T. C. Ling, "Constant Frequency Torque and Flux controllers for Direct Torque Control of Induction Machines". Power Electronics Specialist Conference, 2003. PESC '03. 2003 IEEE
- [OzAk10] O. Akin, I. Alan, "The use of FPGA in field-oriented control of an induction machine", Turkish Journal of Electric Engineering and Computer Scientific, Vol.18, No.6, 2010.
- [PaNo03] P.Nouel, "Langage VHDL et conception de circuits ". Notes de cours, Ecole Supérieur Electronique, Informatique et télécommunications de BORDEAUX, 07-2003.
- [RAbd11] R. Abdelli, D. Rekioua, T. Rekioua, "Performances improvements and torque ripple minimization for VSI fed induction machine with direct control torque". ISA Transactions 2011. Elsevier publisher.
- [RajK08] Rajesh K, Gupta R, Bhangale S, Himanshu G. "Artificial Neural Network Based Direct Torque Control of Induction Motor Drives". IETECH Journal of Electrical Analysis, vol 2, IETECH Publications, 2008.
- [RalK06] Ralph K, Elwy E, Sabry M, Abdou E, Farouk E. "Improved direct torque control for induction motor drives with rapid prototyping system". Energy Conversion and Management 2006. Elsevier Publisher.
- [RaPr11] R. Precup, H. Hellendoorn, "A survey on industrial applications of fuzzy control", Computers in Industry Journal 213–226. Elsevier 2011.
- [RAru10] R. Arulmozhiyal, K. Baskaran, "Implementation of a Fuzzy PI Controller for Speed Control of Induction Motors Using FPGA", Journal of Power Electronics, Vol. 10, No. 1, January 2010
- [RaTa11] R. Trabelsi, "Commande Non Linéaire D'un Moteur A Induction". Thèse de Doctorat en Génie Electrique. L'école Nationale d'Ingénieurs de Monastir. Avril 2011.
- [RaTr12] R. Trabelsi, A. Kheder, M.F. Mimouni, F. M'sahli, "Backstepping control for an induction motor using an adaptive sliding rotor-flux observer"Electric Power Systems Research. 2012 ELSEVIER.

- [RBoj05] R. Bojoi, F. Farina, G. Griva, F. Profumo, A. Tenconi, "Direct torque control for dual three-phase induction motor drives". IEEE Transactions on Industry Applications, volume 41, Issue 6, Nov-Dec. 2005.
- [ReKa60] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45. 1960
- [RiAl02] R. Alvarez Salas, "Développement De Lois De Commandes Avec Observateurs Pour Machine Asynchrone". Thèse de Doctorat en Automatique-Productique. L'Institut National Polytechnique de Grenoble. Janvier 2002.
- [RiTo07] R. Toufouti, S. Meziane, H. Benalla, "Direct Torque Control strategy of Induction Motors". Acta Electrotechnica et Informatica No1, Vol 7, ISSN 1335-8243, 2007.
- [RiTo08] R. Toufouti, "Contribution A La Commande Directe Du Couple De La Machine Asynchrone". Thèse de Doctorat, FACULTÉ DES Sciences DE L'ingénieur, Université Mentouri Constantine, Algérie. Juin 2008.
- [RpDh12] R.P. Dhobale, D.M. Chandwadkar, "FPGA Implementation of Three-Phase Induction Motor Speed Control Using Fuzzy Logic and Logic Based PWM Technique", 1st International Conference on Recent Trends in Engineering & Technology, Mar-2012, Special Issue of International Journal of electronics, Communication & Soft Computing Science & Engineering, ISSN: 2277-9477. Mar-2012
- [RRaj10] R. Rajendran, N. Devarajan, "FPGA Based Implementation of Space Vector Modulated Direct Torque Control For Induction Motor Drive", International Journal of Computer and Electrical Engineering, Vol. 2, No. 3, 1793-8163, June, 2010
- [RRaj12] R. Rajendran, N. Devarajan, "Simulation and Implementation of a High Performance Torque Control Scheme of IM Utilizing FPGA", International Journal of Electrical and Computer Engineering (IJECE), Vol.2, No.3, pp. 277~284, ISSN: 2088-8708, June 2012.
- [RuEl02] R. Ruelland, "Apport de la co-simulation dans la conception de l'architecture des dispositifs de commande numérique pour les systèmes électriques". Thèse de Doctorat, l'institut national polytechnique de Toulouse. Septembre 2002.
- [SaHs10] S. Hadj Said, F. Mimouni, F. M'Sahli, M. Farza, "High gain observer based on-line rotor and stator resistances estimation for IMs", 2010 IEEE International Conference on Control Applications Part of 2010 IEEE Multi-Conference on Systems and Control Yokohama, Japan, September 8-10, 2010.
- [SaZa09] S.A. Zaid, O.A. Mahgoub, K.A. El-Metwally, "Implementation of a new fast direct torque control algorithm for induction motor drives". IET Electric Power Applications. 2009.
- [SeIb04] S. Ibarra-Rojasa, J. Morenob, G. Espinosa, "Global observability analysis of sensorless induction motor", Automatica, Vol.40, Issue : 6, pp. 1079-1085, ELSEVIER Juin 2004.
- [ShIs08] S. Islam, M. Zaman, B. Madon, M. Othman, "Designing Fuzzy Based Mobile Robot Controller using VHDL", International Journal Of Mathematical Models And Methods In Applied Sciences, Issue 1, Volume 2, 2008.

- [SivA09] A. Sivasubramanian, B. Jayanand. "Application of Neural Network Structure in Voltage Vector Selection of Direct Torque Control Induction Motor". International Journal of Applied Engineering Research. Volume 4 Number 6, 2009.
- [SmGa09] S.M. Gdoue, D. Giaouris, J.W. Finch, "Artificial intelligence based speed control of DTC induction motor drives. A comparative study", Electric Power Systems research 210-219, 2009. Elsevier Publisher.
- [SoBo00] S. BOUCHOUX, "Apports de la reconfiguration dynamique au traitement d'images embarqué - Etude de cas : implantation du décodeur entropique de JPEG 2000". Thèse de Doctorat, l'Université de Bourgogne. Mai 2005
- [SoGd08] **S. Gdaim**, "Synthèse de la commande vectorielle de la machine asynchrone", SENDA 08, Workshop international, Monastir, Tunisia, October 24-26 2008.
- [SoGd09] **S. Gdaim**, A. Mtibaa, M.F. Mimouni, "Direct Torque Control of Induction Machine using Fuzzy Logic Technique", 10th International conference on Sciences and Techniques of Automatic control & computer engineering. December 2009, Hammamet Tunisia. Pages 1898-1909
- [SoGd10a] **S. Gdaim**, N. Slama, A. Mtibaa, M.F. Mimouni, "Direct Torque Control based on Artificial Neural Networks of Induction Machine", JTEA2010. Mars 2010, Hammamet Tunisia.
- [SoGd10b] **S. Gdaim**, A. Mtibaa, M. F. MIMOUNI, "Direct Torque Control of Induction Machine based on Intelligent Techniques ", International Journal of Computer Applications (0975 – 8887) Volume 10– No.8, November 2010.
- [SoGd13a] **S. Gdaim**, A. Mtibaa, M.F. Mimouni, "Experimental Implementation of Direct Torque Control of Induction Machine on FPGA". International Review of Electrical Engineering, Vol. 8, N. 1. ISSN 1827- 666. Février 2013.
- [SoGd13b] **S. Gdaim**, A. Mtibaa, M. F. Mimouni, "Design and implementation of Direct Torque Control of Induction Machine on FPGA ". 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO). 2013 IEEE.
- [Synopsys] <http://www.synopsys.com/home.aspx>
- [TaBo01] T. Boumegoura, "Recherche De Signature Electromagnétique Des Défauts Dans Une Machine Asynchrone Et Synthèse D'observateurs En Vue Du Diagnostic". Thèse de Doctorat en Génie Electrique. L'école Centrale De Lyon. Mars 2001.
- [TakI89] I. Takahashi and Y. Ohmori, "High-performance direct torque control of an induction motor,"IEEE Transaction on Industrial Application., vol. 25, pp. 257–264, Mar./Apr. 1989.
- [TeOr11] T. Orłowska-Kowalska, and M. Kaminski, "FPGA Implementation of the Multilayer Neural Network for the Speed Estimation of the Two-Mass Drive System ", IEEE Transactions on Industrial Informatics, VOL. 7, NO. 3, AUGUST 2011.
- [TeRa12] T. Ramesh, A. K. Pandl, Y. Suresh, S. Mikkili, "Direct Flux and Torque Control of Induction Motor Drive for Speed Regulator using PI and Fuzzy Logic Controllers". IEEE-

International Conference On Advances In Engineering, Science And Management March 30, 31, 2012 288, ICAESM -2012.

