



UNIVERSITE DE MOHAMED BOUDIAF- M'SILA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME MASTER EN
GENIE ELECTRIQUE

OPTION : INGENIERIE DES SYSTEMES ELECTROMECHANIQUE

THEME

Implémentation des RNA sur FPGA en vue de commande de la MAS

Proposé et encadré par :

Dr. KHODJA Djalal Eddine

Réalisé par :

BEN MAKHLOUFI Amar

Année Universitaire : 2015/2016

إهداء

باسمك اللهم قد بدأنا ***** وبغير اسمك لا يبدأ الكلام
وعلى خطي نبيك قد توكلنا ***** فصلي اللهم على خير الأنام

ها هي سفينة العلم والإجتهاد تصل الي محطتها الأخيرة، لترمي بمرساتها في ميناء قلوبكم،
لنتستوقفنا قليلا ،،،،، لتنتقل لكم تحية معطرة بروحك ، وتهدي سلاما كبيرا ،،،،،
وعندما أتحدث عن الروح ،،، فلا غيركي **أما** أغلي من الروح ،،،
أهديكي نجاحي اليوم يا مهجة الفؤاد وبلسما لكل الجروح ،،،،،
ولا شيء بدنيتي يليق بكى لأهديكي ،،، وكيف لا ، وجنة الله تحت أقدامكي ،،،
أبي الغالي ،،، قدوتي وحياتي ،،، جدار سميك من ورائي ،،،،،
كيف للحروف أن تهديك كلماتي ،،، وأنت من علمني صنع المعاني ،،،
أهديك سلامي، نجاحي وسعادتي، وأنت البسمة تزين بها أيامي ،،،
إخوتي وأخواتي، يامن بجمعكم أستظل من حر الأيام ،،،،، وبروحم وبسمتكم ،
تكتب الأقلام ،،،،، وبوجودكم في سمائي ، ينبير القمر لي فرحا عتمة الأيام ،،،،،
وهنا تعجز الألسن عن نطق حروفكم **أصدقائي** ،،،،، وقطرت الندى تتساقط مغازلة
لأسمائكم بعيوني ،،، فالقلب مسكنكم والروح مأواكم ، فكيف أهديكم اليوم كلماتي ،،،
ونعمة من الله أنتم أحبتي ،،،،، ومنة منه بقربكم أهداني ،،،
وفي هذا المقام، في **فلسطين** أهديكم تحية العظماء **عائلي** ،،،،، من مقام الشهيد اهديها لكم أحبتي ،،،
على أجنحة الحمام فوق مسجد الأقصى ،،، لغزة الإباء تنشد شوقي وحنيني ،،،،،

وفي الختام ،،، أدامكم الله لي فخرا وحبا ومسكا لكل ختام ،،،،،

شكر

قال تعالى ** { لئن شكرتم لأزيدنكم } *** صدق الله العظيم
قد تاهت الكلمات في ثنايا الروح,,,,, وتناثرت الحروف بشكرك،
فكنت بلسما لكل الجروح,,,,,
ولأنك معلمي,,,, وفي الحياة قدوتي,,, ولأنك في المقام مثل أبى,,,,,
لم تجد عباراتي في بدايتها أجدر بشكرك,,,,,
والكلمات أن تنصفك,,, غير آيات من كتاب الهي,,,,,
أستاذي القدير/د. جلال الدين خوجة. لك منى كل الامتنان.
وكشمس ساطعة إطلالتك قد أنارت دربي,,,, وفي عتمة الصعاب كنت قمرا
وسراجا منيرا لأيامي,,,,,
وباقة من زهور الياسمين، بعطرها كنت مسكا لخاتمتي,,,,,
فأي شكر يليق بك,,,,,, وأي عرفان أصفه لك,,,,,
وحروف اللغة قد توقفت وانحنى فخرا واحتراما
واجلالا لحروف اسمك,,,,,
فتقبل منى معلمي شكري,,,,, وعبارات الإمتنان والعرفان
بحروفي وأسطري,,,,,

,,,,, ادامك الله لنا فخرا,,,,,

Sommaire

LISTE DES FIGURES.....	IV
LISTE DES TABLEAUX	V
LISTES DES ACRONYMES ET SYMBOLES.....	VI
INTRODUCTION GENERALE	1
CHAPITRE 1 : ETAT DE L'ART ET MODELISATION ET COMMANDE DE LA MAS.....	2
I.1 ETAT DE L'ART	2
I.1.1 Introduction	2
I.1.2 Définition de la machine asynchrone.....	2
I.1.3 Constitution	2
I.1.4 Principe de fonctionnement du moteur asynchrone triphasé	3
I.1.4.1 Avantages de la machine asynchrone.....	3
I.1.4.2 Inconvénients de la machine asynchrone	3
I.2 MODELISATION DE LA MACHINE ASYNCHRONE.....	3
I.2.1 Les hypothèses simplificatrices.....	3
I.2.2 Les équations électriques	4
I.2.3. Les équations électromagnétiques	7
I.2.4. L'équation mécanique.....	7
I.3 COMMANDE DE LA MAS	8
I.3.1 Commande scalaire.....	8
I.3.2 Commande vectorielle à flux orienté (FOC)	8
I.3.3 Commande directe du couple (DTC)	9
I.3.4 Comparaison entre FOC et DTC.....	10
I.3.5 Commandes sans capteur de vitesse	10
I.4 MODELISATION DE L'ONDULEUR DE TENSION	11
I.4.1 Introduction.....	11
I.4.2 Les Composants de l'électronique de puissance	12
I.4.3 Modèle mathématique de l'onduleur de tension	16
I.4.4 La technique de modulation "Sinus-triangle" (STPWM)	18
I.4.5 La technique de modulation vectorielle (SVPWM)	20
CHAPITRE 2 : COMMANDE SCALAIRE	30
II.1 INTRODUCTION	30
II.2 COMMANDE SCALAIRE EN TENSION.....	30
II.2.1 Contrôle du flux et réglage du couple.....	30
II.2.2 Structure de régulation de vitesse:	31
II.3 COMMANDE SCALAIRE EN COURANT	32
II.3.1 Contrôle du courant magnétisant et Réglage du couple	32
II.3.2 Structure de régulation de vitesse.....	33
II.3.3 Stratégies de contrôle de courant.....	33
II.3.3.1 Contrôle des courants par hystérésis	34
II.3.3.2 Contrôle des courants par MLI	34
III.4 CONCLUSION SUR LA COMMANDE SCALAIRE	35
CHAPITRE 3 : RESEAUX DE NEURONES ARTIFICIELS	37
III.1 INTRODUCTION.....	37
III.2 DEFINITION	37
III.3 NEURONE BIOLOGIQUE	37
III.4 LE NEURONE FORMEL	38
III.4.1 Structure	39
III.4.2 Modélisation générale	40
III.4.2.1 Nature des entrées et de sortie	40

III.4.2.2 Fonction d'entrée totale.....	40
III.4.2.3 Fonction d'activation	41
III.4.2.4 La fonction de sortie	43
III.5 LES STRUCTURES NEURONALE.....	44
III.5.1 Les réseaux bouclés :.....	45
III.5.2 Les réseaux non bouclés :.....	46
III.6 PROPRIETE FONDAMENTALE DES RESEAUX DE NEURONES : L'APPROXIMATION UNIVERSELLE.....	46
III.7 L'APPRENTISSAGE DES RESEAUX DE NEURONES.....	47
III.7.1 Apprentissage supervisé	48
III.7.2 Apprentissage non supervisé.....	48
III.7.3 Apprentissage par renforcement	49
III.8 PRINCIPALES REGLES D'APPRENTISSAGE.....	49
III.8.1 Règle de Hebb	50
III.8.2 Règle de Widrow Hoff.....	51
III.9 MODELES DE PERCEPTRON DU RESEAU DE NEURONES.....	52
III.9.1 Perceptron multicouches	54
III.10 METHODES D'APPRENTISSAGE DU RESEAU DE NEURONES	55
III.10.1 Méthode d'apprentissage retenue et développée.....	55
III.10.2 Avantage de la règle LM.....	59
III.11 L'ALGORITHME DE RETROPROPAGATION	59
III.12 LA COMMANDE MLI NEURONAL INTRODUCTION	61
III.12.1 Architecture du système	61
III.12.2 Apprentissage :.....	61
III.12.4 Conception de RNA sous Simulink	62
III.12.3 Comparaisent entre MLI et MLI neural :.....	63
III.12.3.1 Interprétation des résultats :.....	65
III.13 CONCLUSION	65
CHAPITRE 04 : IMPLEMENTATION DES RNA SUR CIRCUIT FPGA.....	67
IV.1 LES CIRCUITS PROGRAMMABLES	67
IV.2 LES CIRCUITS LOGIQUES PROGRAMMABLES DU TYPE FPGA.....	68
IV.2.1 Critères de choix du circuit programmable FPGA.....	68
IV.2.2 Avantages et inconvénients des FPGA.....	69
IV.2.2.1 Avantages.....	69
IV.2.2.2 Inconvénients	69
IV.3 Architecture interne des FPGA :.....	70
IV.3.1 Structure des CLB.....	70
IV.3.2 Structure des IOB	72
IV.3 LANGAGE VHDL	72
IV.3.1 Définition	72
IV.3.2 Pourquoi utiliser le VHDL	72
IV.3.3 Unité de conception (module VHDL).....	73
IV.3.3.1 L'entité	73
IV.3.3.2 L'architecture.....	74
IV.3.3.3 Syntaxe générale d'une architecture.....	74
IV.3.3.4 Les paquetages	75
IV.4. LA CONFIGURATION DES FPGA	75
IV.4.1 De l'algorithmique à la conception CAO.....	75
IV.4.2 Méthodologie de conception	76
IV.4.2.1 Spécification du design	77
IV.4.2.2 Développement du design.....	77
IV.4.2.3 Synthèse.....	77
IV.4.2.4 Placement et routage.....	77
IV.5. INTEGRATION ET IMPLEMENTATION.....	78
IV.6 LA STRUCTURE MATERIELLE DE COUPLAGE (FPGA /MACHINE).....	78
IV.7 SYNTHÈSE DE RNA ET SIMULATION SUR XILINX :.....	79
IV.7.1 Génération du code VHDL de bloc RNA	79

IV.7.2 Implémentation de RNA sur FPGA :	80
IV.7.3 Discussions des résultats	81
IV.8 LA CO-SIMULATION	81
IV.4.1 Définition d'un bus de co-simulation :	81
IV.4.2 Les différents types de communication Inter-Processus :	82
IV.4.3 Application de la Co-simulation pour la commande	82
IV.4.3.1.1 Construire le bloc en Xilinx Simulink	83
IV.5 RESULTATS DE CO-SIMULATION	85
IV.5.1 Interprétation des résultats	86
IV.6 CONCLUSION :	87
V. CONCLUSION GENERALE	88
ANNEXE	90
ANNEXE A	90
ANNEXE B	92
REFERENCES	93

Liste des figures

Figure I.1 : constitution de la machine asynchrone	2
Figure I.1.2 : Représentation schématique d'une machine asynchrone triphasée.....	4
Figure I.1.3 : Représentation biphasée d'une machine asynchrone.....	6
Figure I.1.4 : l'onduleur de tension associé à la MAS	16
Figure I.1.5 : principe de la STPWM	18
Figure I.1.6. a : La technique de modulation "Sinus-triangle"	خطأ! الإشارة المرجعية غير معرّفة.
Figure I.1.6.b : La technique de modulation "Sinus-triangle"	خطأ! الإشارة المرجعية غير معرّفة.
Figure I.1.6.c : La technique de modulation "Sinus-triangle"	19
Figure I.1.7 : Les états de l'onduleur et les vecteurs des tensions correspondants	20
Figure I.1.8 : Représentation des états de l'onduleur dans le repère $\alpha\beta$, et les différents secteurs	21
Figure I.1.9 : La tension de référence comme combinaison de deux vecteurs adjacents dans le secteur 1	23
Figure I.10: Signaux de commande dans le cas où V_{ref} est au secteur 1	25
Figure I.1.11: Signaux de commande pour chaque secteur.....	26
Figure I.1.12 : Tensions simples	خطأ! الإشارة المرجعية غير معرّفة.
Figure I.1.13: Comparaison entre théorie et simulation des signaux de	27
Fig. 3.1. Titre de la figure style Légende.	خطأ! الإشارة المرجعية غير معرّفة.

Liste des tableaux

Tableau I.1. Valeurs des tensions correspondantes	22
Tab. 3.1. Impédances comparées de fe	خطأ! الإشارة المرجعية غير معرّفة.

Listes des Acronymes et Symboles

Acronyme

MAS : machine asynchrone.

MLI : modulation de largeur d'impulsion.

FPGA : Field Programmable Gate Array

RNA : Réseau de neurone artificielle

VHDL : est un langage de description de matériel.

Symbole

Vs : vecteurs des tensions statoriques.

Vr : vecteurs des tensions rotoriques.

Is : vecteurs des courants statoriques.

Ir : vecteurs des courants rotoriques.

Φ_s : vecteurs des flux statoriques.

Φ_r : vecteurs des flux rotoriques.

Rs et **Rr** : résistances statorique et rotorique.

Ls, **Lr** et **Lsr** : matrices d'inductances statoriques, rotoriques et des mutuelles inductances entre les phases du stator et rotor.

L^{sab} et **L^{rab}** : inductance mutuelle entre phases statoriques et entre phases rotoriques.

L^{sr} : inductance mutuelle maximale entre phases statorique et rotorique.

Xabc : grandeur X représentée dans les systèmes triphasé (a, b, c).

Xdqo : grandeur X représentée dans les systèmes triphasé (d,q,o).

T : est la matrice de transformation.

Msr : est l'inductance mutuelle cyclique.

ω_s et **ω_r** : représentent respectivement les pulsations statorique et rotorique.

Cem : Le couple électromagnétique.

σ : coefficients de dispersion de Blondel.

n_p : nombre de paires de pôles.

J_m : Inertie mécanique du moteur.

f_m : Coefficient des frottements visqueux du moteur.

C_c : Couple de charge résistant.

S_j : représente sortie d'un neurone,

w_{ij} : les poids synaptiques en provenance du $i^{\text{ème}}$ neurone de la couche d'entrée vers le $j^{\text{ème}}$ neurone de sortie.

θ_i : paramètre qui agit sur le seuil en sortie.

n : le nombre d'élément du vecteur d'entrée.

m : nombre de sorties.

η : paramètre qui contrôle le taux d'apprentissage.

$y_d(t)$: la sortie désirée.

$w(t)$: le vecteur de poids.

Introduction générale

Introduction Générale

Le moteur asynchrone, appelé aussi moteur à induction, a été inventé par Nikola Tesla vers 1890, aux USA. Sur le plan constructif, il est le plus simple des moteurs électriques, donc le plus économique à l'achat. C'est le plus répandu du fait de sa robustesse et de sa simplicité d'utilisation, il est disponible dans une large gamme de puissances allant de 10 W à 25 MW. Alimenté directement par le réseau électrique triphasé, pour l'utilisation à vitesse fixe ou, à vitesse variable (couplé à un convertisseur électronique).

Grâce à l'évolution technologique de l'électronique de puissance et de la micro-informatique, le domaine d'entraînement électrique à vitesse variable, a connu ces dernières années un essor considérable. Cet avantage a joué en faveur de la MAS, car actuellement elle est utilisée pour la réalisation de la majorité des entraînements à vitesse variable.

Pour contrôler la vitesse variable, il existe plusieurs types de commande. Dans notre travail on s'intéresse principalement par la commande scalaire, ce dernier est la première commande qui a été introduite dans l'industrie, très répandue pour sa simplicité et son coût réduit, elle a occupé une grande partie des applications industrielles à vitesses variables.

L'exigence de cette vitesse est d'alimenter le moteur asynchrone à partir des ondes de tensions à amplitudes et fréquences variables imposés par un onduleur de tension à MLI neuronale et piloté par la commande scalaire en appliquant le rapport (V_s/ω_s) constant.

Pour une implémentation numérique de la commande MLI généré par le réseau de neurone artificiel, le circuit électrique choisi est le FPGA grâce à ses plusieurs avantages comparés par les autres circuits comme le DSP, l'ASIC..., etc

Chapitre 01 :Etat de l'art et modélisation et commande de la MAS

Chapitre 1 : Etat de l'art et modélisation et commande de la MAS

I.1 Etat de l'art

I.1.1 Introduction

Le Moteur Asynchrone ou Moteur à Induction est actuellement le moteur électrique dont l'usage est le plus répandu dans l'industrie. Son principal avantage réside dans l'absence de contacts électriques glissants, ce qui conduit à une structure simple et robuste facile à construire. Il permet aussi la réalisation d'entraînements à vitesse variable, et la place qu'il occupe dans ce domaine ne cesse de croître.

Dans les pays industrialisés, plus de 60% de l'énergie électrique consommée est transformée en énergie mécanique par des entraînements utilisant les moteurs électriques. Le modèle mathématique d'une Machine Asynchrone (MAS) nous facilite largement son étude et permet sa commande dans les différents régimes de fonctionnement transitoire ou permanent. [01]

I.1.2 Définition de la machine asynchrone

Les machines asynchrones, appelées également machine à induction sont des convertisseurs électromagnétiques tournants transformant l'énergie électrique en énergie mécanique, elles sont utilisées en moteur bien qu'elles puissent fonctionner en générateur.

I.1.3 Constitution

I.1.3.1 Le stator : c'est la partie magnétique fixe du MAS. Elle comporte des enroulements qui, alimentés en énergie électrique, vont produire un champ magnétique tournant. Le stator est fixé sur la carcasse.



Figure I.1 : constitution de la machine asynchrone

I.1.3.2 le rotor : partie libre en rotation comportant des conducteurs qui seront soumis au champ tournant. Ces conducteurs peuvent être soit des bobines (technologie en voie de disparition) soit des

barres de cuivres ; on parle alors de rotor en court-circuit (ou à cage à d'écureuil). La rotation possible grâce aux paliers supportés par la carcasse.