[ToLi05a] T. Chuen Ling, "Implementation of Direct Torque Control of Induction Machines Utilizing Digital Signal Processor (DSP) And Field Programmable Gate Arrays (FPGA)", Thèse de Doctorat, Université technologique de Malaysia. 2005.

[ToLi05b] T. C. Ling, Idris N, Yatim A, Muhamad N, Elbuluk M. "Implementation of a New Torque and Flux Controllers for Direct Torque Control (DTC) of Induction Machine Utilizing Digital Signal Processor (DSP) and Field Programmable Gate Arrays (FPGA)". Power Electronics Specialists Conference, 2005. IEEE 36th 16-16 June 2005.

[TOS08] Toshinori Munakata, "Fundamentals of the New Artificial Intelligence Neural", Evolutionary, Fuzzy and More Second Edition 2008 springer.

[ToSu11] T. Sutikno, N. Nik Idris, A. Zakwan Jidin, M. Zaki Daud, "FPGA Based High Precision Torque and Flux Estimator of Direct Torque Control Drives", Applied Power Electronics Colloquim (IAPEC), 2011 IEEE.

[TzLi03] T. Li, S. Chang, Y. Chen, "Implementation of Human-Like Driving Skills by Autonomous Fuzzy Behavior Control on an FPGA-Based Car-Like Mobile Robot", IEEE Transactions On Industrial Electronics, Vol. 50, No. 5, October 2003.

[VAmb04] V. Ambrozic, G. S. Buja, R. Menis, "Band-constrained technique for direct torque control of induction motor", IEEE Transactions on Industrial Electronics, vol. 51, no.4, pp. 776–784, Août 2004.

[XiSh07] X. Shao, D. Sun, "Development of a New Robot Controller Architecture with FPGA-Based IC Design for Improved High-Speed Performance", IEEE Transactions on Industrial Informatics, VOL. 3, NO. 4, NOVEMBER 2007.

[XSG11] Xilinx System Generator for DSP, User Guide, UG640 (v 13.1) March 1, 2011

[YiKu09] Y. Kung, C. Huang, M. Tsai, "FPGA Realization of an Adaptive Fuzzy Controller for PMLSM Drive", IEEE Transactions On Industrial Electronics, Vol. 56, No. 8, August 2009.

[YiKu11] Y. Kung, N. Quynhb, N. T. Hieuc, C. Huangd, L. Huange, "Simulink/Modelsim Co-Simulation and FPGA Realization of Speed Control IC for PMSM Drive", International Conference on Power Electronics and Engineering Application, PEEA 2011.

[YuPa10] Y. Pan, Y. Zhang, Z. Wang, "A Novel Variable Domain Adaptive Fuzzy Control of Direct Torque Control for Induction Motor Based on Space Vector Control". Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).

[ZhXu07] Z. Xu, M. F. Rahman, "An Adaptive Sliding Stator Flux Observer for a Direct-Torque-Controlled IPM Synchronous Motor Drive". IEEE Transactions on Industrial Electronics, VOL. 54, NO. 5, OCTOBER 2007.

[ZhZh10] Z. Zhang, R. Tang, B. Bai, D. Xie, "Novel Direct Torque Control Based on Space Vector Modulation With Adaptive Stator Flux Observer for Induction Motors" IEEE Transactions On Magnetics, VOL. 46, NO. 8, AUGUST 2010



# **Annexes**

## 6.1 Annexe A : Bibliothèque de virgule fixe

### 6.1.1 Vue d'ensemble

Pour représenter le nombre réel, on a développé une bibliothèque VHDL d'opérateurs sous le format virgule fixe, qui est le format le plus simple et le plus utilisé. Lors du développement de cette bibliothèque, on a essayé d'élaborer un code VHDL portable en employant une syntaxe simple, garantissant sa portabilité avec la majorité des outils de synthèse, ainsi qu'on n'a pas utilisé des ressources spécifiques à un modèle de FPGA particulier.

### 6.1.2 Principe

On peut définir un système binaire en virgule fixe à l'aide des paramètres Be et Bf définissant respectivement la taille en nombre de bits de la partie entière et celle de la partie fractionnaire de nombre à représenter. Ainsi, la représentation en virgule fixe d'un nombre X peut être décomposée en deux parties, partie entière EX et partie fractionnaire FX, codées respectivement sur Be et Bf bits, et telles que [JeDe03] :

$$X = EX + FX * 2^{-Bf} \quad (6.1)$$

Puisque l'intervalle de balayage de nombre réel vari entre  $10^{-6}$  et 350, on a opté pour la représentation en virgule fixe 32 bits formés par :

- Un seul bit de signe Bs = 1 ;
- Neuf bits pour la partie entière Be = 9 ;
- Vingt deux bits pour la partie fractionnaire Bf = 22.

Un autre format de virgule fixe (16 bits avec Bs = 1, Be = 5 et Bf = 10) a été ajouté pour l'optimisation de certaines parties de l'algorithme.

### 6.1.3 Les opérateurs

Les opérateurs implémentés dans la bibliothèque sont :

- les opérateurs classiques nécessaires à tous type d'application : addition, soustraction, multiplication, division, racine carrée ;
- les différentes fonctions de décalage : décalage à gauche, décalage à droite, décalage multiple à gauche, décalage multiple à droite ;
- les fonctions de conversion de type entier vers virgule fixe et de virgule fixe vers le type entier.
- Et juste pour la simulation, les fonctions de conversion de type réel vers virgule fixe et de virgule fixe vers le type réel.

A titre d'exemple sur l'implémentation des opérations mathématique dans cette bibliothèque, la figure 6.1 illustre le déroulement de l'opération de multiplication sur 16 bits.

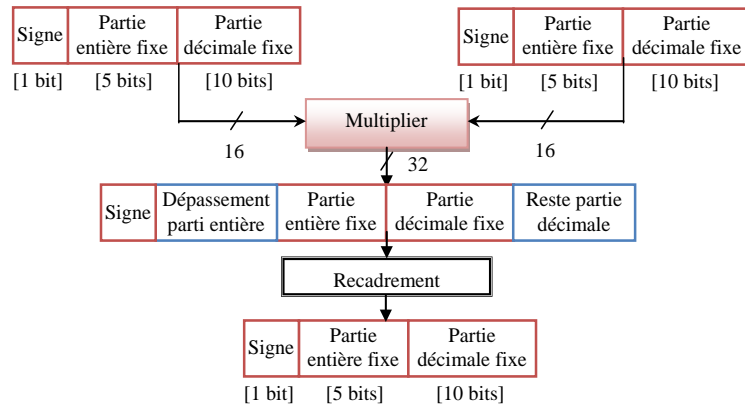


Figure 6.1. Opération de multiplication sur 16 bits

Le tableau 6.1 récapitule les différents opérateurs implémentés dans la bibliothèque ainsi que le format et le nombre de bits des opérandes des entrées et de sorties :

Tableau 6.1. Tableau des entrées et des sorties de la bibliothèque en virgule fixe

Les opérateurs	Les opérandes d'entrées		L'opérande de sortie
+, - et /	32 bits	32 bits	32 bits
+, - et /	16 bits	16 bits	16 bits
*	32 bits	32 bits	32 bits
*	32 bits	16 bits	32 bits
*	16 bits	32 bits	16 bits
*	16 bits	16 bits	16 bits
To_int	32 bits		Integer
To_SFix	integer / réel		32 bits
To_real	32 / 16 bits		Réel
Decalage_D	32 / 16 bits		32 / 16 bits
Decalage_G	32 / 16 bits		32 / 16 bits
Decalage_M_D	32 / 16 bits	N : nombre de décalage	32 / 16 bits
Decalage_M_G	32 / 16 bits	N : nombre de décalage	<b>32 16 bits</b>

## 6.2 Annexe B : Acquisition des grandeurs électriques

La connaissance des grandeurs de courants et de tensions instantanés devienne une nécessité, surtout avec le développement des techniques modernes de commande et de contrôle des machines à induction, d'où la recherche des moyens appropriés pour relever les grandeurs en question. C'est pour cela on a opté à l'utilisation des capteurs de courant et de tension à effet hall de type LEM : LA25-NP et LV25-P.

### 6.2.1 Principe de l'effet hall

Si un courant  $I$  traverse un barreau en matériau conducteur ou semi-conducteur, et si un champ magnétique d'induction  $B$  appliqué perpendiculairement au sens du passage de courant, une tension  $V_h$ , proportionnelle au champ magnétique et au courant  $I$  apparait sur les faces latérales du barreau à cause des électrons qui sont déviés par le champ magnétique. C'est la tension Hall (nom de celui qui a remarqué le phénomène en 1879). Les capteurs à effet Hall sont beaucoup utilisés en raison de leur mise en œuvre aisée, de leur petite dimension et de leur précision. Par ailleurs, les capteurs utilisés se caractérisent par leur bonne précision, faible consommation, faible temps de réponse et aussi par leur grande immunité aux perturbations extérieurs.

### 6.2.2 Représentation des capteurs de courant

Lorsqu'un courant  $i_a$  est présent dans le circuit de puissance, il est recopié par un capteur de courant qui délivre un courant  $I_s$  proportionnel à ce lui ci. Le rapport de transformation entre le courant primaire  $i_a$  et le courant secondaire  $I_s$  est égal à 2/1000. La figure 6.2 illustre le principe de fonctionnement d'un capteur de courant.

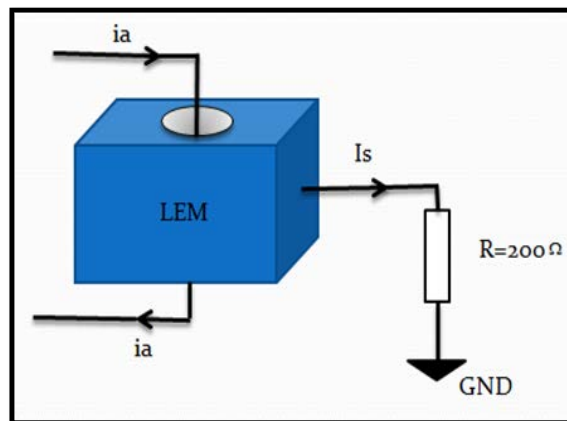


Figure 6.2. Capteur de courant

La résistance  $R$  de sortie du capteur doit vérifier la condition suivante  $R_{\min} \leq R \leq R_{\max}$  avec  $R_{\min} = 100 \Omega$  et  $R_{\max} = 320 \Omega$ .

On prend au choix  $R$  égale à  $200 \Omega$ . Le courant nominal de sortie du capteur est égale à  $25 \text{ mA}$ .

### 6.2.3 Chaîne de mesure et d'adaptation du signal

La chaîne de mesure et d'adaptation de signal pour un capteur de courant est donnée par la figure 6.3.

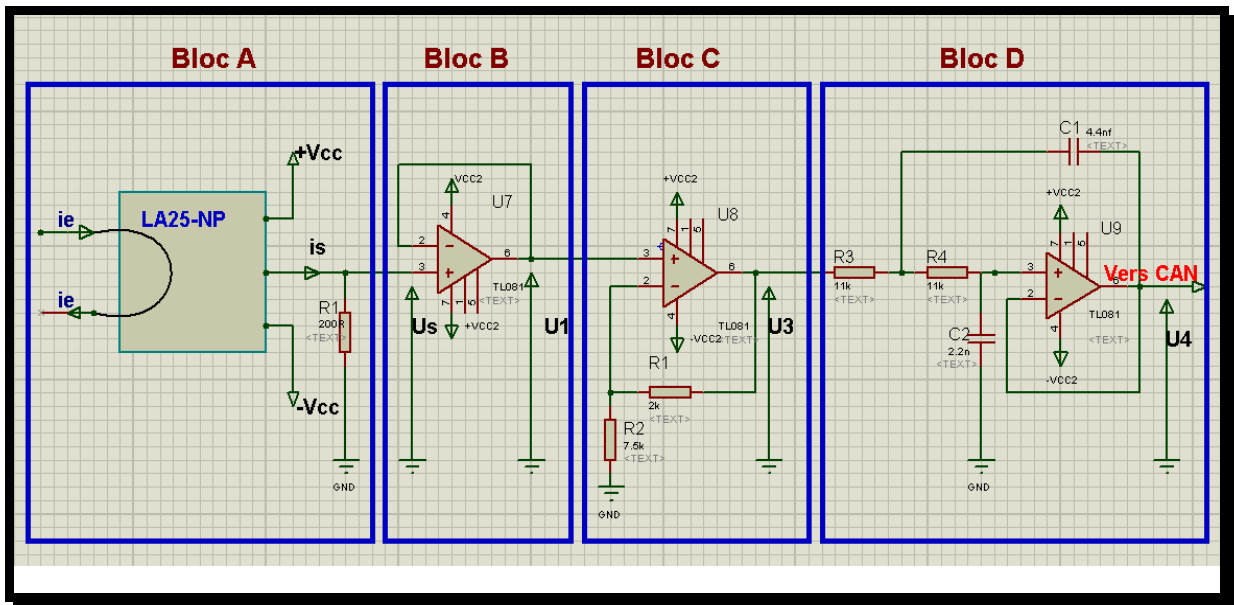


Figure 6.3. Schéma de montage de la chaine de mesure et d'adaptation du signal

• **Description des différents blocs :**

- Bloc A : C'est un bloc de mesure qui nous donne l'image de la valeur instantanée du courant statorique.
- Bloc B : C'est un suiveur pour prélever de sa sortie la tension U1 image de courant  $I_s$

$$U_s = R_1 \times I_s \text{ or } \frac{U_1}{U_s} = 1 (\text{Gain unité, } R_e = \infty) \text{ alors, } U_1 = R_1 \times I_s.$$

- Bloc C : C'est un amplificateur non inverseur dont la fonction de transfert est :

$$H = \frac{R_1 + R_2}{R_1}$$

On choisit  $R_1 = 2k\Omega$  et  $R_2 = 7.5k\Omega$ , Le gain :  $G = |H| = \frac{R_1 + R_2}{R_1}$ . AN :  $G = 1.2666$

- Bloc D : C'est un filtre passe-bas dont la structure est à source contrôlée. La fonction de transfert de ce filtre :

$$H(j\omega) = \frac{1}{1 - R_4 R_3 C_1 C_2 \omega^2 + j(R_4 + R_3)C_2 \omega}$$

Gain statique:  $K=1$ .

Pour  $R_4 = R_3 = R$ ,  $C_2 = 2C_1 = 2C$

$$\text{On obtient : } H(j\omega) = \frac{1}{1 - 2(RC\omega)^2 + j2RC\omega}$$

$$\text{Le gain correspondant est: } G(\omega) = |H(j\omega)| = \frac{1}{\sqrt{1 + 4(RC\omega)^4}}$$

Le Gain est maximum ( $G_{max}=1$ ) en continue ( $w=0$ ) et chute de 3dB pour la pulsation de coupure  $W_c$  telle que  $G(w_c)=\frac{1}{\sqrt{2}}$ , soit  $2(RCW_c)^2=1$  ou  $W_c=\frac{1}{RC\sqrt{2}}$ , on fixe  $f_c$  de quelque KHz ( $f_c=4.65\text{kHz}$ ) et  $R=11\text{k}\Omega$  on obtient  $C=2.2\text{nF}$ .

Le Gain en dB est :

$$G_{dB} = 20\text{Log}|H(jw)| = -10\text{Log}(1 + (RCW)^4)$$

Au-delà de la fréquence de coupure ( $f > f_c$  : hors bande passante), on a  $G_{dB} \approx -10\text{Log}(4(RCW^4)) \approx -40\text{Log}w - \text{Log}(RC\sqrt{2})$ .

- L'asymptote du diagramme de Bode  $G_{dB}(\text{Log}w)$  a donc pour pente : -40dB/décade hors la bande passante.
- La présence des 2 condensateurs provoque une décroissance rapide du gain pour  $f > f_c$ .

## 6.2.4 Conversion analogique numérique

Pour que ces grandeurs électriques puissent être traitées par FPGA, il faut qu'elles soient en leurs équivalents numériques. Cette opération est assurée par des convertisseurs analogiques numériques.

### 6.2.4.1 Caractéristique du convertisseur analogique/ numérique

La conversion est réalisée par un étage de conversion analogique/numérique à base d'un convertisseur A/N de 16 bit. Parmi les critères de choix d'un convertisseur A/N :

- La précision : qui concernant la résolution en nombre de bit.
- La rapidité de conversion : en générale quelque  $\mu\text{s}$ .

Le ADS8509 est disponible en SO 20-pin, il est spécifié pour fonctionner sur le secteur industriel -40°C à 85°C.