I.1.4 Principe de fonctionnement du moteur asynchrone triphasé

On dispose sur le stator trois bobines identiques réparties géométriquement à 120° l'une de l'autre. Ces trois bobines alimentées par trois tensions décalées à 120° électrique (période = 360°), issues donc d'un système triphasé, produisent un champ magnétique tournant (champ inducteur), le rotor se trouvant au centre de ce champ. Tant que le rotor a une fréquence de rotation différente que celle du champ inducteur, chaque point de rotor « voit » une variation de champ. Les conducteurs rotoriques produisent donc une f.é.m. qui, dans le circuit fermé, va donner naissance à des courants induits. Ces courants vont à leur tour produire un champ magnétique induit qui va s'opposer à la cause qui lui a donné naissance. Cela se traduit concrètement par un phénomène de poursuite du rotor vis à vis du champ tournant sans qu'il n'arrive jamais à le rattraper.

I.1.4.1 Avantages de la machine asynchrone

Parmi les avantages de la machine asynchrone et notamment en fonctionnement moteur, on peut citer :

- sa robustesse mécanique.
- son faible coût et sa très grande standardisation.
- l'absence d'entretien constant.

I.1.4.2 Inconvénients de la machine asynchrone

L'inconvénient majeur du moteur asynchrone est relié à l'absorption du réactif qu'il faut parfois compenser, à des pertes de glissement et surtout à la nécessité de fonctionner pratiquement au voisinage de la vitesse de synchronisme.

I.2 Modélisation de la machine asynchrone

Plusieurs ouvrages [2, 3, 4, 5] ont été consacrés à l'étude de la machine asynchrone où sa constitution, ses modes de fonctionnement et sa modélisation ont été développés. Dans cette section la modélisation de la machine asynchrone en vue de sa simulation, de sa commande et de développement d'observateurs est rappelée.

I.2.1 Les hypothèses simplificatrices

- Système triphasé équilibré.
- Entrefer constant.
- Effet des encoches négligé.
- Circuit magnétique non saturé et à perméabilité constante.
- Distribution spatiale sinusoïdale des forces magnétomotrices d'entrefer.
- Influence de l'effet de peau et de l'échauffement n'est pas prise en compte.

I.2.2 Les équations électriques

La représentation d'une machine asynchrone triphasée est donnée sur la figure (I.2) où S_a , S_b et S_c sont les axes des phases statoriques et R_a , R_b et R_c sont ceux des phases rotoriques. θ_e est l'angle électrique entre les phases S_a et R_a prisent comme références. Le comportement de la machine est alors régi par les équations suivantes données sous forme matricielle :

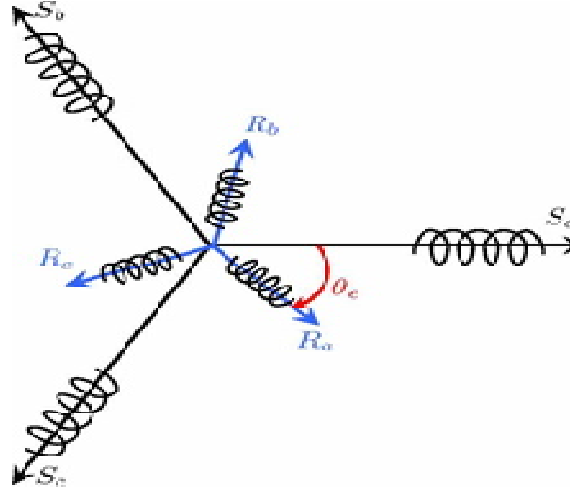


Figure I.2 : Représentation schématique d'une machine asynchrone triphasée

$$V_s = R_s * I_s + \frac{d\phi_s}{dt} \quad (I.1)$$

$$V_r = R_r * I_r + \frac{d\phi_r}{dt} \quad (I.2)$$

$$\Phi_s = L_s * I_s + L_{sr} * I_r \quad (I.3)$$

$$\Phi_r = L_r * I_r + L_{sr} * I_s \quad (I.4)$$

Avec :

$$V_s = \frac{d\phi_r}{dt} \quad \text{et} \quad V_r = \begin{bmatrix} V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix} \quad \text{vecteurs des tensions statoriques et rotoriques.}$$

$$I_s = \begin{bmatrix} I_{sa} \\ I_{sb} \\ I_{sc} \end{bmatrix} \quad \text{et} \quad I_r = \begin{bmatrix} I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix} \quad \text{vecteurs des courants statoriques et rotoriques.}$$

$$\Phi_s = \begin{bmatrix} \phi_{sa} \\ \phi_{sb} \\ \phi_{sc} \end{bmatrix} \quad \text{et} \quad \Phi_r = \begin{bmatrix} \phi_{ra} \\ \phi_{rb} \\ \phi_{rc} \end{bmatrix} \quad \text{vecteurs des flux statoriques et rotoriques.}$$

R_s et R_r résistances statorique et rotorique.

L_s , L_r et L_{sr} matrices d'inductances statoriques, rotoriques et des mutuelles inductances entre les phases du stator et rotor.

$$L_s = \begin{bmatrix} l_s & l_{sab} & l_{sab} \\ l_{sab} & l_s & l_{sab} \\ l_{sab} & l_{sab} & l_s \end{bmatrix} \quad L_r = \begin{bmatrix} l_r & l_{rab} & l_{rab} \\ l_{rab} & l_r & l_{rab} \\ l_{rab} & l_{rab} & l_r \end{bmatrix}$$

L_s et l_r : inductance propre de phase statorique et rotorique.

l_{sab} et l_{rab} : inductance mutuelle entre phases statoriques et entre phases rotoriques.

L_{sr} : inductance mutuelle maximale entre phases statorique et rotorique.

De (I.1), (I.2), (I.3) et (I.4) on a :

$$V_s = R_s \cdot I_s + L_s \cdot \frac{dI_s}{dt} + \frac{d(L_{sr} \cdot I_r)}{dt} \quad (I.5)$$

$$V_r = R_r \cdot I_r + L_r \cdot \frac{dI_r}{dt} + \frac{d(L_{sr} \cdot I_s)}{dt} \quad (I.6)$$

Les équations différentielles (I.5) et (I.6) du système triphasé sont à coefficients variables vu que les inductances mutuelles entre les phases statorique et rotorique L_{sr} dépendent de termes trigonométriques en fonction de l'angle électrique θ_e variable dans le temps. Pour simplifier le modèle de la machine et avoir un système à coefficients constants, on utilise la représentation diphasée magnétiquement équivalente qui est obtenue avec l'utilisation de la transformation de Park.

Cette transformation permet de passer d'un système triphasé $\mathbf{X}_{abc}$ à enroulements d'axes $\mathbf{S}_a$, $\mathbf{S}_b$ et $\mathbf{S}_c$ à un système à enroulements d'axes $\mathbf{S}_d$, $\mathbf{S}_q$ et $\mathbf{S}_o$ orthogonaux qui est équivalent du point de vue de l'intensité des champs magnétiques, des énergies électriques et magnétiques instantanées. Le passage d'un système à l'autre est régi par les équations (I.7) et (I.8) :

$$\mathbf{X}_{dco} = \mathbf{T} \cdot \mathbf{X}_{abc} \quad (I.7)$$

$$\mathbf{X}_{abc} = \mathbf{T}^{-1} \cdot \mathbf{X}_{dco} \quad (I.8)$$

Avec :

$$\mathbf{X}_{abc} = \begin{bmatrix} x_a \\ x_s \\ x_c \end{bmatrix} \quad \text{et} \quad \mathbf{X}_{dco} = \begin{bmatrix} x_d \\ x_q \\ x_o \end{bmatrix} \quad \text{grandeur } X \text{ représentée dans les systèmes triphasé (a, b, c) et (d,q,o)}$$

Respectivement. $\mathbf{T}$ est la matrice de transformation qui peut être choisie soit pour la conservation de puissance avec :

$$\mathbf{T} = \sqrt{\frac{2}{3}} * \begin{bmatrix} \cos(\theta_a) & \cos(\theta_a - \frac{2\pi}{3}) & \cos(\theta_a - \frac{4\pi}{3}) \\ -\sin(\theta_a) & -\sin(\theta_a - \frac{2\pi}{3}) & -\sin(\theta_a - \frac{4\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

(I.9)

Ou pour la conservation d'amplitude en régime permanent avec :

$$\mathbf{T} = \frac{2}{3} * \begin{bmatrix} \cos(\theta_a) & \cos(\theta_a - \frac{2\pi}{3}) & \cos(\theta_a + \frac{2\pi}{3}) \\ -\sin(\theta_a) & -\sin(\theta_a - \frac{2\pi}{3}) & -\sin(\theta_a + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (\text{I.10})$$

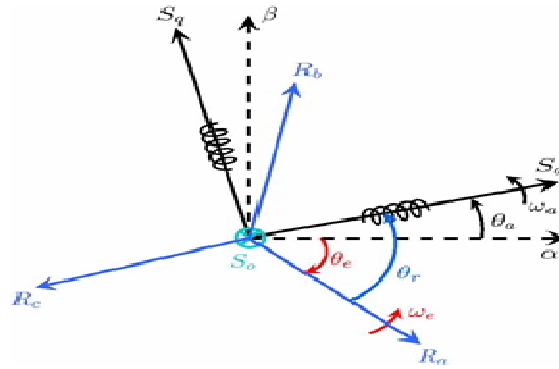


Figure I.3 : Représentation biphasée d'une machine asynchrone

Où θ_a est l'angle que fait l'axe α lié à l'axe de la phase a pris comme référence pour le système triphasé (a, b, c), avec l'axe d pris comme référence du système (d,q,o).

La représentation du stator de la machine asynchrone dans le repère (d,q,o) est donnée sur la figure (I.2). θ_r est l'angle électrique que fait l'axe de la première phase Ra du rotor pris comme référence du système triphasé rotorique avec l'axe Sd,

$\omega_e = \frac{d\theta_e}{dt}$ et $\omega_a = \frac{d\theta_a}{dt}$ sont respectivement les vitesses angulaires électriques du rotor et du repère

(Sd, Sq, So) du stator. Pour un système triphasé équilibré, la composante homopolaire $x_0 = 0$, le modèle de la machine asynchrone biphasée est donné par les équations suivantes :

$$V_{sdq} = R_s * I_{sdq} + \frac{d\phi_{sdq}}{dt} + J * \omega_a * \Phi_{sdq} \quad (\text{I.11})$$

$$V_{rdq} = R_r * I_{rdq} + \frac{d\phi_{rdq}}{dt} + J * (\omega_a - \omega_e) * \Phi_{rdq} \quad (\text{I.12})$$

$$\Phi_{sdq} = L_s * I_{sdq} + M_{sr} * I_{rdq} \quad (\text{I.13})$$

$$\Phi_{rdq} = L_r * I_{rdq} + M_{sr} * I_{sdq} \quad (\text{I.14})$$

Avec :

$$V_{sdq} = \begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix}; V_{rdq} = \begin{bmatrix} v_{rd} \\ v_{rq} \end{bmatrix}; I_{sdq} = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}; I_{rdq} = \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix}; \Phi_{sdq} = \begin{bmatrix} \phi_{sd} \\ \phi_{sq} \end{bmatrix}; \Phi_{rdq} = \begin{bmatrix} \phi_{rd} \\ \phi_{rq} \end{bmatrix}$$

Sont respectivement les vecteurs tensions, courants et flux statoriques et rotoriques du système biphasé.

$L_s = l_s - l_{sab}$ et $L_r = l_r - l_{rab}$: sont les inductances cycliques statorique et rotorique.

$M_{sr} = 3/2 * L_{sr}$: est l'inductance mutuelle cyclique.

Selon l'utilisation à laquelle le modèle est destiné, il est judicieux de choisir le référentiel le mieux adapté :

- Référentiel lié au stator : dans ce cas on aura $\omega_a = 0$, et les grandeurs électriques et magnétiques (courants, tensions et flux) dans ce repère restent sinusoïdales.

Cette transformation particulière de Park est appelée transformation de Concordia.

- Référentiel lié au rotor : $\omega_a = \omega_e$, les grandeurs électriques et magnétiques définies dans ce repère évoluent à la pulsation des courants rotoriques $\omega_r = \omega_s - \omega_e$.

- Référentiel lié au champ tournant : $\omega_a = \omega_s$, dans ce repère, les grandeurs électriques et magnétiques sont continuées. $\omega_s = \frac{d\theta_a}{dt}$ et $\omega_r = \frac{d\theta_r}{dt}$ représentent respectivement les pulsations statorique et rotorique

I.2.3. Les équations électromagnétiques

Le couple électromagnétique instantané de la machine asynchrone peut être exprimé de plusieurs façon dans le repère tournant (d,q) suivant les grandeurs choisies. En prenant la transformation qui permet une conservation de puissance (équation I.9), il vient :

$$C_{em} = -n_p * M_{sr} * (I_{sdq} \wedge I_{rdq}) \quad (I.15a)$$

$$= -n_p * \frac{M_{sr}}{L_r} * (I_{sdq} \wedge \Phi_{rdq}) \quad (I.15b)$$

$$= -n_p * (I_{sdq} \wedge \Phi_{sdq}) \quad (I.15c)$$

$$= -n_p * \frac{M_{sr}}{L_s L_r \sigma} * (\Phi_{sdq} \wedge \Phi_{rdq}) \quad (I.15d)$$

Avec :

$$\sigma = 1 - \frac{M_{sr}^2}{L_s * L_r} \quad : \text{coefficients de dispersion de Blondel.}$$

n_p : nombre de paires de pôles.

Si une conservation d'amplitude est souhaitée, la matrice de transformation donnée en équation (I.10) est utilisée et les expressions du couple électromagnétique doivent alors être multipliées par le coefficients. Les équations (I.15) montrent bien que le couple électromagnétique résulte de l'interaction de composantes de courants et/ou flux statoriques et rotorique en quadrature.

I.2.4. L'équation mécanique

Le mode mécanique de la machine asynchrone est régi par le principe fondamental de la dynamique décrit par la relation suivante :

$$J_m * \frac{d\omega_e}{dt} = n_p (C_{em} - C_c) - f_m * \omega_e \quad (I.16)$$

Avec :

J_m : Inertie mécanique du moteur.

f_m : Coefficient des frottements visqueux du moteur.

C_c : Couple de charge résistant

I.3 Commande de la MAS

I.3.1 Commande scalaire

La commande scalaire, permet de contrôler le couple en régime permanent avec le maintien du flux dans la machine à une valeur fixe. Ce type de contrôle convient surtout à des performances moyennes de fonctionnement de la machine asynchrone.

Cette commande est ce, nonobstant ses inconvénients vis-à-vis ses performances, beaucoup utilisé dans l'industrie car elle est favorisée par sa simplicité et son coût plutôt bon marché. Mais néanmoins il existe dans la littérature des travaux qui visent à son amélioration en utilisant des techniques modernes tel que l'optimisation des régulateurs par logique floue, ou par l'adjonction d'algorithme stabilisant, tout en restant simples à mettre en œuvre [6][7].

I.3.2 Commande vectorielle à flux orienté (FOC)

Quant à la commande vectorielle abrégé FOC, avec ses deux formes, directe DFOC et indirecte IRFOC, elle dépasse largement par ses capacités la précédente lorsqu'il s'agit de contrôle à hautes performances, et reste très compétitive dans le domaine de la commande des machines électriques, car depuis son développement en Allemagne à la fin des années soixante et début soixante-dix par Hasse (IRFOC) et Blaschke (DFOC), elle est toujours mise à jour avec les nouvelles techniques qui se présentent.

Le principe de la FOC repose sur le fait que le couple et le flux de la machine sont contrôlés indépendamment, comme dans une machine à courant continu à excitation séparée. Les courants instantanés statoriques sont transformés dans un repère tournant aligné au vecteur du flux rotorique, statorique, ou ce de l'entre-fer, afin de produire deux composantes du courant, selon l'axe d (composante qui contrôle le flux), et celle de l'axe q (composante qui contrôle le couple).

Le problème majeur de la technique FOC à régulateurs proportionnel intégral (PI) est qu'elle est très sensible aux variations paramétriques de la machine, surtout les constantes de temps rotorique T_r et statorique T_s [8], sur ce fait tous les chercheurs qui ont suivi les formes de bases de la FOC ont eu pour soucis de la désensibiliser en la rendant robuste et fiable vis-à-vis des variations paramétriques malheureusement inévitables. D'innombrables travaux ont été mis en œuvre, parmi eux ce qui ont proposé l'identification en-ligne (on-line) de la constante de temps statorique T_s , et rotorique T_r pour le calcul juste de la vitesse angulaire du glissement, et du fait ont contribué énormément à la l'amélioration de la réponse dynamique du couple et de la vitesse [9][10].

En terme de robustesse envers les incertitudes dont les variations paramétriques et les perturbations externes, [11] a introduit l'application des régulateurs à mode de glissement pour la commande de la MAS, méthode qui a fait ses preuves, et depuis, plusieurs travaux ont suivi [12][13], mais qui avaient le problème du phénomène de broutement (chattering) dû à la discontinuité de la commande, toutefois ce phénomène est parfaitement maîtrisé par l'introduction

de commande douce, intégrale ou par l'utilisation de techniques à base de logique floue [14][15][16].

Il est intéressant de mentionner qu'il y a des travaux qui associent dans la même commande des régulateurs à mode glissant et des régulateurs proportionnels intégrales PI, et en introduisant la technique de modulation de largeur d'impulsion vectorielle SVPWM [17] [18] [19]. Les résultats obtenus sont satisfaisants car la technique SVPWM réduit les harmoniques des courants statoriques, et améliore en régime permanent le problème d'ondulations du couple, flux et courants, tandis que le contrôleur à mode glissant contribue à la robustesse de la commande.