Les caractéristiques techniques de l'ADS8509 sont les suivantes :

- Résultat numérique sur 16 bits (résolution =  $1/2^{16}$ ).
- Conversion à approximation successive.
- Temps de conversion 4 $\mu\text{s}$ .

Les chaînes d'entrées analogiques du convertisseur sont : 0-4V, 0-5V, 0-10V,  $\pm 3.3$ ,  $\pm 5$ ,  $\pm 10$ . Pour notre application, on a utilisé la chaîne  $\pm 10$ .

Un convertisseur analogique-numérique à approximation successive est caractérisé par un temps de conversion plus court et conversion fixe, quelle que soit la valeur de la grandeur analogique d'entrée.

### 6.2.4.2 Programmation de l'ADS8509 et interfacement avec l'FPGA :

L'exploitation de ce convertisseur nécessite la génération des signaux de contrôle. Pour générer ces signaux on a implémenté une description VHDL sur la carte FPGA.

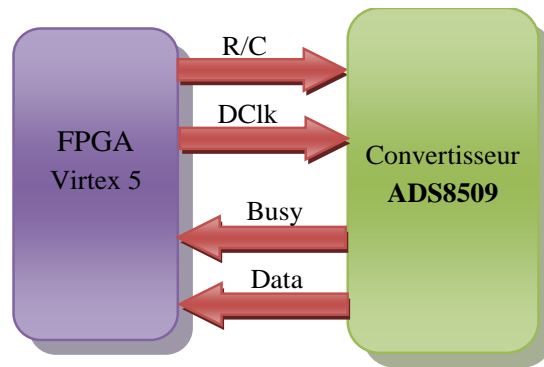


Figure 6.4. Echange des données entre le convertisseur ADS8509 et l'FPGA.

Ce convertisseur analogique numérique possède :

- Une entrée (début de conversion) qui permet de démarrer la conversion (Start= R/C).
- Une sortie (Fin de conversion) qui indique que la conversion est terminée (End=Busy).
- Une entrée analogique.
- une sortie numérique série Data.

La figure 6.5 présente le schéma du câblage du convertisseur avec une entrée analogique variée entre  $\pm 10V$ .

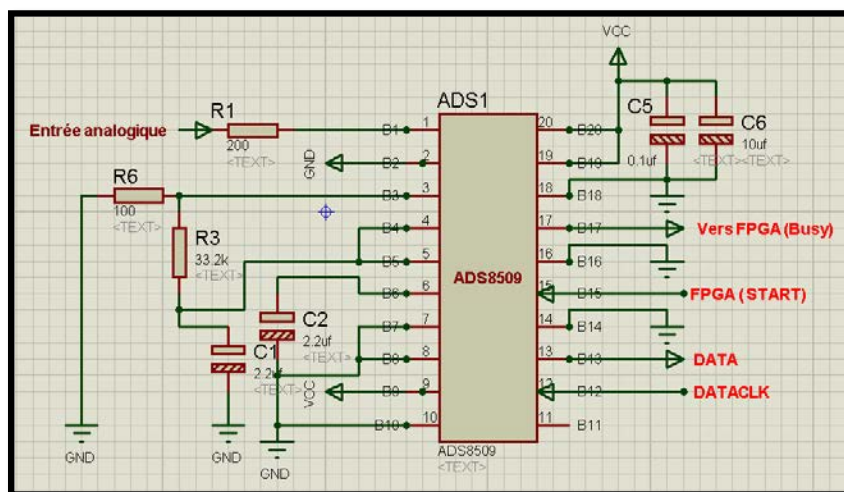


Figure 6.5. Schéma de câblage de l'ADS8509

Un front descendant de R/C déclenche la conversion, une fois la conversion est achevée le signal busy sera mis à 1. A cet instant on envoie une horloge (Dclk) de période 120ns pour recevoir le résultat de conversion en série sur 16 bits, à chaque front descendant de Dclk un bit est reçu.

La figure 6.6 illustre l'organigramme de déroulement d'une phase de conversion.

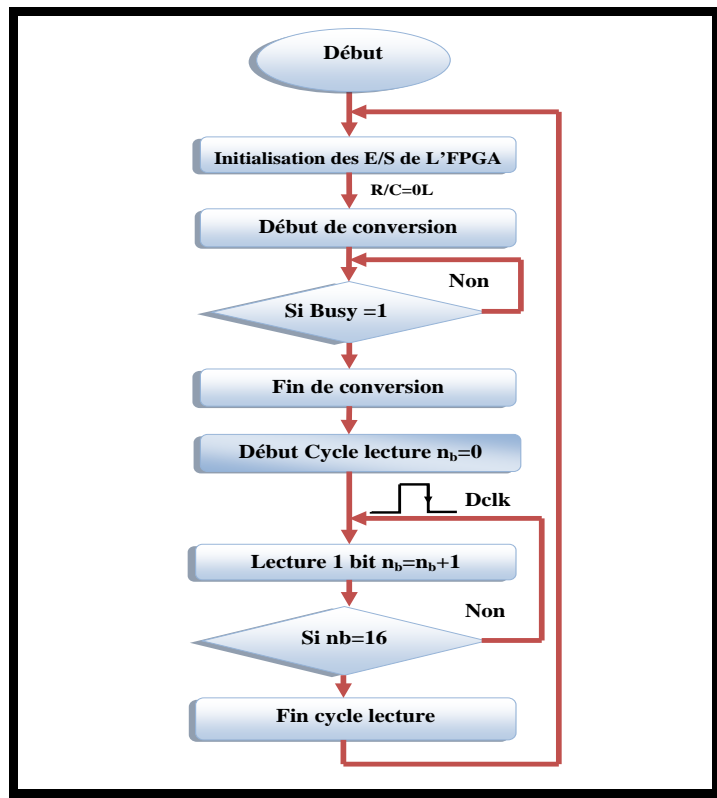


Figure 6.6: Organigramme de déroulement d'une conversion

## 6.2.5 Réalisation de la carte

La carte d'acquisition est subdivisée en deux cartes, une carte de mesure qui contient les capteurs, et une carte pour l'adaptation du signal issue du capteur, le multiplexage ainsi que la conversion analogique numérique.

### 6.2.5.1 Schématique de la carte d'acquisition

La conception de la carte a été élaborée sous l'environnement Proteus Professional de labcenter Electronics. Le schématique est développé sous ISIS et le PCB de la carte sous ARES.

- La carte d'acquisition qui comporte une partie d'adaptation de signal et la partie de conversion analogique numérique est présentée par la figure 6.7.



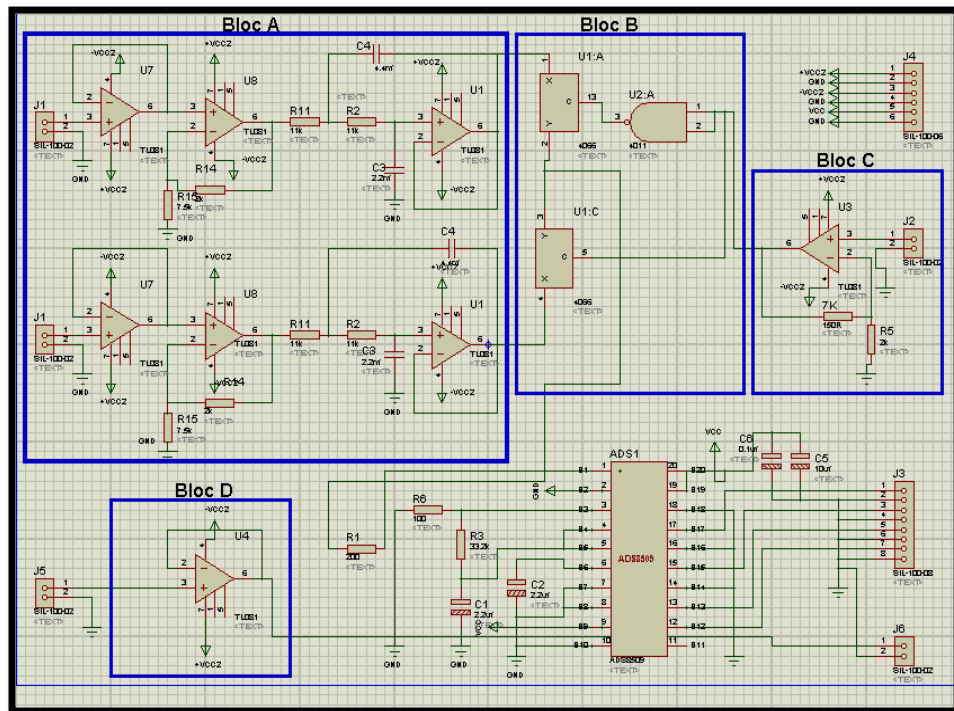


Figure 6.7: Schéma de montage de la carte 1 d'acquisition

Le bloc B contient un multiplexeur qui permet de sélectionner l'une des deux chaînes dont le but est de réduire le nombre de convertisseurs. L'entrée de sélection est envoyée par la carte FPGA, elle est d'amplitude 3.3V et puisque le circuit logique CD4011 (NAND à deux entrées) et CD4066 (multiplexeur) sont de type CMOS, il faut les amplifier à partir d'un montage amplificateur non inverseur qui correspond au bloc C. La figure 6.8 présente une visualisation en 3D de la carte d'acquisition.

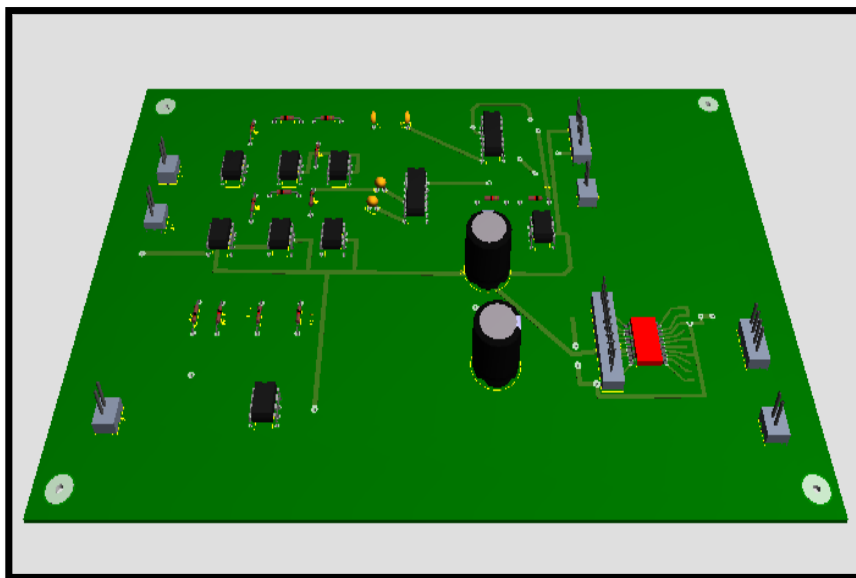


Figure 6.8: Visualisation en 3D de la carte d'acquisition

- La carte des capteurs est donnée par la figure 6.9.

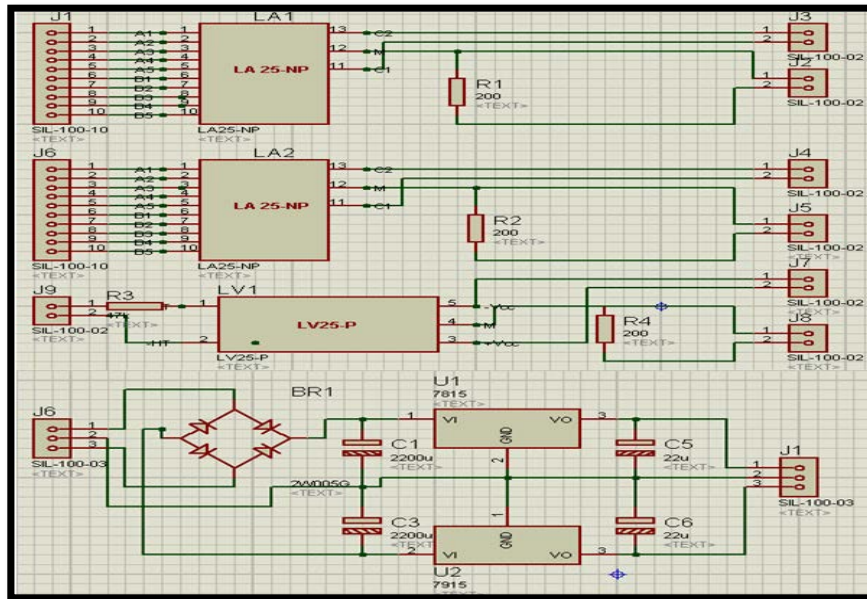


Figure 6.9: Schématique de la carte des capteurs.

La carte de mesure contient deux capteurs de courant (LA25-NP) pour l'acquisition de deux courants statoriques, un capteur de tension (LV25-P) pour la visualisation de la tension simple pour une phase de la machine et une alimentation stabilisée de +15V/-15V pour polariser ses capteurs. La figure 6.10 présente la visualisation en 3D de la carte des capteurs.

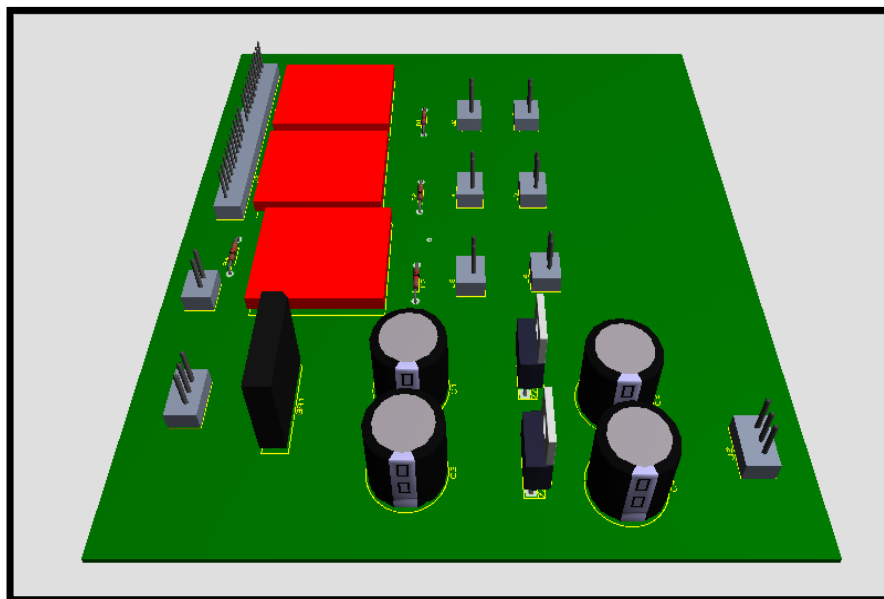


Figure 6.10: Visualisation en 3D de la carte des capteurs

## 6.3 Annexe C : Réalisation de la carte de commande de l'onduleur

### 6.3.1 Structure de la carte de commande

La commande de l'onduleur nécessite une maquette de commande rapprochée. Cette carte, illustrée par la figure 6.11 doit assurer :

- L'isolation optique entre les signaux de commande générée par la carte de commande numérique (FPGA) et la carte de commande rapprochée.
- Amplifier les signaux de commande générés par la carte FPGA (car les signaux issus de la carte sont de valeur 3.3V et les signaux de commande sont de 15V).
- Création du temps mort fixes entre les signaux de commandes complémentaires d'un même bras d'onduleur.
- L'isolation optique entre l'onduleur et la carte de commande rapprochée.

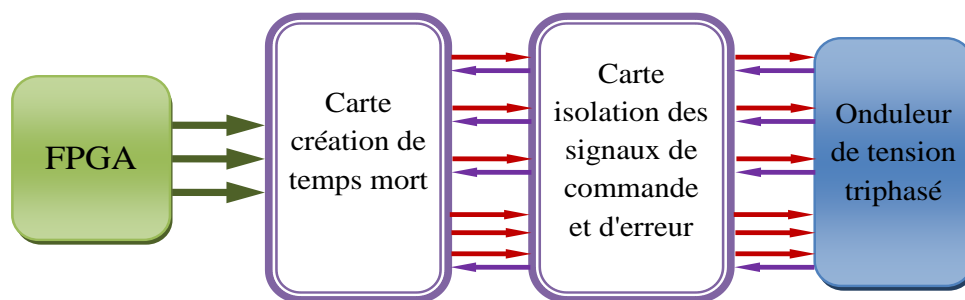


Figure 6.11 : Schéma synoptique de la carte de commande de l'onduleur.

### 6.3.2 Etude de la carte de création du temps mort

La carte de création du temps mort a pour rôle, de garantir l'isolation optique des signaux de commande issus du processus numérique et de générer à l'aide d'un seul signal deux signaux décalés d'un temps  $\tau$  qui est le temps mort. La figure 6.12 détaille cette carte de commande.