I.3.3 Commande directe du couple (DTC)

Dans les années récentes beaucoup d'études ont été développées afin de trouver des solutions pour la commande de la machine à induction ayant pour but d'avoir une réponse précise est rapide du couple, et réduire la complexité de la commande vectorielle à flux orienté. L'introduction de la commande directe du couple et du flux abrégé DTC par Takahashi (DTC) et Depenbrosk : (DSC) au milieu des années quatre-vingt a été reconnue en tant que solution viable pour réaliser ces conditions. La DTC a ouvert un nouvel horizon dans le domaine de la commande, en effet le principe de cette méthode est de contrôler le couple et le flux de la machine directement, cela est réalisé à travers des comparateurs à hystérésis qui comparent les valeurs de références avec celles estimées, puis commande directement les états de l'onduleur afin de réduire les erreurs de couple et de flux dans les limites de la bande d'hystérésis. Sans passer par des calculs rigoureux de transformation entre repères, et moins de régulateurs jugés trop sensibles aux variations des paramètres ni le besoin de générateur à modulation de largeur d'impulsion (MLI) ou (PWM en anglais). Ce qui conduit à une réponse dynamique beaucoup plus intéressante comparée avec la technique FOC [20] [21]. Quand le flux est hors de la bande d'hystérésis, la fréquence de l'onduleur change et le flux prend un chemin optimal vers la valeur désirée, ceci est la cause du principal inconvénient de la DTC qui est l'ondulation que présente le couple, le flux, et le courant en régime permanent, ceux-ci sont reflétés sur l'estimation de la vitesse et sa réponse, et aussi se traduisent par des bruits acoustiques accrus [22].

Comme la FOC, plusieurs voir même beaucoup de travaux sont issus des deux formes de base proposées par Takahashi et Depenbroek afin de pallier à ces problèmes persistants. il y a ce qui ont utilisé l'onduleur multi niveaux [23], solution qui s'avère complexe et coûteuse, et d'autres qui ont utilisé la technique SVM(space vector modulation), son principe est d'imposer le vecteur de tension approprié par modulation vectorielle d'espace [24], les ondulations sont réduits considérablement mais à fréquence de commutation pas tout à fait constante d'autant plus que cette méthode avait des dépendances paramétriques notamment la résistance statorique et exigeait de grands calculs en ligne. Dans [25] les auteurs ont remplacé la table de vérité par plusieurs en appliquant un nombre de vecteurs plus grand que ceux appliqué en DTC classique en utilisant des comparateurs à cinq niveaux, cette technique est appelée DSVM (discret» spece vector modulation), la méthode a été implémentée en simulation et réalisé expérimentalement, les résultats obtenus ont montré l'efficacité de cette méthode vis-à-vis des ondulations sans augmenter la complexité de la DTC originale. Récemment d'autres études ont utilisé la logique floue ; soit pour adapter la bande

d'hystérésis [21] et ont obtenus des résultats satisfaisants même en basse vitesse, soit pour optimiser la table de vérité [26] avec de bonne performance du couple et du flux en régime permanent, d'autres ont utilisé la technique qui associe SVM avec logique floue FWTC [27], les ondulations ont été remarquablement minimisées à fréquence de commutation quasi constante.

I.3.4 Comparaison entre FOC et DTC

Des études comparatives entre la FOC et la DTC ont eu lieu afin de montrer les avantages et inconvénients de chacune, dans [28] les auteurs ont conclu après une étude expérimentale que la DTC avait les mérites par rapport à la FOC, pas de nécessité de régulateurs de courants ni de transformation de coordonnées, pas sensible aux paramètres de la machine mis à part la résistance statorique et pas besoin de capteur de vitesse pour l'implémenter. Mais l'article a omis la comparaison en termes d'ondulations du couple et du flux. Par contre dans [29] les auteurs ont mis en œuvre une comparaison assez juste des deux techniques de base DTC et DFOC car ils ont en commun d'être intrinsèquement sans capteur de vitesse, les résultats ont montrés clairement qu'en régime permanent et pour de différentes valeurs de la vitesse et du couple de charge, les courants dans la DFOC sont nettement moins ondulés par rapport à ceux de la DTC, qui en plus présentent un spectre riche en harmoniques de faibles amplitudes mais qui s'étale sur toute la plage de fréquence à l'inverse de la DFOC. Aussi en grande vitesse, l'amplitude des ondulations du couple dans la DTC été légèrement plus grande que celle en DFOC qui avait une forme plus régulière et uniforme. Par suite et en régime transitoire, les performances de chaque commande ont été testé suite à la réponse à un échelon de couple de charge pour différentes valeurs de vitesse, les résultats ont montrés que la DTC était meilleur en terme temps de réponse, ceci est dû à la présence des régulateurs PI dans la DFOC qui retarde la réponse du couple. Aussi les auteurs ont remarqué que dans les basses vitesses la DTC été moins stable dû à l'effet de la résistance statorique. La conclusion de l'article est que les performances des deux techniques sont comparables, et peuvent être amélioré pour éviter ainsi les inconvénients que présente chacune d'elle.

La meilleure technique sera donc celle qui sera améliorée en termes de robustesse en vers les incertitudes internes et externes et en termes de réponse dynamique, toute en restant moins compliquer à l'implémentation industrielle.

I.3.5 Commandes sans capteur de vitesse

L'utilisation des commandes développées pour la machine asynchrone lui confère beaucoup d'avantages, mais l'utilisation de capteurs mécaniques (tel que des génératrices tachymétriques, resolvers, ou codeurs incrémentaux) pour la vitesse nuit à la robustesse et la fiabilité des commandes pour de multiples raisons. D'abord placé sur l'arbre de la machine le capteur augmente le volume et le coût global du système, en effet dans les machines à petites puissances (2 à 5 kW) le coût du capteur est environ semblable à celle de la machine, aussi pour les machines à 50 kW, le coût est de 20 à 30% par rapport au coût de la machine elle-même [30]. En plus, le placement du capteur nécessite de modification sur l'arbre de la machine et demande un montage délicat et une attention spéciale aux bruits de mesure.

Afin de pallier ces inconvénients inhérents, beaucoup de travaux ont été proposés pour la commande vectorielle de la machine asynchrone sans capteur mécanique, Ces méthodes se

favorisent selon, leurs sensibilités aux variations paramétriques, leurs simplicités à l'adaptation et à l'implantation.

Il y a celles qui sont basées sur l'exploitation de l'anisotropie spatiale (saillance) du circuit magnétique tel que les encoches rotorique [31].

D'autres techniques que sont les observateurs, utilisent le modèle de la machine telle que l'observateur de Luenberger [32], et le filtre de Kalman [33], bien que ces méthodes sont en mesure de palier les variations importantes de la résistance rotorique, malheureusement l'implantation du filtre de Kalman en temps réel est difficile à cause du nombre d'opérations nécessaire à chaque période d'échantillonnage pour réactualiser les estimations. Dans [34] qui a utilisé un observateur à mode glissant, il a augmenté la robustesse du système et a étendu la plage de vitesse de zéro aux grandes vitesses, mais le phénomène du chattering reste l'inconvénient majeur de ces techniques. La méthode basée sur le système adaptatif à modèle de référence abrégé MRAS a prouvé que c'est l'une des meilleurs techniques proposés par les chercheurs, grâce aux grandes performances qu'elle présente en termes de fiabilité, stabilité, et moins d'efforts de calculs [35]. Depuis son introduction par [36], les chercheurs ne cessent d'améliorer et d'exploiter les avantages qu'offre cette technique [37] [38] [39] [40]. Cette dernière se base sur deux modèles de la machine asynchrone, le modèle dit de référence et le modèle adaptatif, qui se basent sur le flux, la contre force électromotrice (f_{cem}), ou la puissance réactive, ces deux modèles sont comparés et leur différence est introduite dans un mécanisme d'adaptation (régulateur PI) dont la sortie est la vitesse désirée. Cette dernière technique sera adoptée dans la suite de ce travail.

I.4 Modélisation de l'onduleur de tension

I.4.1 Introduction

L'onduleur de tension assure la conversion de l'énergie continue vers l'alternatif (DC/AC). Cette application est très répandue dans le monde de la conversion d'énergie électrique aujourd'hui.

L'onduleur peut être utilisé à fréquence fixe, par exemple alimenter un système alternatif à partir d'une batterie, ou à fréquence (MLI) variable pour la variation de vitesse des machines électriques. L'onduleur de tension à MLI permet d'imposer à la machine des ondes de tensions à amplitudes et fréquences variables à partir du réseau standard 230/400V, 50Hz.

La structure du convertisseur statique qui alimente la machine est constituée essentiellement, d'un pont redresseur (AC/DC) connecté au réseau, contrôlé ou pas [41]. Après redressement, la tension (étage continu) est filtrée par des composants passifs C ou LC, pour être finalement appliquée à l'onduleur.

L'onduleur qui est connecté à la machine, est constitué de trois bras formés d'interrupteurs électroniques choisis essentiellement selon la puissance et la fréquence de travail, chaque bras compte deux composants de puissance complémentaires munis de diode montée en antiparallèle. Les diodes de roue libres assurent la continuité du courant dans la MAS une fois les interrupteurs sont ouverts.

À noter qu'un temps de retard doit exister pratiquement entre les interrupteurs haut et bas d'un même bras afin d'éviter le court-circuit de la source continu.

L'onduleur est commandé par la technique de modulation de largeur d'impulsion (MLI), appelée en anglais (Pulse Width Modulation PWM). Il existe plusieurs techniques PWM, dont deux seront mentionnées, la PWM dite sinus-triangle (STPWM), et la MLI vectorielle ou (space vector PWM) abrégée (SVPWM), devenue très sollicitée par les industriels et chercheurs en commande des machines électriques [42].

Avant d'entamer la modélisation de l'onduleur, on a jugé intéressant de faire un descriptif sur les interrupteurs statiques en semi-conducteurs utilisés en électronique de puissance qui existent actuellement, car l'élément clé de la conversion d'énergie est l'interrupteur statique qui va permettre, en interrompant ou non le transfert d'énergie entre les divers éléments du circuit, et de gérer les valeurs moyennes des courants et tensions.

I.4.2 Les Composants de l'électronique de puissance

Les composants de l'électronique de puissance, doivent répondre à une commande qui exige l'état passant ou bloqué, ce changement de conductivité est possible grâce aux structures spéciales des semi-conducteurs qui contrôlent les porteurs de charges mobiles, trois différents types sont définis :

- **semi-conducteurs non contrôlés** : diodes de puissance : PiN, SCHOTTKY, JBS.
- **semi-conducteurs contrôlés avec commande amorçage** : Thyristors, Triacs.
- **semi-conducteurs contrôlés avec commande amorçage/blocage** :
 - Thyristors GTO (Gate Turn Off)
 - Transistors bipolaires de puissance BJT (Bipolar Junction Transistor)
 - Transistors de puissance MOSFET (Metal Oxide Semiconductor Field Effect Transistor)
 - IGBT (Insulated Gate Bipolar Transistor)
 - GCT et IGCT (Integrated Gate Commutated Turn-off Thyristor)
 - MCT (Mas Controlled Thyristor)
 - SIT (Static Induction Transistor)
 - SITH (Static Induction thyristor)

Un nouvel horizon, qui est l'intégration en électronique de puissance, évolue vers une intégration de systèmes de puissance, comprenant les composants de conversion de l'énergie, des composants passifs, des éléments de commande et de protection. Deux technologies distinctes sont définies, la technologie hybride ou monolithique qui se divisent elle-même en :

- Circuits intégrés à haute tension HVIC (High Voltage Integrated Circuits).

- La technologie « Smart Power » : puissance intelligente. On entend par « intelligent », le fait que les composants puissent réagir à différentes situations sans aide extérieure, donc ils puissent avoir une certaine autonomie.

I.4.2.1 Choix des interrupteurs

Les paramètres suivants sont importants pour un interrupteur en semi-conducteur conçu pour les applications de conversion de puissance [43] :

- Courant maximal supporté.
- Tension maximale bloquée.
- Chute de tension à l'état passant et sa dépendance de la température.
- Courant de fuite à l'état bloqué.
- Capacité thermique.
- Temps de transition de commutation pendant l'ouverture et la fermeture.
- La tenue en dv/dt à l'état bloqué ou pendant l'ouverture.
- La tenue en di/dt à l'état passant ou pendant la fermeture.
- Capacité de résister aux courants et tensions excessifs.
- Pertes de commutation.
- La puissance contrôlée.

De manière générale, la constitution des semi-conducteurs de puissance est différente de leurs équivalents de courant faible. Des modifications sont faites sur les structures de bases afin de les adapter aux hautes tensions et aux forts courants.

En électronique de puissance, la diode bipolaire est souvent appelée diode PiN où i désigne une région peu dopée de type N, ainsi, le profil du champ électrique en régime inverse et les tenues en tensions seront augmentées par rapport à la diode PN standard [44].

La jonction MS (metal-semiconductor) est couramment connue sous la diode Barrière-Schottky, les principales caractéristiques d'une diode Schottky sont, la capacité à travailler en haute fréquence, ses faibles pertes (chute de tension très réduite) en polarisation directe. Elle est employée intensivement dans le domaine de l'électronique de puissance, en particulier en basses tensions et fort courants [45].

La diode JBS est un composant hybride issu d'un croisement entre une diode PiN et une diode Schottky. L'intérêt d'une telle structure est d'obtenir une chute de tension directe plus faible qu'une diode bipolaire, tout en ayant un courant de fuite inverse plus faible qu'une diode Schottky [46].

La technologie des thyristors est intrinsèquement plus performante que celle des transistors pour les valeurs de tension de blocage supérieures à 2,5 kV, elle offre un meilleur compromis entre la tension à l'état passant et la tension de blocage. L'ajout de la fonction d'extinction de gâchette impose plusieurs restrictions aux GTO. En effet La technologie de commande standard des GTO, largement répandue, provoquait des transitoires de commutations non homogènes, imposant l'utilisation de coûteux circuits amortisseurs de dv/dt et di/dt , et de circuits de gâchette encombrants. Toutefois, elle a trouvé des domaines d'application intéressants pour les puissances entre 1 et 20 MV A, principalement dans les entraînements électriques à vitesse variable et dans la traction ferroviaire. Ces interrupteurs fonctionnent à des fréquences de commutations relativement basses [44] [47].

Les transistors bipolaires BJT sont utilisés dans les applications allant jusqu'à plusieurs centaines de kilowatts et à des fréquences allant jusqu'à 10 kHz. Les plus utilisés généralement dans les convertisseurs de puissances sont de type NPN, car ils possèdent, une chute de potentielle plus faible à l'état passant que le PNP, et ils ont une fréquence de commutation plus élevée (les électrons circulent plus vite que les trous) [48]. Pour la raison de son temps de commutation relativement grand, les pertes de commutation sont d'autant plus grandes que la fréquence augmente. Pour éliminer le besoin de grand courant de base, la configuration Darlington est généralement utilisée [44]. Aussi pour conserver l'état passant le courant de base doit être présent en permanence, et un fort courant inverse et requis pour commuter à l'état ouvert rapidement, de ce fait le circuit de commande de la base devient complexe et coûteux [49].

Toutes ces limitations ont conduit au développement des transistors MOSFET, ces derniers ont permis d'obtenir des fréquences de commutations très élevées voisines de 1 MHz, avec une commande en tension qui nécessite une énergie pratiquement négligeable. Cependant, les MOSFET présentent de chutes de potentiels dus à la résistance à l'état passant, et donc des pertes en conduction d'autant plus importantes que la tension est grande par rapport aux BJT de même puissance. Ce qui limite leurs applications en haute tension à environ 1,2 kV [48].

Le transistor IGBT est l'un des composants de puissance le plus utilisé actuellement. Sa structure associe en sortie celle d'un transistor bipolaire BJT et celle d'un MOSFET en entrée. Donc l'IGBT regroupe les avantages en conduction du transistor bipolaire BJT, et les avantages en commutation et simplicité de commande du transistor MOSFET. Il est largement utilisé pour les applications de moyenne et dans certaines applications de haute tension, avec des tensions d'alimentation comprises entre 600V et 6.5kV, et des fréquences de l'ordre de 130 kHz [50] [51].

Les dispositifs « Smart Power » et HVIC, sont réalisés à partir de filières technologiques de type circuits intégrés. Des associations plus ou moins complexes sont envisageables, allant du simple interrupteur formé d'un IGBT avec sa diode en antiparallèle, le bras de pont, les ponts en H, ou triphasé (six-pack), ou menu du hacheur (seven-pack), pour s'acheminer vers le module de puissance complet, baptisé PIM (power integrated module), réunissant dans un même boîtier : un pont redresseur à diodes, un hacheur de freinage, ainsi que les six IGBT de l'onduleur avec leurs diodes de roue libre. Quant aux modules dits « intelligents » ou /PM (intelligent power module), ils incluent en sus le driver et différents capteurs chargés de mettre en évidence un échauffement de température excessif, une surcharge ou un court-circuit. Ils sont Conçus spécialement pour la commande des moteurs [52].

L'IGCT ou GCT (IGCT est un produit de ABE, GCT est un produit de Mitsubishi, mais le concept est le même), est un GTO optimisé et muni d'un circuit de commande de gâchette à faible inductance (constitué principalement de substrat, MOSFETs, et capacités), afin de réduire considérablement la limitation due au di/dt , et permettre un fonctionnement sans circuit limiteur de dv/dt (snubber), qui sont les principaux inconvénients des GTO, en plus de la fréquence de commutation assez lente. De ce fait il exploite de manière optimale la technologie des thyristors (pertes faibles et haut niveau de fiabilité) ainsi que l'extinction commandée sans circuit amortisseur (plus économique) pour les applications aux puissances de 0,5 MV A à plusieurs centaines de MV A à des fréquences allant jusqu'à plusieurs kHz [47].

Le thyristor MCT est une amélioration du thyristor (GTO) classique, où deux MOSFET sont intégrés afin de pouvoir forcer le composant au blocage et à l'amorçage. Ces thyristors offrent une solution intéressante pour la réalisation de convertisseurs, ils ne nécessitent pas d'impulsion d'extinction. Le MCT possède deux principaux avantages vis à vis du GTO, une commande plus simple pour commuter de l'état passant à l'état bloqué (un fort courant négatif n'est pas nécessaire) et des temps de commutation plus brefs (de l'ordre de quelques microsecondes). Le MCT est un composant commandé en tension comme l'IGBT ou le MOSFET, et la même énergie est nécessaire pour commuter un MCT, un IGBT ou un MOSFET. Les MCT présentent également de plus faibles tensions à l'état passant comparé aux IGBT ayant des caractéristiques similaires [43] [53].

On parlera maintenant des deux éléments de puissance le SIT et le SITH, tous deux très semblables dans la structure de base avec une légère modification au niveau des couches semi-conductrices. Leurs principes de fonctionnement sont identiques au transistor JFET, qui peut s'assimiler à celui d'un barreau de semi-conducteur pour lequel on vient modifier sa résistance par l'application d'une tension négative sur la grille. Pour le SIT et le SITH, Le courant principal circule entre le drain et la source, et son passage est limité par la résistance de la région N faiblement dopée, une grille, constituée de petits domaines de type P+, permet de modifier la conduction du substrat par effet d'induction. Les transistors statiques à induction SIT peuvent fonctionner à une puissance 100 kW à 100 kHz ou 10 W à 10 GHz [48].

Avant de fermer cette brèche, notons que tous les composants cités sont au Silicium, les recherches actuelles s'orientent vers un nouveau matériau semi-conducteur, on parle du Carbure de Silicium (Silicon Carbide) SiC. Le SiC est en mesure de bloquer une tension nettement plus élevée que le Silicium, cet avantage significatif hisse le SiC parmi les meilleurs candidats pour les hautes tensions dans les dispositifs de l'électronique de puissance. Le nombre de composants de puissance réalisés sur carbure de silicium par diverses équipes augmente, la diversité des composants. On note ainsi dans la littérature des caractérisations de diodes PiN, Schottky, JBS, de MOSFET de puissance, de thyristors et d'IGBT [46]. Dans [44], une synthèse très satisfaisante est mise, des composants sus mentionnés en SiC avec des comparaisons de leurs homologues en Si.

I.4.2.2 Conclusion

Le MOSFET est très bien adapté pour les convertisseurs basse tension et à fréquence élevée (inférieure à 200Vet supérieure à 50kHz) alors que l'IGBT est utilisé pour les tensions supérieures à 300V et des fréquences rarement supérieures à 50kHz. Les GTOs et thyristors sont dédiés aux applications haute tension (>1 kV) fort courant (> 1kA). Le SIT ne peut répondre qu'à des applications où la tension à bloquer est inférieure à 1 kV. Les composants à base de thyristor, sont celle de l'avenir, Il paraît probable que le GTO soit supplanté par le SITH, et peut-être aussi l'IGBT par le MCT pour les grandes puissances. Les modules IGBT ont un domaine d'application qui recouvre totalement celui des transistors bipolaires, partiellement celui des MOSFET et des GTO.

C'est pourquoi les IGBT qu'ils soit en modules ou discret sont les composants d'avenir dans les fortes et moyennes puissances, en effet les IGBTs ont trouvé leurs domaines de prédilection dans la commande industrielle des machines électriques, ils sont proposés à une très large variétés qui couvre une gamme de puissance très étendue [54].

Actuellement l'enjeu des industriels est de concevoir des dispositifs permettant de travailler dans des conditions de plus en plus difficile, augmentation de la température de fonctionnement, diminution de la taille des composants et augmentation des calibres courant/tension. Les compromis sont de plus en plus serrés et le silicium atteint ses limites. Aujourd'hui, le silicium répond au marché mais de nouvelles applications apparaissent pour lesquelles de nouveaux composants sont développés sur des nouveaux matériaux tels que le SiC.

I.4.3 Modèle mathématique de l'onduleur de tension

L'état des interrupteurs, supposés parfaits peuvent être définis par trois grandeurs booléennes de commande S_i ; ($i = a, b, c$) :

- $S_i = 1$ le cas où l'interrupteur de haut est fermé et celui d'en bas ouvert, (voir figure. I. 10 plus bas).
- $S_i = 0$ le cas où l'interrupteur de haut est ouvert et celui d'en bas fermé.

Dans ces conditions on peut écrire les tensions via en fonction des signaux de commande S_i et en tenant compte du point fictif '0' représenté sur la figure.1.4 :

$$V_{io} = V_{dc} \left(S_i - \frac{1}{2} \right) \quad (I-17)$$

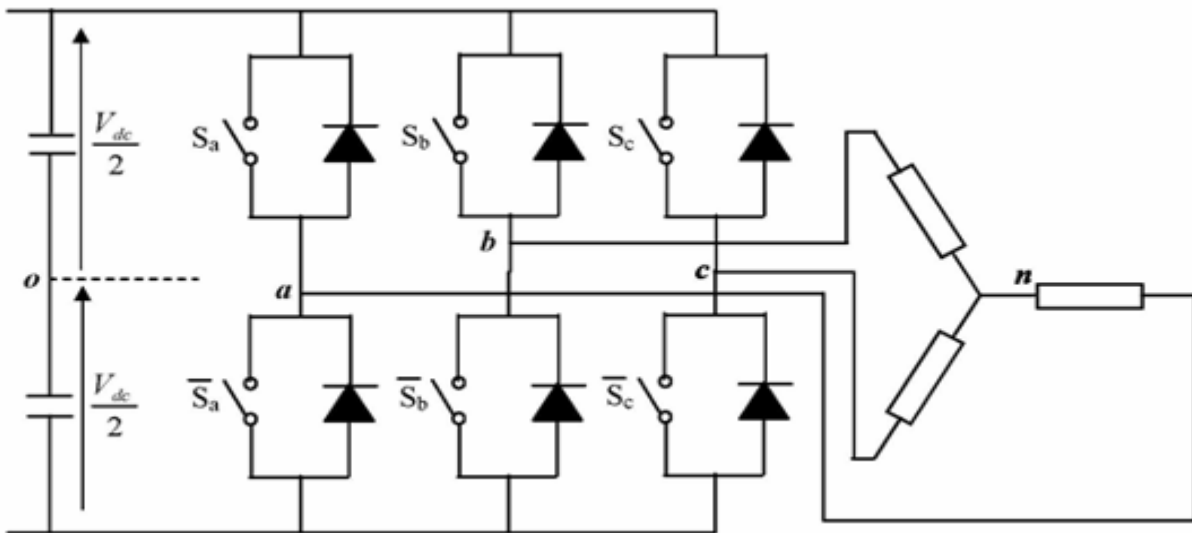


Figure I.4 : l'onduleur de tension associé à la MAS

Soit 'n' le point neutre du côté alternatif (MAS), alors les trois tensions composées : V_{ab} , V_{bc} , et V_{ca} sont définies par les relations suivantes:

$$\begin{cases} v_{ab} = v_{an} - v_{bn} \\ v_{bc} = v_{bn} - v_{cn} \\ v_{ca} = v_{cn} - v_{an} \end{cases} \quad (I.18)$$

La charge constituée par la machine est équilibrée ($v_{an} + v_{bn} + v_{cn} = 0$), on aura donc :

$$\begin{cases} v_{an} = \frac{1}{3} (v_{ab} - v_{ca}) \\ v_{bn} = \frac{1}{3} (v_{bc} - v_{ab}) \\ v_{cn} = \frac{1}{3} (v_{ca} - v_{bc}) \end{cases} \quad (I.19)$$

En faisant apparaître le point "o", les tensions entre phases peuvent aussi s'écrire :

$$\begin{cases} v_{ab} = v_{ao} - v_{bo} \\ v_{bc} = v_{bo} - v_{co} \\ v_{ca} = v_{co} - v_{ao} \end{cases} \quad (I.20)$$

En remplaçant (I.20) dans (I.19) on obtient :

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_{ao} \\ v_{bo} \\ v_{co} \end{bmatrix} \quad (I.21)$$

Des relations suivantes :

$$\begin{cases} v_{ao} = v_{an} + v_{no} \\ v_{bo} = v_{bn} + v_{no} \\ v_{co} = v_{cn} + v_{no} \end{cases} \quad (I.22)$$

On peut déduire le potentiel entre les points n et o :

$$v_{no} = \frac{1}{3} (v_{ao} + v_{bo} + v_{co}) \quad (I.23)$$

L'utilisation de l'expression (I.17) permet d'établir les équations instantanées des tensions simples en fonction des grandeurs de commande :

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{v_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix} \quad (I.24)$$

Les techniques de MLI ou (PWM) ont été l'objet de recherches intensives, un nombre important de méthodes, différentes de par leurs concepts et leurs performances ont été développées. Le choix d'une technique dépend du type de machine à commander, de la gamme de puissance, des semi-conducteurs utilisés pour l'onduleur et de la simplicité d'implantation de l'algorithme. Ce sont finalement des critères de coût et de performance qui vont déterminer ce choix. Les critères de performances permettent d'évaluer et de comparer les qualités des différentes techniques PWM. Ils se résument en : l'index de modulation, le facteur de distorsion harmonique et le spectre harmonique de courant (ou analyse FFT : Fast Fourier Transformation), l'harmoniques de couple, et les pertes à la commutation [44].

Parmi les nombreuses techniques PWM, deux d'entre elles sont considérées dans notre travail, la STPWM et la SVPWM.

I.4.4 La technique de modulation "Sinus-triangle" (STPWM)

La STPWM considérée est classique : il s'agit d'une modulante sinusoïdale d'amplitude A_m et de fréquence f_m , combinée à une porteuse triangulaire d'amplitude A_p de haute fréquence f_p , les angles de commutation de la tension d'entrée d'un pont sont situés aux intersections de la porteuse et de la modulante (figure I.5).

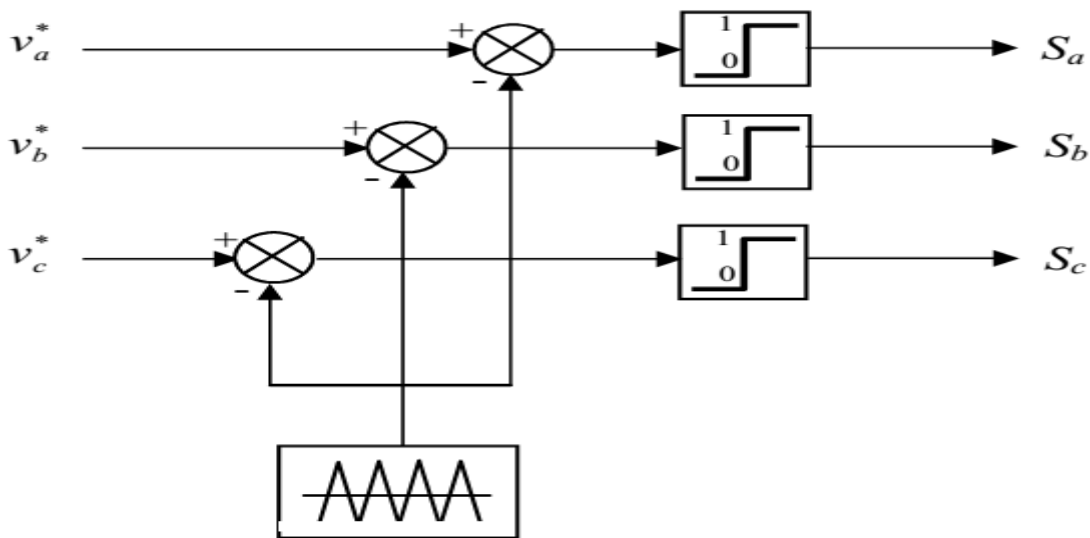


Figure I.5 : principe de la STPWM

L'index de modulation maximal m_{MAX} d'une PWM est un critère important puisqu'il montre la capacité d'une méthode à utiliser au maximum la tension du bus-continu. Pour la STPWM le maximum de l'index de modulation est (voir annexe A.1) :

$$m_{\max_spwm} = 0,7854 \quad (I.25)$$

Remarque : il existe d'autres méthodes pour améliorer cette valeur, parmi elles, celle qui consiste à injecter l'harmonique d'ordre trois, mais elles ne seront pas considérées dans ce travail.

Les figures.I.6 montrent le principe de la méthode STPWM ainsi que des différentes grandeurs appliquées à l'onduleur.

La figure.I.6 (a), représente les signaux de références issue de la commande v_{aref} , v_{bref} , v_{cref} et le signal de la porteuse v_p , chaque intersection génère les signaux de commande qui seront appliqués aux interrupteurs statiques.

Les figures.I.6 (b), et (c), montrent respectivement les tensions entre phases et ceux qui seront aux bornes de la machine.

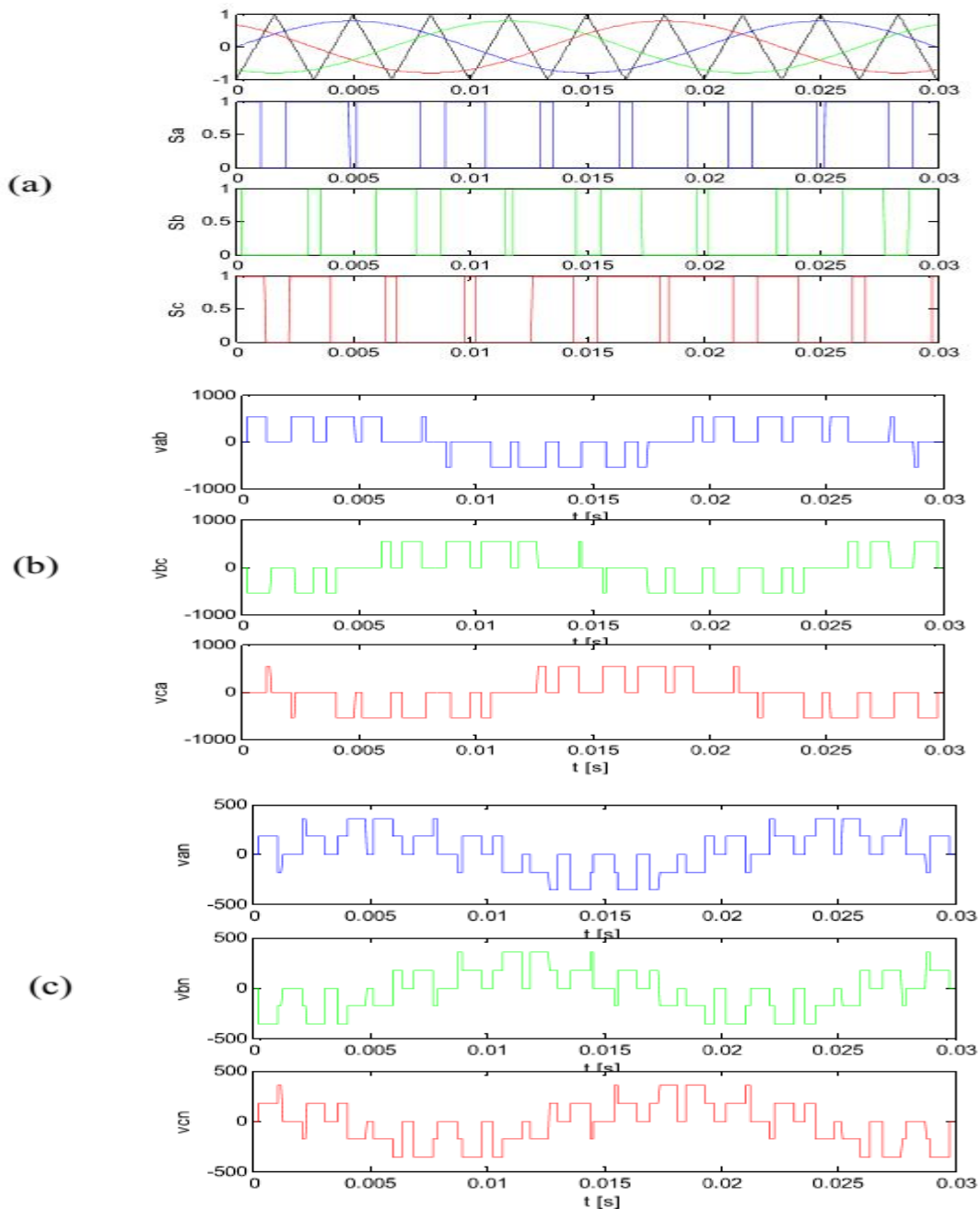


Figure I.6. : La technique de modulation "Sinus-triangle"

I.4.5 La technique de modulation vectorielle (SVPWM)

Cette technique est beaucoup sollicitée dans le domaine de la commande, ses effets sur les ondulations du courant et couple sont remarquables, c'est pourquoi elle est la plus utilisée par les chercheurs et industriels, elle permet de déterminer les séquences des allumages et extinctions des composants du convertisseur et de minimiser ainsi les harmoniques des tensions appliquées au moteur [42].

La technique de modulation vectorielle SVPWM se diffère de celle STPWM par le fait qu'au lieu d'employer un modulateur séparé pour chacune des trois phases, les tensions de référence sont données par un vecteur de contrôle global approximé sur une période de modulation T_z .

Elle est basée sur la représentation spatial du vecteur de tension dans le repère fixe $\alpha\beta$. Les huit états de commutation (figure.I.7) peuvent s'exprimer dans le plan $\alpha\beta$, par huit vecteurs de tension ($V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7$) parmi eux deux sont nuls V_0 et V_7 qui correspondent respectivement à S (000) et S (111), les autres sont appelés états actifs.