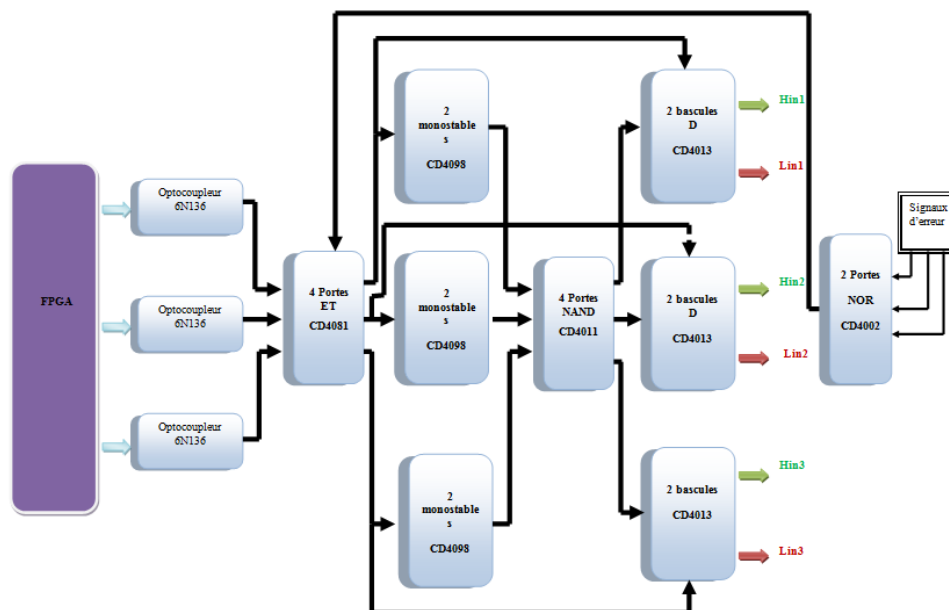


Figure 6.12: Schéma bloc de la carte de commande rapprochée

### 6.3.3 Isolation Optique

Les signaux logiques issus de la carte FPGA se connectent à l'entrée des optocoupleurs de type 6N136 dont le but est d'assurer l'isolation optique entre la carte de commande FPGA et la carte de commande de l'onduleur. Le schéma du câblage de l'optocoupleur est donné par la figure 6.13.

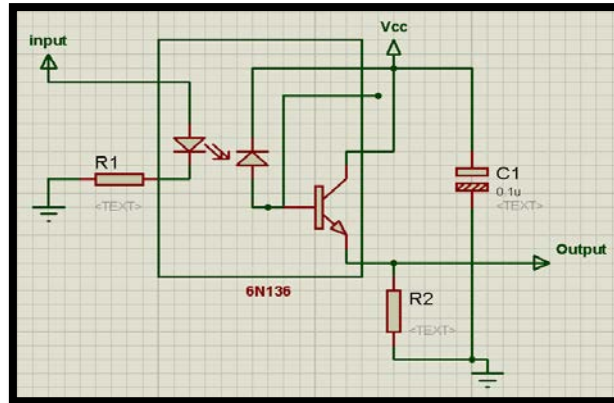


Figure 6.13 : Schéma de câblage de l'optocoupleur 6N136

#### ❖ Dimensionnement de la résistance de l'optocoupleur :

Pour limiter le courant dans la diode de l'otocoupleur, il faut savoir :

- La valeur maximale du courant direct admissible par la diode  $I_d$ .
- La chute de tension au borne de la diode  $V_d$ .

Si  $V_e$  : La tension de polarisation de la diode.

$V_F$  : La tension directe.

$I_F$  : Le courant direct maximum.

La valeur minimale de la résistance utilisée pour limiter le courant dans la diode est calculée par la relation suivante :

$$R_{1\min} = \frac{V_{e\max} - V_F}{I_F}$$

- D'après le datasheet :  $V_F = 1.7V$ ,  $I_F = 16mA$
- $V_e$  prend la valeur 0 ou 3.3V.

Alors  $R_{1\min} = (3.3 - 1.7) / 0.016 = 100\Omega$ .

### 6.3.4 Création du temps mort

Dans le cas réel, le transistor IGBT possède un temps de fermeture et un temps d'ouverture dû à la charge et au décharge de sa capacité d'entrée. Donc la possibilité de la conduction simultanée des deux interrupteurs d'un même bras d'onduleur. Par conséquent, l'envoi de la commande complémentaire issue de la carte FPGA directement vers l'onduleur est pratiquement impossible. Pour éviter ce problème, on doit retarder l'entrée en conduction des deux interrupteurs par l'insertion d'un temps mort entre chaque signal de commande et son complémentaire. La figure 6.14 illustre le principe de la création du temps mort.

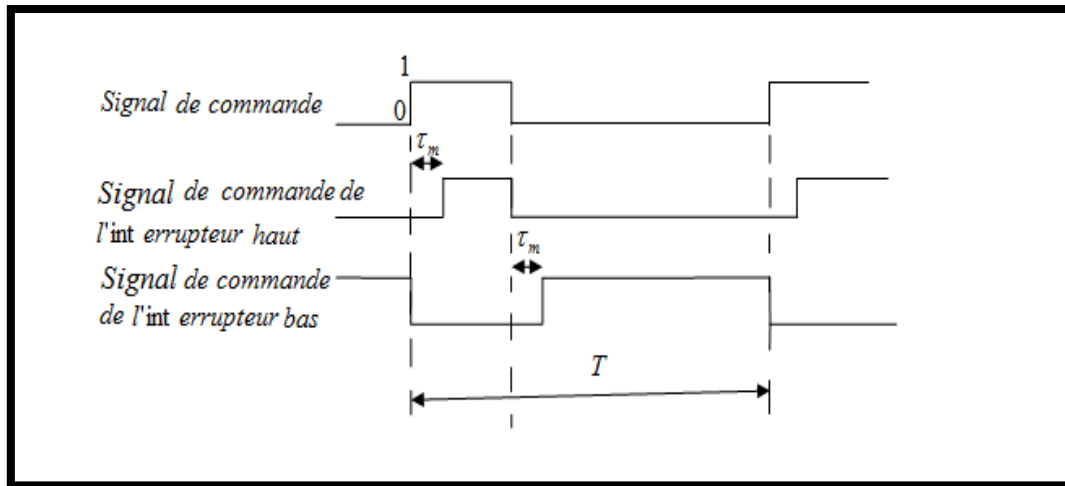


Figure 6.14: principe de création du temps mort

– **Remarque :**

Il faut que le temps mort  $\tau_m$  soit supérieur au temps de retard à la fermeture et à l'ouverture de l'IGBT. Le montage de création du temps mort est basé sur deux monostables CD4098, deux bascules CD4013 et une porte logique NOR.

– **Dimensionnement de la résistance et la capacité du réglage du temps mort :**

CD4098 est un circuit intégré spécialisé CMOS, contient 2 monostables. Il est appelé monostable car il possède un seul état stable. Excité par une impulsion il quitte son état pour y revenir après une certaine temporisation. La figure 6.15 présente le schéma du câblage ce circuit CD4098. Il est utilisé pour générer des impulsions de largeur bien définies. La largeur de l'impulsion est définie par l'expression  $\tau_m = R \times C \times \log(3)$ .

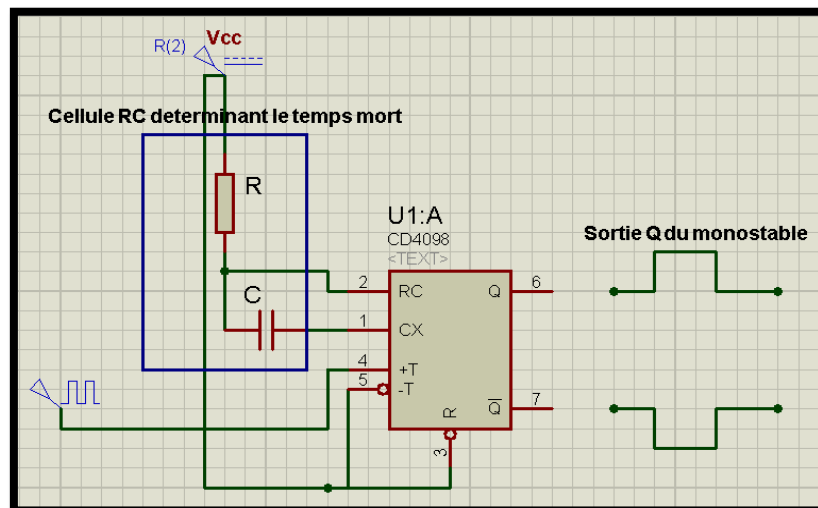


Figure 6.15: Schéma de câblage du circuit intégré CD4098.