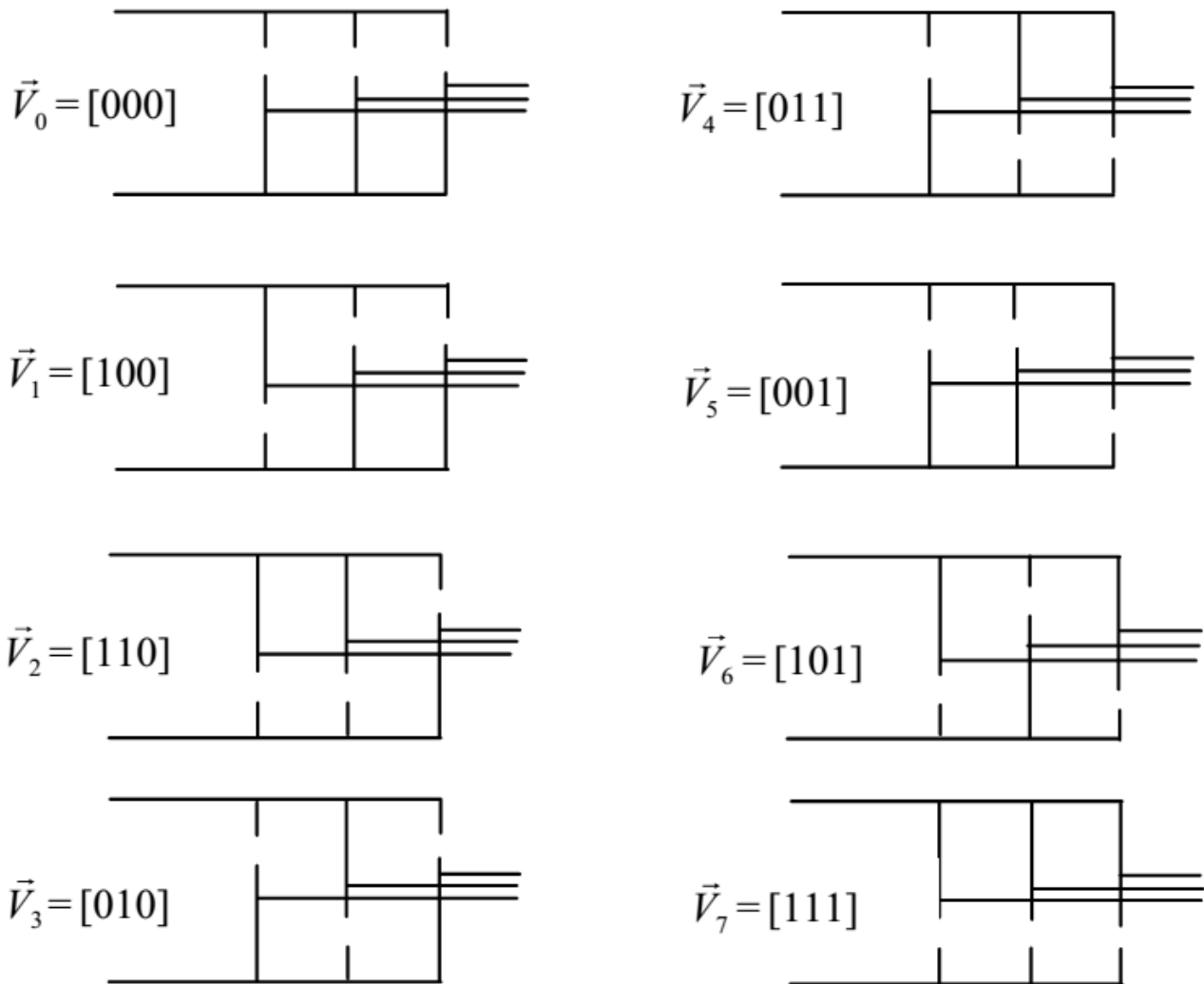


Figure I.7 : Les états de l'onduleur et les vecteurs des tensions correspondants

La représentation dans le plan $\alpha\beta$ de ces vecteurs equi-répartis tous les 60° , permet de déterminer un hexagone (figure.I.11) à l'intérieur duquel le vecteur tension de référence V_{ref} doit s'y trouver pour éviter la saturation de la grandeur de commande.

$$\vec{v}_{ref} = v_\alpha + jv_\beta = \frac{2}{3}(v_{ao} + av_{bo} + a^2v_{co}) = \frac{2}{3}v_{dc}(s_a + as_b + a^2s_c) \quad (I.26)$$

Avec :

$$a = e^{j\frac{2\pi}{3}} = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$$

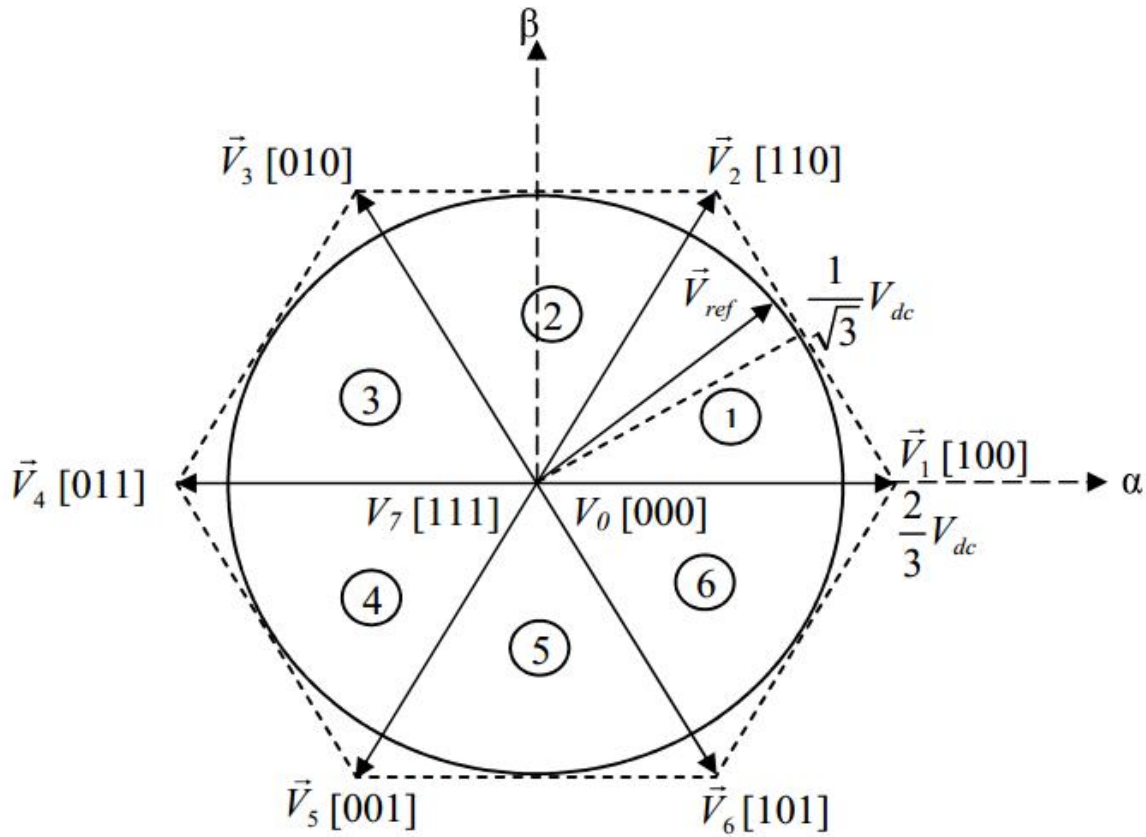


Figure I.8 : Représentation des états de l'onduleur dans le repère $\alpha\beta$, et les différents secteurs

Les vecteurs tensions des six états actifs, peuvent s'écrire en fonction des portions de l'hexagone appelés secteur k , où $k = 1, \dots, 6$:

$$\vec{V}_k = \frac{2}{3}V_{dc}e^{j(k-1)\frac{\pi}{3}} \quad (I.27)$$

Chaque vecteur de référence V_{ref} dans l'hexagone peut être exprimé comme combinaison de deux vecteurs tensions actifs adjacents et de vecteurs tension d'états nuls.

En se reportant à la figure.I.8, si le vecteur de référence est dans le secteur k , les vecteurs actifs adjacents sont V_k et V_{k+1} (avec $k+1 = 1$ pour $k = 6$).

De la figure.I.11 on voit bien que la tension de référence atteint les limites de l'hexagone, d'où l'index de modulation qui vaut (voir annexe A.2) :

$$m_{\max_svpwm} = 0,9069 \quad (I.28)$$

Le tableau.I.1 résume les définitions suscitées : Notons que toutes les valeurs des tensions doivent être multipliées par V_{dc} .

Vecteur d'état	Valeur en $\alpha\beta$	Tensions simples			Tensions composées		
$V_i(s_a s_b s_c)$	V_i	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}
$V_0(0\ 0\ 0)$	0	0	0	0	0	0	0
$V_1(1\ 0\ 0)$	e^{j0}	2/3	-1/3	-1/3	1	0	1
$V_2(1\ 1\ 0)$	$e^{j\frac{2\pi}{3}}$	1/3	1/3	1/3	0	1	-1
$V_3(0\ 1\ 0)$	$e^{j\frac{\pi}{3}}$	-1/3	2/3	-1/3	-1	1	0
$V_4(0\ 1\ 1)$	$e^{j\frac{4\pi}{3}}$	-2/3	1/3	1/3	-1	0	1
$V_5(0\ 0\ 1)$	$e^{j\frac{5\pi}{3}}$	-1/3	-1/3	2/3	0	-1	1
$V_6(1\ 0\ 1)$	$e^{j\pi}$	1/3	-2/3	1/3	1	-1	0
$V_7(1\ 1\ 1)$	0	0	0	0	0	0	0

Tableau I0.1. Valeurs des tensions correspondantes

La méthode SVPWM peut être implémentée par les étapes suivantes :

- Déterminer V_α , V_β , et par conséquent V_{ref} , et l'angle α .
- Déterminer les temps d'application des états de l'onduleur.
- Déterminer les séquences d'impulsions pour chaque bras de l'onduleur.

Détermination de V_α , V_β , V_{ref} et l'angle (α)

Les tensions V_α , V_β , sont issues directement de la transformation de Clarke comme suit :

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} \quad (I.29)$$

$$\Rightarrow v_{ref} = \sqrt{v_{\alpha}^2 + v_{\beta}^2} \quad (I.30)$$

$$\alpha = \tan^{-1} \left(\frac{v_{\alpha}}{v_{\beta}} \right) \quad (I.31)$$

Où : v_{ref} est le module du vecteur tension de référence, et α son argument.

Calcul des temps d'application des états de l'onduleur

La partie centrale de la stratégie SVPWM est la détermination des temps qui doivent être attribués à chaque vecteur de tension durant chaque cycle de modulation ou période d'échantillonnage T_z . À chaque période de commutation de l'onduleur le vecteur V_{ref} , projeté sur ses deux vecteurs adjacents assure le calcul des temps de commutation (figure.I.9).

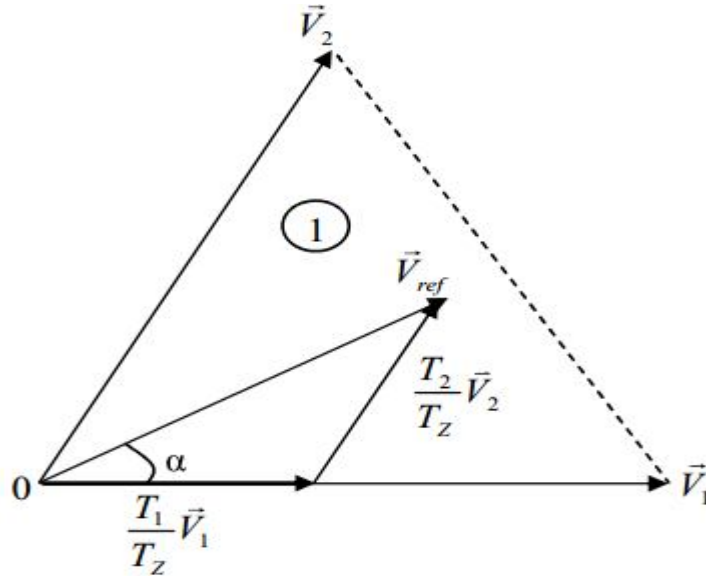


Figure I. 9 : La tension de référence comme combinaison de deux vecteurs adjacents dans le secteur 1

Dans ce qui suit T_k dénote la demi période de l'application de V_k (l'état on), T_0 est le demi temps de l'état nul, la somme des temps de conduction T_0 , T_k , et T_{k+1} , doit être égale à la demi période de commutation de l'onduleur T_z , on aura alors :

$$T_0 + T_k + T_{k+1} = \frac{T_z}{2} \quad (I.55)$$

$$\int_0^{\frac{T_z}{2}} \vec{V}_{ref} T_z = \int_0^{\frac{T_0}{2}} \vec{V}_0 dt + \int_{\frac{T_0}{2}}^{\frac{T_0}{2} + T_k} \vec{V}_k dt + \int_{\frac{T_0}{2} + T_k}^{\frac{T_0}{2} + T_k + T_{k+1}} \vec{V}_{k+1} dt + \int_{\frac{T_0}{2} + T_k + T_{k+1}}^{\frac{T_z}{2}} \vec{V}_7 dt \quad (I.66)$$

En tenant compte que $V_0 = V_7 = 0$, et que V_{ref} est constant pendant la période de commutation, et du fait que V_k et V_{k+1} sont aussi constants, la relation (I.66) est réduite à ceci :

$$\vec{V}_{ref} \frac{T_z}{2} = \vec{V}_k T_k + \vec{V}_{k+1} T_{k+1} \quad (I.67)$$

En exprimant le vecteur tension V_{ref} par ses composantes en $\alpha\beta$ nous aurons :

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} \frac{T_z}{2} = \frac{3}{2} V_{dc} \left(T_k \begin{bmatrix} \cos\left(\frac{(k-1)\pi}{3}\right) \\ \sin\left(\frac{(k-1)\pi}{3}\right) \end{bmatrix} + T_{k+1} \begin{bmatrix} \cos\left(\frac{k\pi}{3}\right) \\ \sin\left(\frac{k\pi}{3}\right) \end{bmatrix} \right) = \frac{3}{2} V_{dc} \begin{bmatrix} \cos\left(\frac{(k-1)\pi}{3}\right) & \cos\left(\frac{k\pi}{3}\right) \\ \sin\left(\frac{(k-1)\pi}{3}\right) & \sin\left(\frac{k\pi}{3}\right) \end{bmatrix} \begin{bmatrix} T_k \\ T_{k+1} \end{bmatrix} \quad (I.68)$$

Où : k est déterminé à partir de la relation α .

Sachant que :

$$\frac{(k-1)\pi}{3} \leq \alpha \leq \frac{k\pi}{3} \quad (I.69)$$

La solution du système (I.68) conduit au résultat suivant :

$$\begin{bmatrix} T_k \\ T_{k+1} \end{bmatrix} = \frac{\sqrt{3} T_z}{2 V_{dc}} \begin{bmatrix} \sin\left(\frac{k\pi}{3}\right) & \cos\left(\frac{k\pi}{3}\right) \\ -\sin\left(\frac{(k-1)\pi}{3}\right) & -\cos\left(\frac{(k-1)\pi}{3}\right) \end{bmatrix} \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} \quad (I.70)$$

Le temps T_0 est défini à partir de la relation (I.55) :

$$T_0 = \frac{T_z}{2} - (T_k + T_{k+1}) \quad (I.71)$$

Sachant que :

$$\vec{V}_{ref} = V_{ref} e^{j\alpha} = V_{ref} (\cos(\alpha) + j \sin(\alpha))$$

le système (I.38) devient :

$$\begin{bmatrix} T_k \\ T_{k+1} \end{bmatrix} = \frac{\sqrt{3} V_{ref}}{2 V_{dc}} T_z \begin{bmatrix} \sin\left(\frac{k\pi}{3}\right) & \cos\left(\frac{k\pi}{3}\right) \\ -\sin\left(\frac{(k-1)\pi}{3}\right) & -\cos\left(\frac{(k-1)\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \quad (I.72)$$

Avec la définition de l'index de modulation m , le calcul des temps d'application T_k , T_{k+1} , n'exige pas la connaissance de la tension adoptée du bus-continu, mais dépend seulement de l'index modulation désiré. La substitution de cette définition dans (I.39) mène à :

$$\begin{bmatrix} T_k \\ T_{k+1} \end{bmatrix} = m \frac{\sqrt{3}}{\pi} T_z \begin{bmatrix} \sin\left(\frac{k\pi}{3}\right) & \cos\left(\frac{k\pi}{3}\right) \\ -\sin\left(\frac{(k-1)\pi}{3}\right) & -\cos\left(\frac{(k-1)\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \quad (I.73)$$

Si par exemple : $0 \leq \alpha \leq \frac{\pi}{3}$ alors le vecteur V_{ref} se situe au secteur 1, donc (I.40) donne :

$$\left\{ \begin{array}{l} T_1 = m \frac{\sqrt{3}}{\pi} T_z \sin\left(\frac{\pi}{3} - \alpha\right) \\ T_2 = m \frac{\sqrt{3}}{\pi} T_z \sin(\alpha) \\ T_0 = \frac{T_z}{2} - (T_1 + T_2) \end{array} \right. \quad (I.74)$$

Détermination des séries d'impulsions pour chaque bras de l'onduleur

Afin d'obtenir de performance optimum en harmoniques (moins d'harmoniques en sortie), et une fréquence de commutation minimum pour chaque composant de puissance, l'ordre des états est arrangé tels que la transition d'un état au prochain est exécuté en commutant seulement un bras de l'onduleur.

Pour cela, le modèle d'impulsion symétrique ou méthode à états-nuls symétriques, est utilisée, ainsi pour les secteurs impairs ($k = 1, 3, 5$), la séquence est la suivante : $V_0 V_k V_{k+1}, V_7 V_{k+1} V_k V_0$, et pour les secteurs pairs ($k = 2, 4, 6$), la séquence est : $V_0 V_k V_{k+1}, V_7 V_{k+1} V_k V_0$, figure I.10 [55].

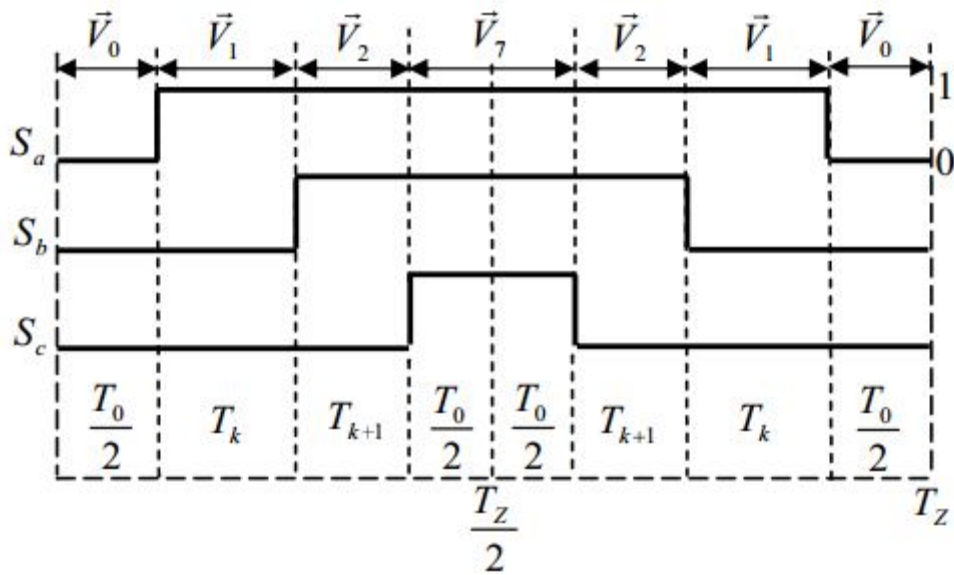


Figure I.10: Signaux de commande dans le cas où V_{ref} est au secteur 1

Les figure I.11 illustrent l'analyse qui vient d'être décrit pour chaque secteur.

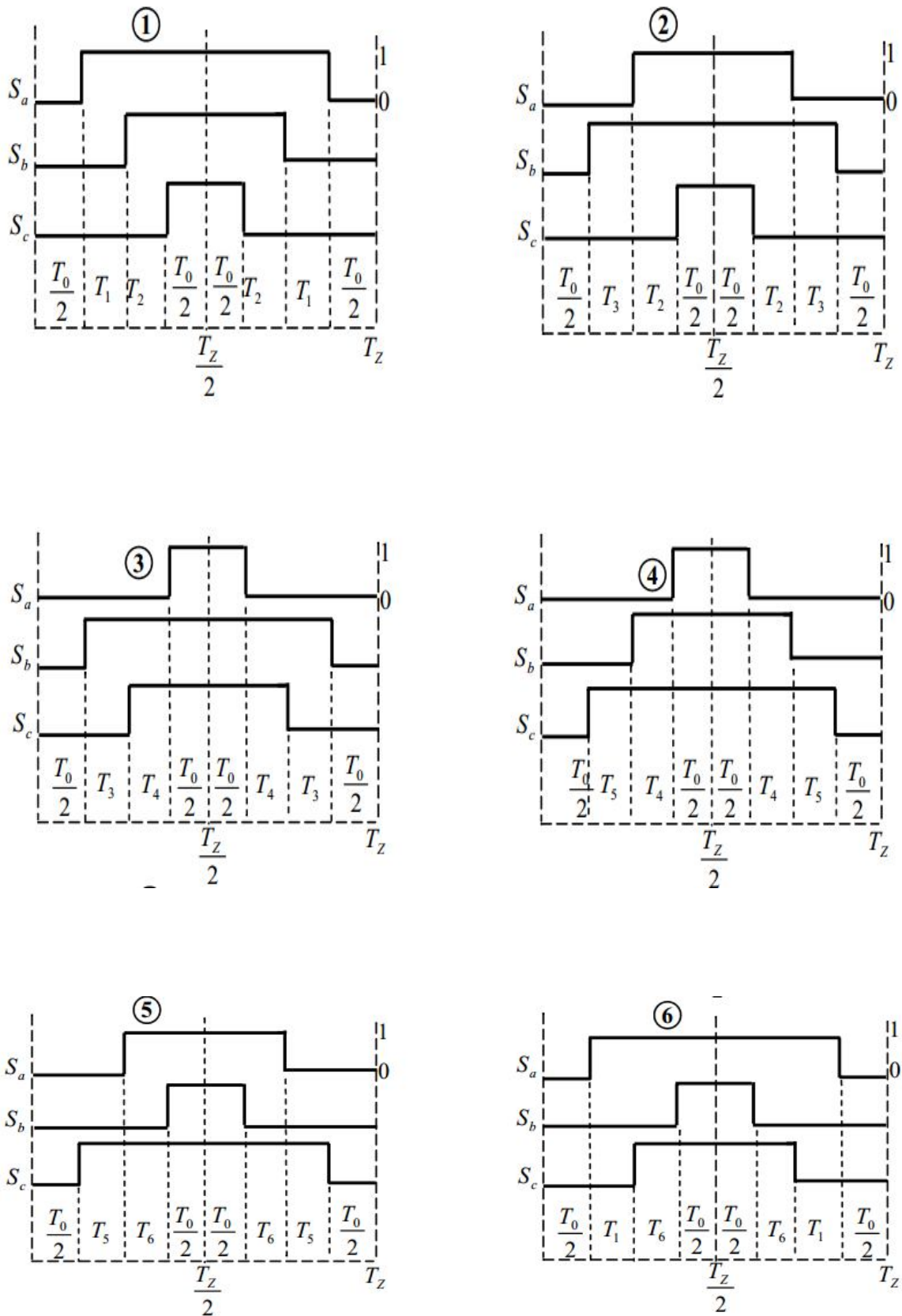


Figure I.11 : Signaux de commande pour chaque secteur

On peut apercevoir sur les figures (1.11, et 1.13) que les résultats de simulations coïncident parfaitement avec l'analyse théorique étendue de la méthode SVPWM. Notons que se sont quasiment les mêmes résultats obtenus par [56].

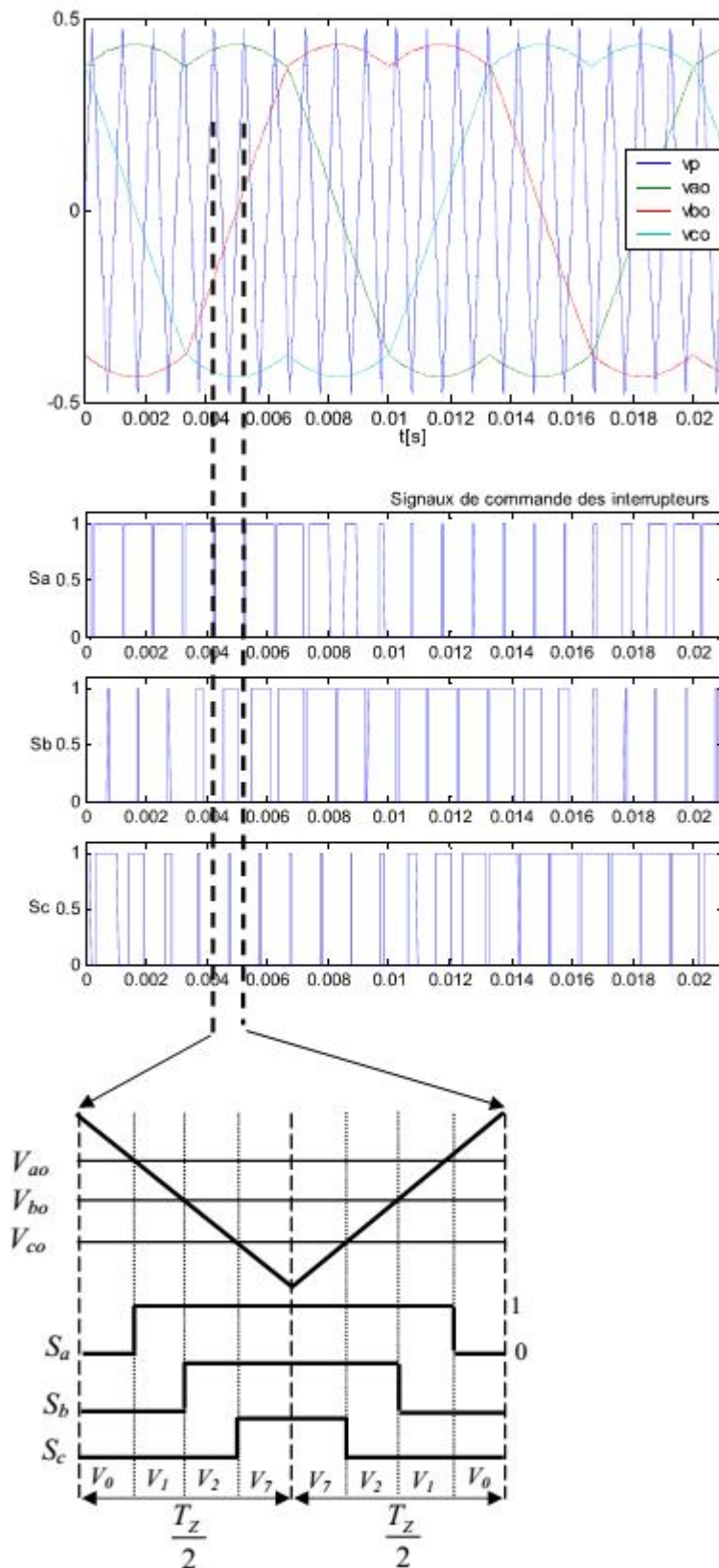


Figure I.12 : Comparaison entre théorie et simulation des signaux de commandes des interrupteurs obtenues par SVPWM.

I.5 Conclusion

Dans ce chapitre, on a commencé par un état de l'art sur les commandes prépondérantes sur le marché actuel, notamment la commande scalaire, la FOC et la DTC, pour faire par la suite une étude comparative entre ces deux dernières qui sont très compétitives et qui attirent l'attention des chercheurs afin d'aboutir à une commande performante qui réponde aux exigences industrielles modernes.

Ensuite, on a présenté le modèle mathématique de la MAS en triphasée, puis on a procédé aux transformations dans les différents repères de Park, ce qui aboutit aux modèles simplifiés.

La représentation en modèle d'état permet d'envisager la commande de la MAS par les techniques qui seront traitées aux chapitres suivants, notamment la technique par orientation du flux rotorique IRFOC et la DTC, et aussi pour l'étude et l'élaboration des observateurs.

Les résultats de simulation ont montré la validité du modèle mathématique de la machine asynchrone.

La STPWM est la plus simple des techniques PWM à base de porteuse, tant du point de vue de son concept que de son implantation (analogique ou numérique). Elle souffre néanmoins d'une sous-utilisation du bus continu, du fait que son index de modulation maximal est $m_{MAX} = 0,785$. Alors que pour la SVPWM il atteint $m_{MAX} = 0,907$, d'autant plus que les avancées technologiques récents, en matière de calculateurs digitaux (DSP, microprocesseurs et microcontrôleurs), ont rendu son implantation industrielle assez simple.

Une étude comparative sera faite sur ces deux méthodes, avec une analyse de leurs impacts sur la commande vectorielle sera traitée aux chapitres suivants.

Chapitre 02 : Commande scalaire

Chapitre 2 : Commande scalaire

II.1 Introduction

Historiquement, elle constitue la première méthode de variation de vitesse d'un MAS. Et malgré le développement de la commande vectorielle des systèmes à MAS, la commande scalaire est toujours utilisable dans plusieurs applications (pompes, ventilateurs).

Le principe de ces méthodes est d'agir sur la fréquence et l'amplitude des courants ou tensions d'entrée afin de faire varier l'amplitude et la vitesse de rotation des vecteurs spatiaux (flux, tension, etc.), et donc faire varier le couple et la vitesse de rotation du moteur.

Avec l'algorithme V/f , on s'arrange pour faire évoluer l'amplitude et la fréquence de la tension d'alimentation du moteur d'une telle façon que leur rapport reste constant, ce qui permet de faire varier la vitesse du moteur pour un couple résistant constant.

Pour mettre en œuvre un tel contrôle, le plus simple est de fournir la fréquence comme consigne du bloc de contrôle, (figure II.1). Celui-ci calcule alors la tension qui sert de consigne à son tour au bloc MLI. [57]

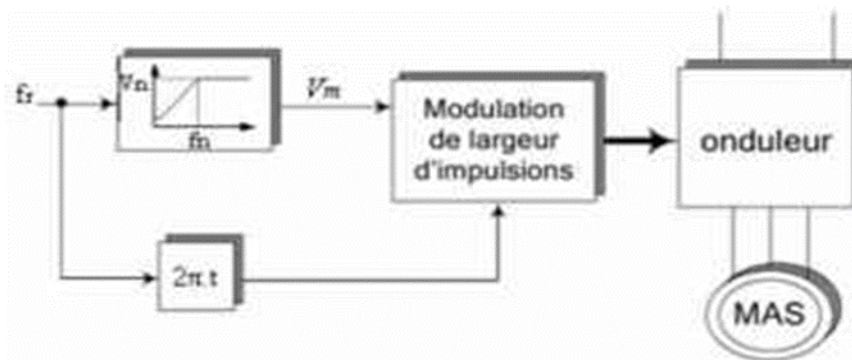


Figure II.1 : Diagramme de la commande de la vitesse de la machine asynchrone en boucle ouverte en V/F constante

II.2 Commande scalaire en tension

Dans une commande scalaire en tension, la variable contrôlée, est la tension statorique ; La fréquence ω_s est déterminée à partir de l'autopilotage fréquentiel $\omega_s = p\Omega \pm \omega_r$, et la valeur efficace est calculée de manière à garder le flux constant.

II.2.1 Contrôle du flux et réglage du couple

Lorsque le glissement est faible, l'expression du couple électromagnétique C_e , qui découle du schéma équivalent à fuites totalisées au rotor se devient :

$$C_e = 3p(\Phi_s)^2 \quad (II.1)$$

L'inductance de fuite N_f est négligée de sorte que la valeur efficace du flux a pour expression :

$$\Phi_s = \frac{V_s}{\sqrt{\left(\frac{R_s}{L_s}\right)^2 + \left(1 + g \frac{R_s}{R}\right)^2} \omega_s^2} \quad (\text{II.2})$$

Cette expression du flux peut être simplifiée si le glissement est suffisamment faible pour considérer $g R_s \ll R$:

$$\Phi_s = \frac{V_s}{\sqrt{\left(\frac{R_s}{L_s}\right)^2 + \omega_s^2}} \quad (\text{II.3})$$

Et si, de plus, la résistance statorique est telle que $R_s \ll L_s \omega_s$ (hypothèse valable surtout pour les machines de puissance élevée), alors l'expression du flux devient très simple :

$$\Phi_s = \frac{V_s}{\omega_s} \quad (\text{II.4})$$

Et le couple permanent devient :

$$C_e = 3p \left(\frac{V_s}{\omega_s} \right) * \left(g \frac{\omega_s}{R} \right) \quad (\text{II.5})$$

Le flux peut être maintenu sensiblement constant en adoptant une loi $V_s/\omega_s = \text{constante}$ qui conduit à un fonctionnement satisfaisant sauf aux vitesses faibles ; en effet, les chutes de tension, négligeables aux grandes vitesses, doivent impérativement être prises en compte aux faibles vitesses, notamment pour les machines de faible puissance. Dans ce dernier cas, le maintien du flux à une valeur constante peut être assuré en utilisant la loi (II.2).

La valeur du flux est choisie en fonction des caractéristiques nominales de la machine, de manière à en optimiser le fonctionnement dans une large plage de vitesse. Toutefois, il est possible d'utiliser la machine à tension constante et fréquence variable, si les exigences de couple le permettent, à des vitesses supérieures à la valeur habituellement appelée vitesse nominale. Dans ces conditions, le flux Φ_s évolue sensiblement en fonction de l'inverse de la vitesse.

Lorsque, dans l'équation (II.1), le flux est maintenu constant, le contrôle du couple est obtenu par action sur la pulsation rotorique ω_r . Ce contrôle oblige à autopiloter la machine de manière à transformer la pulsation rotorique en grandeur indépendante, donc en grandeur de réglage.

La loi d'autopilotage fréquentiel découle de la relation de glissement :

$\omega_s = p\Omega \pm \omega_r$; Le signe caractérisant la nature du fonctionnement moteur (+) ou générateur (-).

II.2.2 Structure de régulation de vitesse:

Le schéma, donné en figure 1.1, montre alors l'application de ce principe dans une régulation de vitesse pour laquelle la pulsation rotorique ω_{ref} est élaborée par le correcteur de vitesse $C_v(s)$. Celui-ci est à action proportionnelle et intégrale, avec action dérivée filtrée dans la boucle de retour. Ce choix est imposé par la nature du système où la volonté d'annuler l'erreur statique conduit à une

double intégration, facteur de désamortissement si des précautions ne sont pas prises.

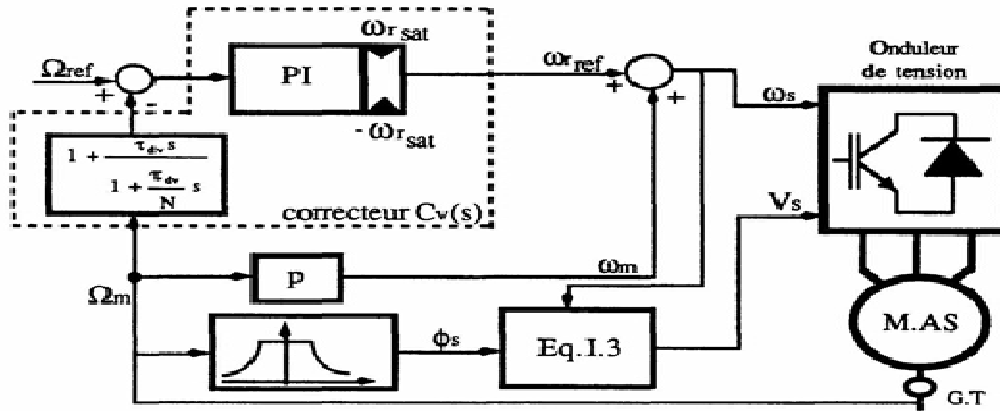


Figure II.2 : **Structure de contrôle de vitesse en commande scalaire en tension.**

II.3 Commande scalaire en courant

La grandeur de réglage est maintenant le courant statorique. Son amplitude I_s est contrôlée de manière à garder le courant magnétisant I_{sm} image du flux, constant, la fréquence (ω_s) est déterminée, comme précédemment, par la formule d'autopilotage fréquentiel.