Les deux composants C et R déterminent la largeur de l'impulsion de sortie  $\tau_m = R \times C \times \log(3)$ . On a  $\tau_m = 3\mu s$ , on fixe  $R = 30k\Omega$ , on peut donc déterminer la valeur de la capacité :  $C = 1.098 * \tau_m / R = 1.098 * 3 * 10^{-6} / 30 * 10^3 = 0.1098nF$ .

La figure 6.16 présente le montage de création du temps mort, la figure 6.17 présente une visualisation en 3D de la carte de commande de l'onduleur et la figure 6.18 montre sa vue réelle.

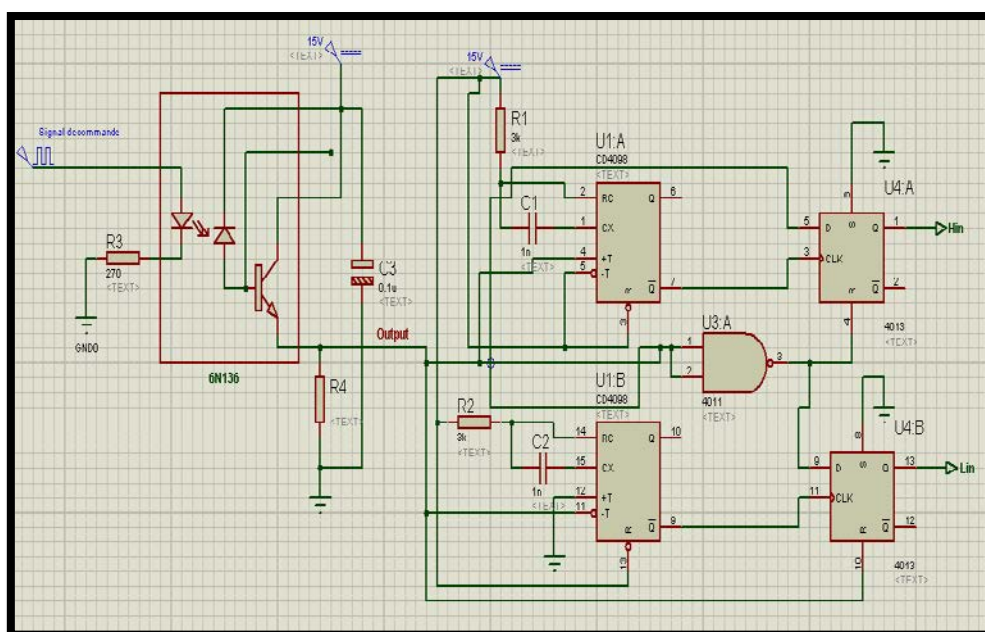


Figure 6.16. Montage de création de temps mort.

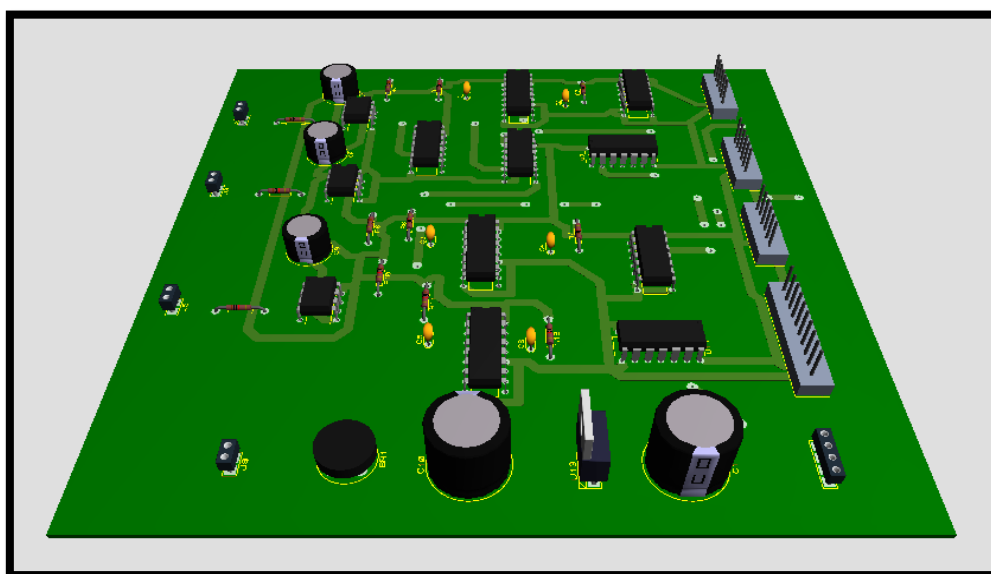


Figure 6.17. Visualisation en 3D de la carte de commande de l'onduleur



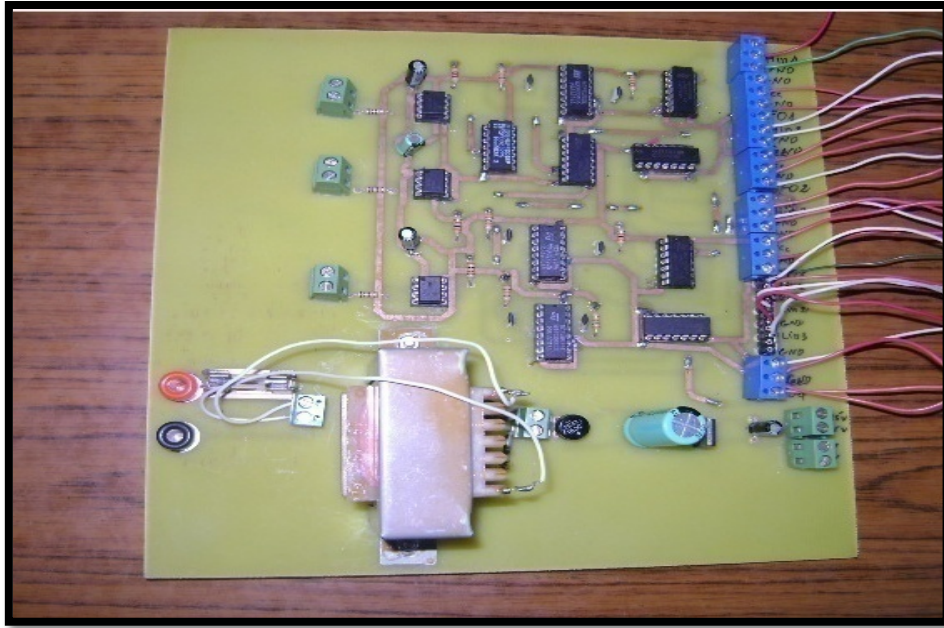


Figure 6.18: Vue réelle de la carte de la commande de l'onduleur.

### 6.3.5 Etude du système de protection de la première carte

L'onduleur possède un système de protection, il génère un signal de défaut dans le cas d'anomalie (surintensité, surtension, court-circuit ou bien une augmentation de température). Notre carte est utilisable pour des systèmes de protection qui possèdent un nombre de pins inférieur ou égale à quatre. Il suffit qu'un seul pin soit activé pour que le système de protection envoie un signal et désactive immédiatement les signaux de commande. Le système de protection illustré par la figure 6.19, comprend deux circuits intégrés CD4002 (deux portes NOR à quatre entrées) et CD4081 (quatre portes ET à deux entrées).

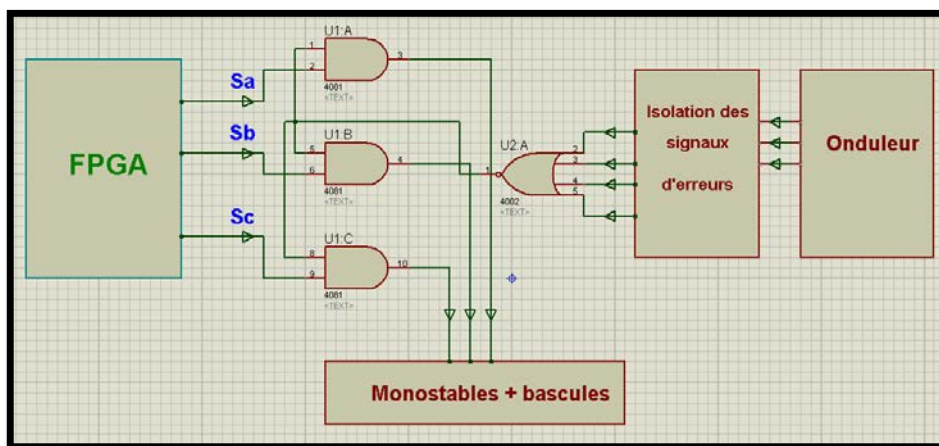


Figure 6.19: Schéma de circuit de protection de la carte de génération du temps mort.