II.3.1 Contrôle du courant magnétisant et Réglage du couple

A partir du schéma équivalent le couple peut s'écrire en fonction de la valeur efficace du courant :

$$C_e = 3p \frac{L_s^2}{R} I_s^2 \frac{\omega_r}{1 + (T_r \omega_r)} \quad (II.6)$$

Avec : $T_r = (N_f + L_8) / R = L_r / R_r$, constante de temps rotorique.

En faisant intervenir le courant magnétisant I_{sm} :

$$\bar{I}_{sm} = \frac{1 + j \frac{N_f}{R} \omega_r}{1 + j \omega_r T_r} \bar{I}_s \quad (II.7)$$

Qui devient, si on considère que : $\frac{N_f}{R} \omega_r \ll 1$ alors :

$$\bar{I}_{sm} = \frac{1}{1 + j \omega_r T_r} \bar{I}_s \quad (II.8)$$

L'expression du couple devient alors :

$$C_e = 3p \frac{L_s^2}{R} (I_{sm}^2) \omega_r \quad (II.9)$$

Si on maintient, cette fois, le courant magnétisant constant I_{sm} le contrôle du couple est obtenu, comme dans la structure de commande en tension, par action sur la pulsation rotorique ω_r . Le réglage du courant magnétisant peut être assuré en utilisant l'expression simplifiée (II. 8) :

$$I_{\text{sm}} = \frac{I_s}{\sqrt{1 + (T_r \omega_r)^2}} \quad (\text{II.10})$$

II.3.2 Structure de régulation de vitesse

Le schéma, donné en figure (II.3), montre alors l'application de ce principe dans une régulation de vitesse. Les deux grandeurs de réglage sont la pulsation et la valeur efficace des courants statoriques.

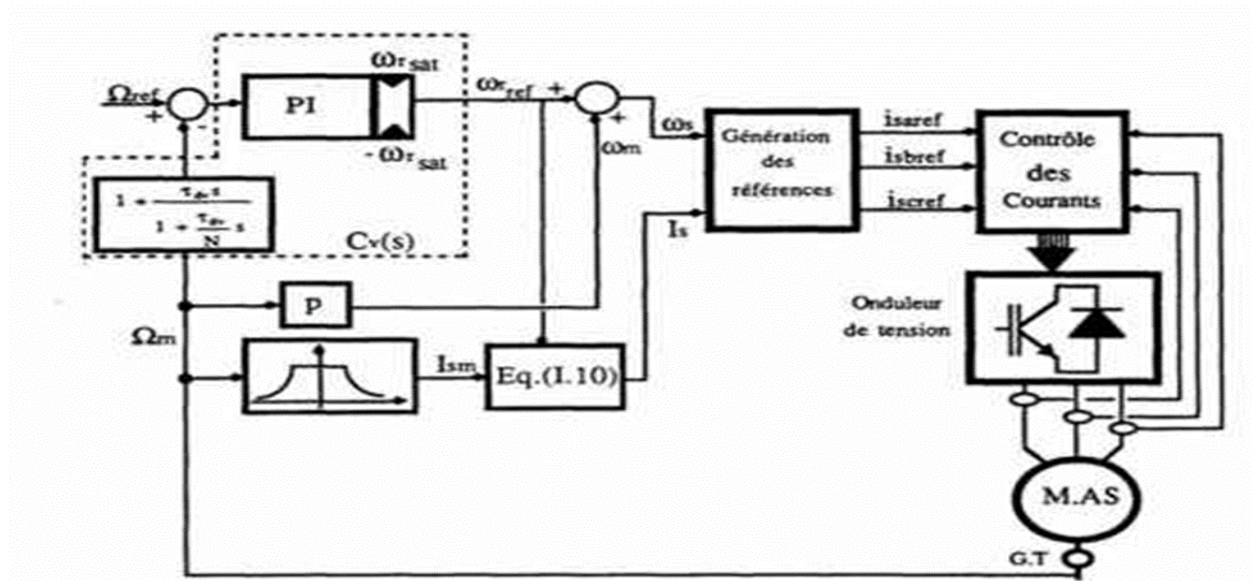


Figure II.3 : Structure de contrôle de vitesse en commande scalaire en courant.

A partir de la pulsation statorique w_s élaborée par l'autopilotage, et la valeur efficace I_s , élaborée par le contrôle du courant magnétisant, on génère trois références qui seront comparées, dans le bloc contrôle des courants, aux courants réels dans la machine, pour délivrer les signaux de commande de l'onduleur.

II.3.3 Stratégies de contrôle de courant

Dans la pratique, les méthodes les plus classiques pour asservir le vecteur courant à sa référence sont basées sur la comparaison de trois courants réels i_{sa} , i_{sb} , i_{sc} à trois courants de référence i_{saref} , i_{sbref} et i_{scref} . Constituant un système triphasé équilibré. Chaque méthode de contrôle possède des caractéristiques propres qui confèrent au montage des comportements différents. Il est cependant possible de distinguer deux types de stratégies :

- les stratégies effectuant un traitement de l'erreur par des contrôleurs de type " tout ou rien " : contrôle par hystérésis.
- Les stratégies qui utilisent des régulateurs linéaires : contrôle par MLI.

II.3.3.1 Contrôle des courants par hystérésis

Dans ce type de commande, chaque courant de phase est comparé au courant de référence et un régulateur à hystérésis commande directement, de manière complémentaire, les deux transistors du bras correspondant de l'onduleur, Figure(II.4).

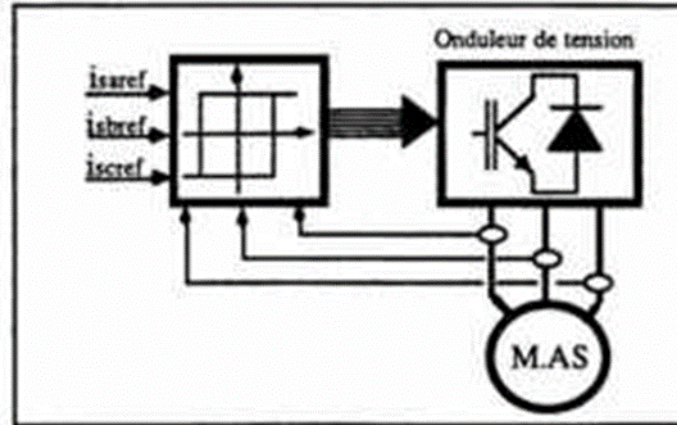


Figure II.4 : Contrôle par hystérésis.

Ce mode de contrôle est très simple et présente une grande dynamique. Les ondulations de courant autour de la valeur désirée (au centre de la plage d'hystérésis) sont fixées par la largeur de la plage d'hystérésis choisie. Cependant, cette solution présente quelques inconvénients :

- L'indépendance des 3 boucles conduit à l'apparition d'une période de roue libre qui provoque un dépassement des limites fixées par la plage d'hystérésis.
- La modulation disparaît pour les faibles valeurs de courant. Toutefois, le comportement peut être amélioré en mettant en œuvre des boucles de courant couplées.

Le contrôle des courants par hystérésis est caractérisé par une fréquence de modulation variable avec les conditions de fonctionnement. En moyenne, cette fréquence est d'autant plus élevée que la bande d'hystérésis est étroite. A fréquence élevée, les pertes de commutation augmentent et les limites de commutation des interrupteurs peuvent être atteintes. [58]

Dans cette partie, nous avons analysé les performances des principales structures de commande en vitesse de l'ensemble onduleur-moteur asynchrone en marche triphasée. Ainsi, nous avons d'abord montré que la commande scalaire en courant, dont l'algorithme de commande est facile à implémenter, présente une bonne dynamique en poursuite ainsi qu'en régulation (réponse à un couple de charge) ; cependant, ces qualités sont fortement réduites lors du fonctionnement en basse vitesse.

Cette structure exigeant le contrôle des courants statoriques, nous avons alors analysé les deux principales méthodes d'asservissement de courants que sont : la méthode par hystérésis et la méthode M.L.I. La première est simple à mettre en œuvre en analogique ; la seconde peut être adaptée pour une réalisation de l'ensemble de la commande par les techniques intelligentes comme le réseau de neurones artificielles.

II.3.3.2 Contrôle des courants par MLI

L'erreur entre le courant réel et le courant de référence de chaque phase est appliquée à l'entrée d'un régulateur linéaire. La commande de la branche correspondante de l'onduleur est obtenue en comparant la sortie de ce régulateur avec un signal de modulation à haute fréquence commun aux

trois phases (fig.II.5). Les sorties des trois correcteurs donnent l'image du système de tensions triphasées appliquées à la machine [58].

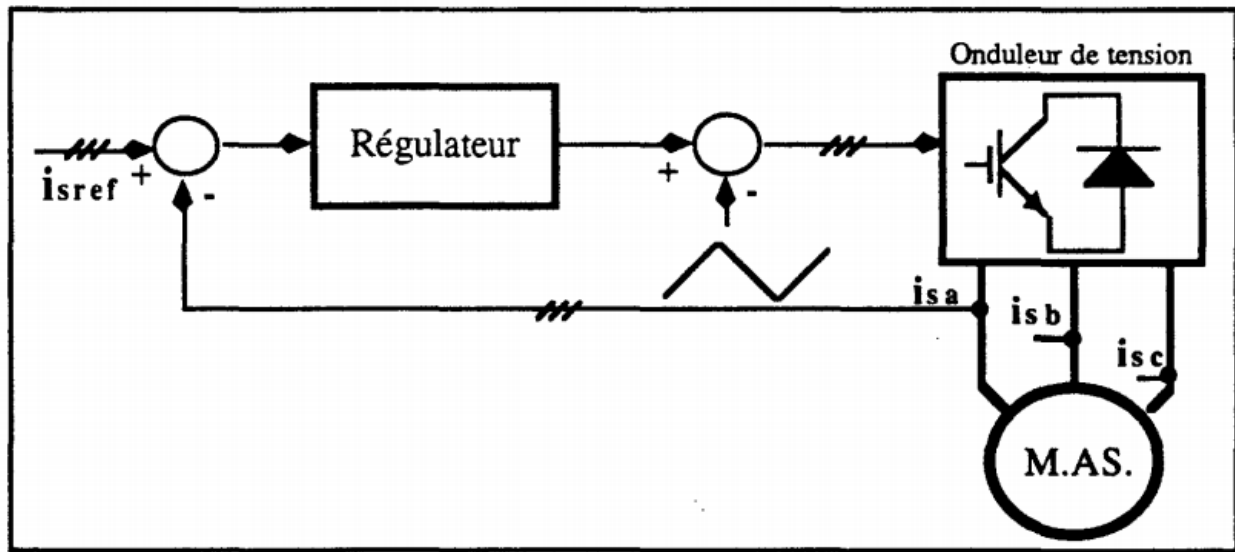


Fig II.5 Contrôle des courants par MLI

III.4 Conclusion sur la commande scalaire

Dans cette partie nous avons présenté deux types de structure de contrôle de couple issues des équations en régime permanent : Commande scalaire en tension et commande scalaire avec contrôle des courants statoriques. Cette dernière semble plus adaptée à un bon fonctionnement de la régulation car elle permet, d'une part, un meilleur contrôle du couple et, d'autre part, une surveillance des courants statoriques protégeant ainsi les éléments de puissance de l'onduleur contre les surcharges ; cette disposition va dans le sens d'une sûreté de fonctionnement accrue.

Cependant, à faible vitesse, les performances d'une telle structure se dégradent et ne sont plus satisfaisantes, même avec l'utilisation, dans la boucle de retour, d'une action dérivée qui stabilise le système mais qui réduit sensiblement les performances de sa dynamique, de poursuite ou de régulation.

Pour obtenir une meilleure dynamique en poursuite et en régulation, ainsi qu'une meilleure stabilité de fonctionnement en basse vitesse, le contrôle du couple transitoire s'impose, puisqu'il s'agit de la grandeur mécanique dont dépend le contrôle de la vitesse en vertu même de la relation fondamentale de la mécanique.

Chapitre 03

Réseaux de neurones artificiels

Chapitre 3 : réseau de neurones artificiels

III.1 Introduction

Malgré la constante augmentation de la puissance des calculateurs, plusieurs opérations restent toujours difficiles à effectuer. Parmi ces opérations on trouve la modélisation des systèmes, le contrôle des processus, le filtrage ou la reconnaissance des signaux [26]. Les techniques des réseaux de neurones relèvent d'une telle approche. comprendre les principes selon lesquels les systèmes biologiques traitent l'information et s'en inspirer pour élaborer de nouvelles techniques en sciences de l'ingénieur

Tout a commencé en 1943, lorsque Mc Culloch et Pitts ont inventé le premier "neurone artificiel", quelques années plus tard, en 1949, Hebb propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques. Le premier réseau de neurones artificiel apparaît en 1958, grâce aux travaux de Rosenblatt qui conçoit le fameux perceptron. En 1982 Hopfield démontre tout l'intérêt d'utiliser des réseaux récurrents (dits "feed-back"), et forme l'algorithme de rétro propagation de l'erreur, qui donne un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron.

Ce chapitre se compose de deux parties principales :

Dans la première partie, nous présenterons les principales notions de base relatives aux réseaux de neurones telles que leurs architectures, les fonctions d'activation et l'apprentissage etc.

Dans la seconde partie, nous appliquons la commande par un réseau des neurones formels sur notre système étudié.

A la fin, nous tirerons la conclusion nécessaire.

III.2 Définition

Inspirés des réseaux de neurones biologiques, un réseau de neurones est un système composé de plusieurs unités de calcul simples fonctionnant en parallèle, dont la fonction est déterminée par la structure du réseau, la solidité des connexions, et l'opération effectuée par les éléments ou nœuds (neurones) [59].

III.3 Neurone biologique

Le cerveau humain possède deux hémisphères latérales reliées par le corps calleux et d'autres ponts axonaux, il pèse moins de deux kilogrammes et contient mille milliards de cellules, dont 100 milliards sont des neurones constitués en réseaux. Les neurones sont des cellules nerveuses, décomposables en 4 parties principales figure (III.1). [60]

- Les dendrites, sur lesquelles les autres cellules entrent en contact synaptique, c'est par les dendrites que se fait la réception des signaux.
- Le corps de la cellule, c'est l'unité de traitement.
- L'axone, où passent les messages accumulés dans le corps de la cellule : l'envoi de l'information se fait par l'axone.
- Les synapses par lesquelles la cellule communique avec d'autres cellules, ce sont des points de connexion par où passent les signaux de la cellule.

Un neurone stimulé envoie des impulsions électriques ou potentielles d'action à d'autres neurones. Ces impulsions se propagent le long de l'axone unique de la cellule. Au point de contact entre neurones, les synapses, ces impulsions sont converties en signaux chimiques. Quand l'accumulation des excitations atteint un certain seuil, le neurone engendre un potentiel d'action, d'une amplitude d'environ 100 mV et pendant une durée de 1 ms.

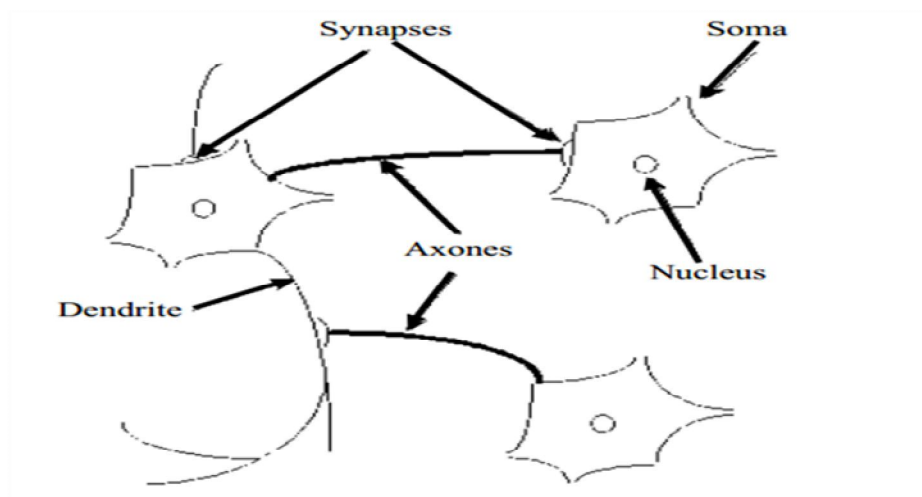


Figure. III.1. Schéma d'un réseau de neurones biologiques.

III.4 Le neurone formel

Le neurone formel est une modélisation mathématique très simple qui a pour but de reproduire le raisonnement « intelligent » d'une manière artificielle, tel que la sommation et la comparaison. Chaque neurone formel calcule une sortie unique sur la base des informations qu'il reçoit.