La fonction NOR retourne l'état logique '1' si toutes les entrées sont à l'état logique '0'. Si non elle retourne l'état logique '0' dans ce cas le système de protection génère un signal 'Fault' (15V). Ce dernier passe par une porte NOR qui donne à sa sortie 0V. Par la suite ce signal attaquera avec les signaux issus des optocoupleurs d'isolation à l'entrée de trois portes logique 'ET' du circuit intégré CD4081B. En cas d'anomalie, les signaux de commande seront inhibés et leur passage vers les monostables sera alors impossible.

### 6.3.6 Etude de la carte d'isolation des signaux de commande des IGBTs haut et bas et des signaux d'erreurs

Les trois cartes d'isolation des signaux 'Hin' sont identiques. Chaque carte contient deux optocoupleurs, Un pour l'isolation de signal 'Hin' et l'autre pour l'isolation de signal 'Fault'. La figure 6.20 présente le schéma bloc des cartes d'isolations des signaux de commande.

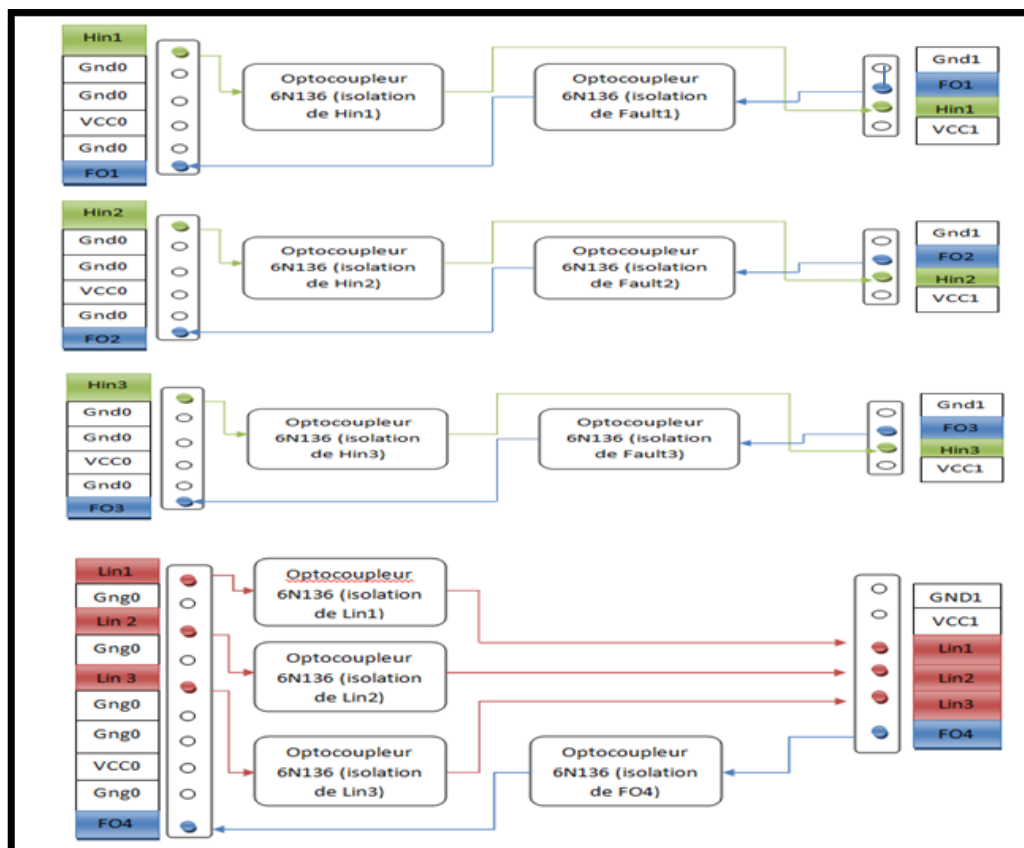


Figure 6.20: Schéma bloc des cartes d'isolations des signaux de commande.

Le signal d'entrée 'Hin' du premier optocoupleur possède un niveau de 15V c'est pour cela il faut redimensionner la résistance d'entrée de l'optocoupleur.  $R = \frac{V_e - V_d}{I_d}$

Application numérique :  $R = \frac{15 - 1.7}{16.10^{-3}} = 831.25\Omega$ .

La figure 6.21 illustre une visualisation en 3D de la carte d'isolation des signaux et la figure 6.22 montre sa vue réelle.



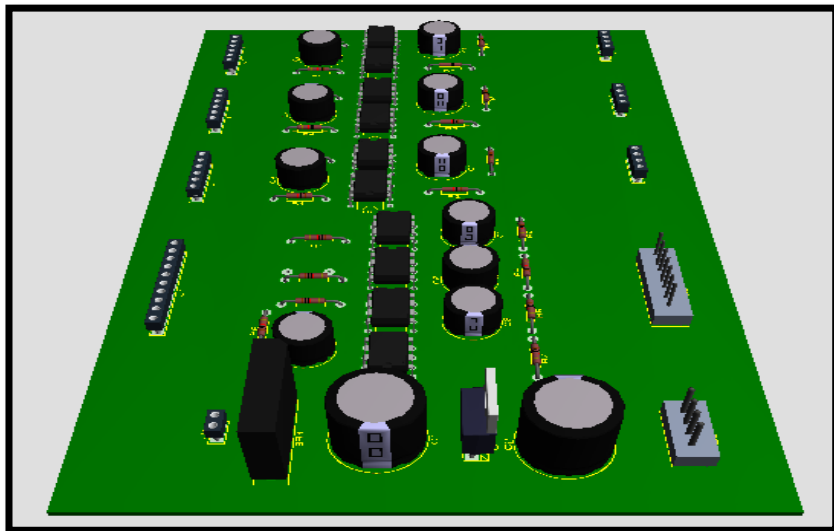


Figure 6.21: visualisation en 3D de carte d'isolation de signaux

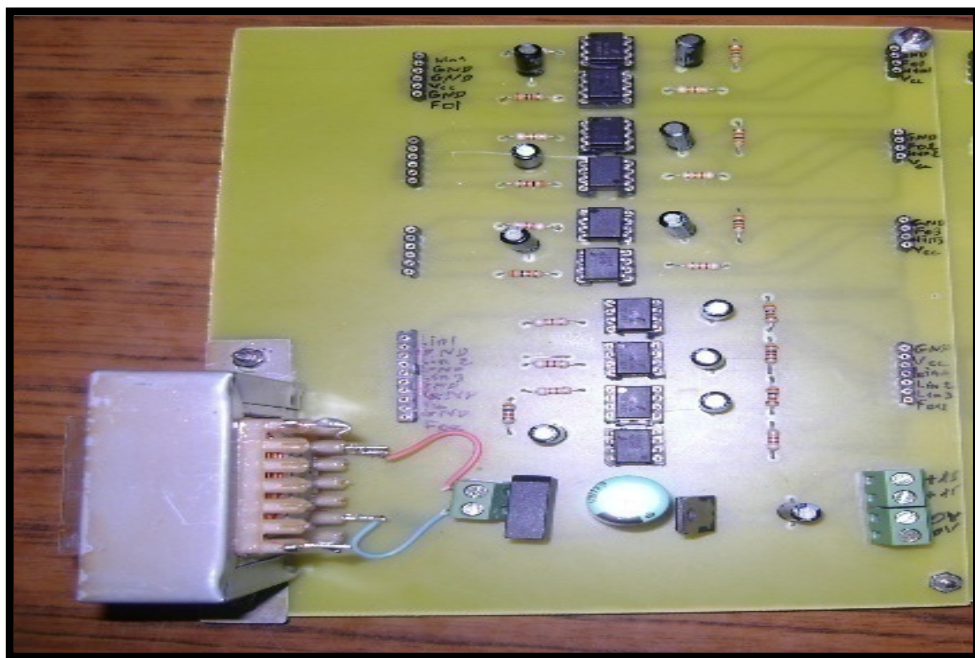


Figure 6.22: vue réelle de carte d'isolation de signaux