Le premier modèle du neurone formel a été présenté par Culloch et Pitts. D'une façon générale, un neurone formel est un élément de traitement qui fait une somme des potentiels d'action venant de n entrées $x_1, \dots, x_n$ pondérée par des coefficients, dits poids synaptique (les potentiels des entres

ont des valeurs numériques). Si la somme de ces potentiels dépasse un certain seuil (fonction de seuillage F, ou fonction d'activation), le neurone est activé et transmet une réponse dont la valeur est celle de son activation. Si le neurone n'est pas activé, il transmet rien Figure (III.2).

chaque connection est associée un poids

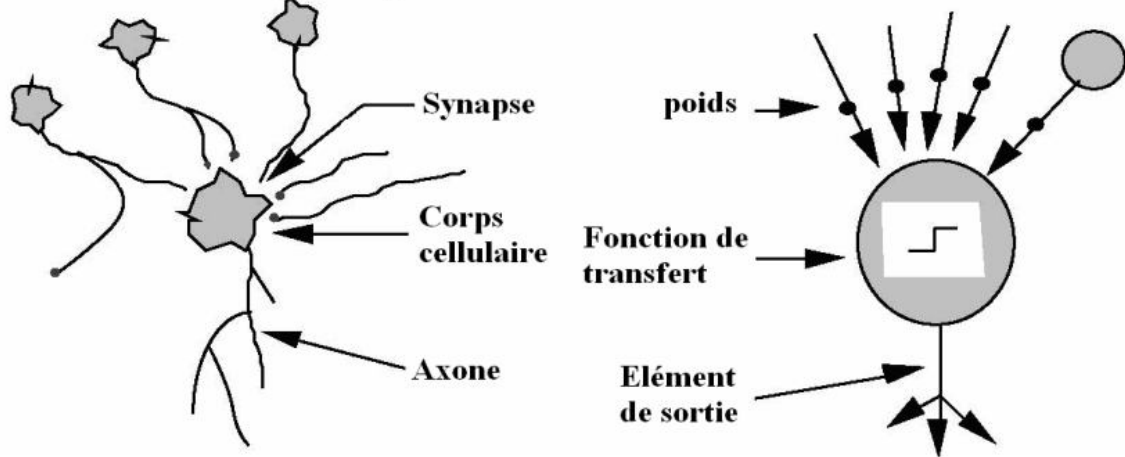


Figure III.2 Mise en correspondance neurone biologique / neurone artificiel

III.4.1 Structure

La Figure (III.3) montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associée un poids [61].

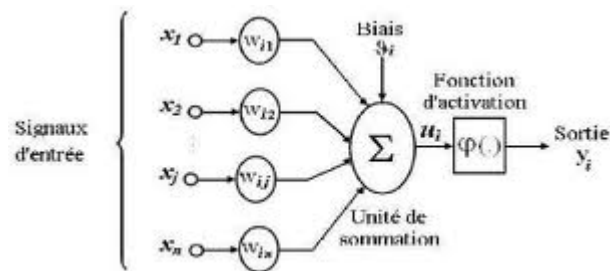


Figure III.3 : Neurone formel

Le neurone formel est donc un automate booléen dont la sortie y est tel que :

$$y = \begin{cases} 1 & \text{si } S \succ \beta \\ 0 & \text{si } S \prec \beta \end{cases} \quad (\text{III.1})$$

$$\text{Avec : } y=f(x) \quad \text{et} \quad S = \sum_{i=1}^n w_i x_i \quad (\text{III.2})$$

Les grandeurs intervenant dans les relations (III.2) et (III.3) sont définies comme suit :

x_i représente la i ème entrée du neurone formel,

y la sortie de neurone,

β seuil du neurone,

w_i sont des paramètres de pondération,

f est une fonction de seuillage,

S la somme pondérée des entrées x_i .

III.4.2 Modélisation générale

D'une façon plus générale, on peut définir un neurone formel par les cinq éléments suivants :

- La nature de ses entrées.
- La fonction d'entrée totale qui définit le prétraitement effectué sur les entrées.
- La fonction d'activation (ou d'état) du neurone qui définit son état interne en fonction de son entrée totale.
- La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation.
- La nature de la sortie du neurone.

III.4.2.1 Nature des entrées et de sortie

Elles peuvent être :

- Binaires : (0, 1).
- Réelles. (-1, +1).

III.4.2.2 Fonction d'entrée totale

h Peut être :

- Booléenne.
- Linéaire : $h(x_1, \dots, x_n) = \sum w_i x_i$ (III.3)
- Affine : $h(x_1, \dots, x_n) = \sum w_i x_i - \theta$ (III.4)

Ces deux derniers cas sont les plus fréquents. Dans le deuxième on peut interpréter $(-\theta)$ comme l'utilisation d'un neurone qui fournirait toujours l'entrée -1 au neurone étudié. Cette notation sera

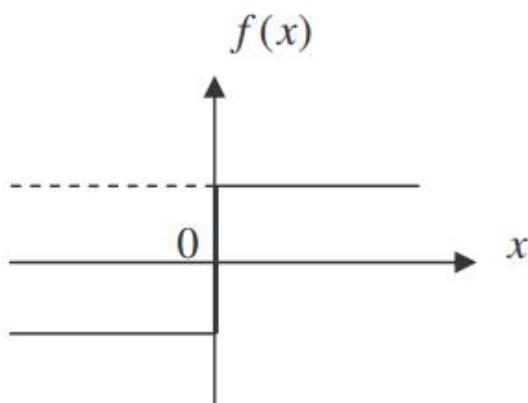
particulièrement utile lorsque nous aborderons les problèmes d'apprentissage avec des Fonctions d'activations à seuil.

III.4.2.3 Fonction d'activation

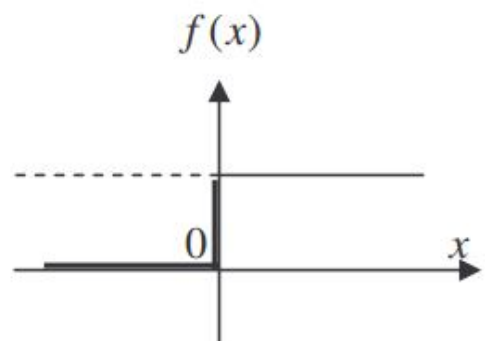
C'est une fonction présentée généralement par une non linéarité appelée aussi fonction de seuil. Elle permet de définir l'état interne du neurone en fonction de son entrée totale et rendre l'activité du neurone bornée.

Les fonctions les plus souvent utilisées sont représentées par la Figure (III.4) [62]. Afin d'améliorer les performances du modèle de W.Mc Culloch et W.Pitts [63], les chercheurs ont adopté d'autre fonctions d'activation que la fonction seuil (0,1) :

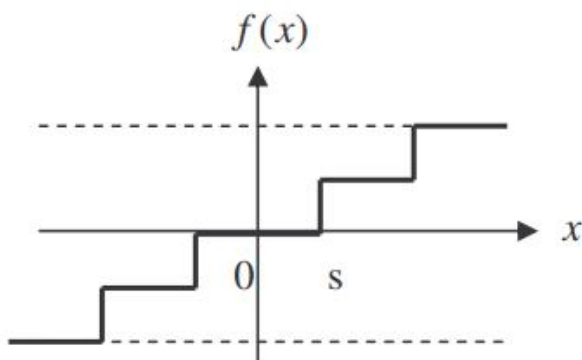
- La fonction à seuil binaire (fonction de Heaviside ou signe).
- La fonction linéaire à seuil (fonction de saturation).
- La fonction sigmoïde : $f(x) = \frac{1}{1 + e^{-kx}}$, est la plus utilisée.
- La fonction probabiliste (gaussienne).



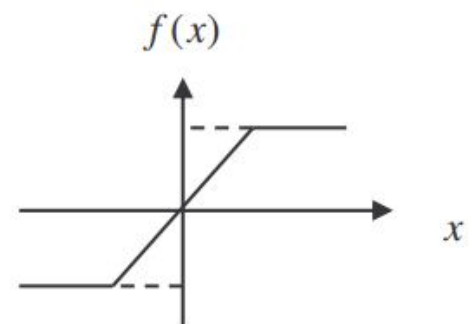
a. Fonction signe



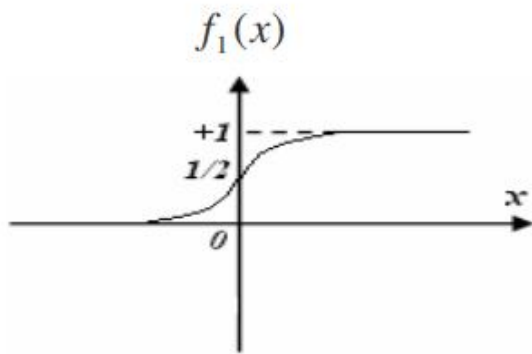
b. Fonction Heaviside



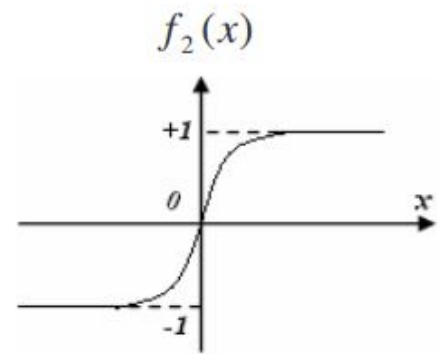
d. Fonction linéaire multi seuil



c. Fonction à seuil



e. Fonction sigmoïde (0, 1)



f. Fonction sigmoïde (-1, +1)

Figure (III.4) : Les fonctions d'activations

La fonction de la sortie du neurone détermine les réponses en fonction de l'état d'activation. Elle est souvent représentée par la fonction identité. On a donc :

$$y = f(x) \quad (\text{III.5})$$

La nature de la sortie :

La combinaison des trois fonctions (d'entrée, d'activation et de sortie) définit le type de neurone

Figure (III.5).

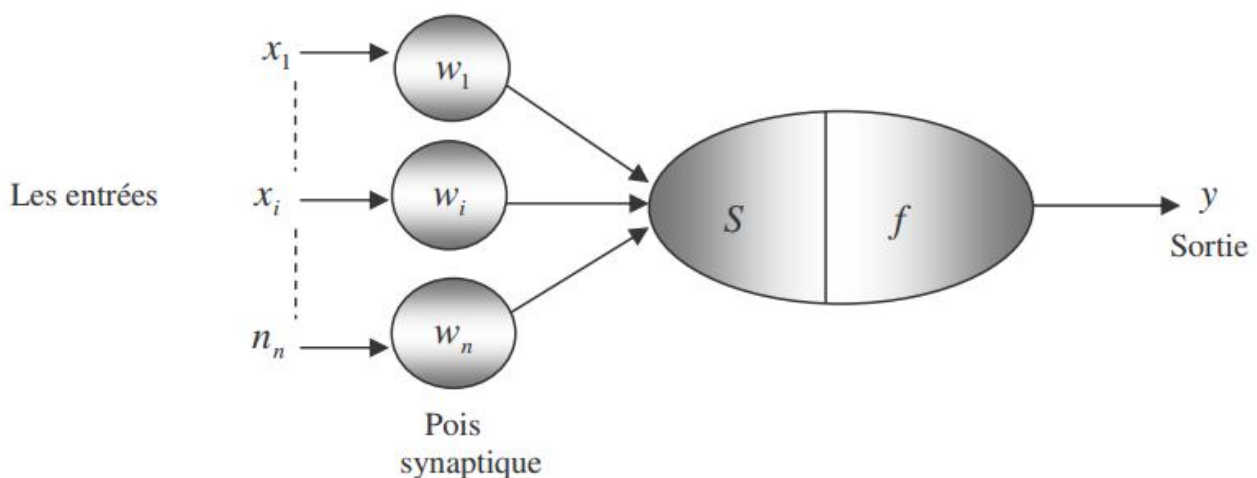


Figure (III.5) : Modélisation d'un neurone formel

Chaque neurone est relié à différentes entrées (x_n). Celles-ci sont soit les variables d'entrées du réseau, soit les sorties des couches précédentes. Chacune de ces entrées est pondérée par un poids (w_i). Un poids total S est ainsi calculé après rajout du biais (β) propre à chaque neurone.

Ce poids normalisé entre -1 et 1 indique le degré d'activation du neurone. Une fonction sigmoïde

(f) est alors utilisée pour la propagation de l'information et donne une sortie (y) toujours normalisée entre -1 et 1. Toutes les informations sont alors stockées dans la valeur des poids et un algorithme d'apprentissage par rétropropagation, sur de nombreux exemples est nécessaire à l'optimisation d'une solution.

En 1949, une fomentation de mécanisme d'apprentissage est introduite par Donald Hebb, sous la forme d'une règle de modification des connexions synaptiques (règle de Hebb) [64]. Elle est concédée comme le premier mécanisme d'évolution proposé pour les synapses. Cette règle est basée sur des notions biologiques, modélise le fait que si des neurones de part au d'autre d'une synapse, sont activés de façon synchrone est répétée, la force de la connexion synaptique va aller croissant. En d'autre terme, si la connexion entre deux cellules est très forte, quand la cellule émettrice s'active, la cellule réceptrice s'active aussi. Il faut donc, pour lui permettre de jouer ce rôle déterminant lors du mécanisme d'apprentissage, augmenter le poids de cette connexion Figure (III.6). La traduction mathématique de ce principe peut s'écrire de la forme suivante :

$$w_{ij}(t + \Delta t) = w_{ij}(t) + \mu A_i A_j \quad (\text{III.6})$$

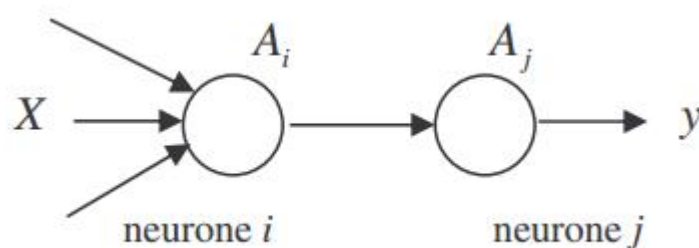


Figure (III.6) : Règle de Hebb pour deux neurones connectés

Où ($\mu > 0$) est un paramètre de l'intensité de l'apprentissage et A_i , A_j sont l'activation des neurones i et j respectivement.

III.4.2.4 La fonction de sortie

En général, cette fonction $g(E)$ est considérée comme étant la fonction identité $g(E) = f(E)$. Par la suite, nous confondrons activation et sortie du neurone.

Les fonctions d'entrée et d'activation sont choisies en fonction des types d'automates utilisés, dépendant de la nature des entrées et des sorties.

En général, ils sont au nombre de six :

- les automates booléens

Les entrées et les sorties sont booléennes.

La sortie est une fonction booléenne des entrées.

- les automates à seuil

- la sortie est binaire alors, que les entrées peuvent être binaires ou réelles ;
- La fonction d'entrée totale est affine.
- La fonction d'activation est la fonction signe ou la fonction de Heaviside.

- les automates linéaires

- Les entrées et les sorties sont réelles.
- La fonction d'entrée totale est linéaire.
- La fonction de sortie est l'identité.

- les automates à saturation

- Les entrées et les sorties sont comprises dans un intervalle $[u,v]$.
- La fonction d'entrée totale est linéaire.
- La fonction d'activation est la fonction SATUR.

Si les entrées et les sorties sont entières, on obtient un automate multiseuil.

- les automates continus

- Les entrées et les sorties sont réelles.
- La fonction d'entrée est linéaire ou affine.
- La fonction d'activation est une fonction sigmoïde.

- les automates probabilistes

- Les sorties sont binaires et les entrées sont quelconques.
- La fonction d'entrée est affine ou linéaire.
- La fonction d'activation est stochastique.

III.5 Les structures neuronale

Un problème qu'on doit impérativement résoudre avant d'utiliser un réseau de neurones est la définition de sa structure. Pour une topologie multicouche MLP, le nombre de neurones d'entrée/sortie du réseau est imposé par la structure de fonctionnement globale où il sera inséré,

tandis que le nombre de couches cachées ainsi que le nombre de neurones correspondant à chaque couche, ne sont pas limités. L'objectif final étant une réalisation matérielle embarquable, de ce fait, pour diminuer le temps de calcul, il est nécessaire de développer une architecture neuronale aussi petite que possible [65].

Les réseaux de neurones peuvent être classés suivant leur architecture, donnant naissances à deux types, les réseaux bouclés et les réseaux non bouclés.

III.5.1 Les réseaux bouclés :

Dans les réseaux bouclés ou entièrement connectés, chaque cellule est reliée à toutes les autres avec un retour sur elle-même, ils sont particulièrement adaptés pour construire des réseaux de type de Hopfield et de Boltzman avec des procédures d'apprentissage sans professeur, (non supervisé). Pour les réseaux bouclés, le temps intervient et le comportement des cellules du réseau est exprimé, en général par, des équations différentielles non linéaires. Pour des conditions initiales données correspondant à une situation perfectible, le réseau évolue au cours du temps pour atteindre un état stable ou instable. [66]

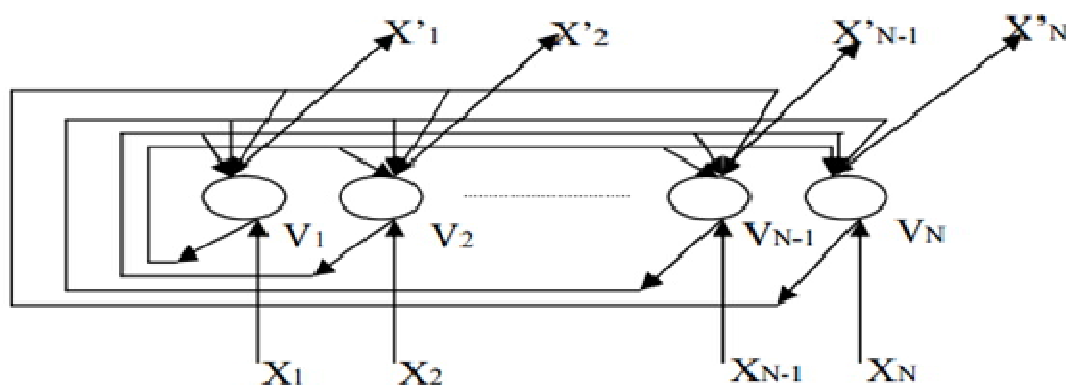


Figure (III.7) : réseau de neurones entièrement connectés (réseau bouclé)

Le réseau original de Hopfield, qui a subi par la suite de nombreuses améliorations est constitué de processus élémentaires effectuant une somme pondérée de toutes les entrées et d'une fonction d'activation à seuil pour fournir une réponse égale à 0 ou à 1. Chaque cellule élémentaire est connectée à toutes les autres de façon bidirectionnelle,

En 1985, Ackely, Hinton, et Sejnowski ont développé un algorithme d'apprentissage pour le réseau de Hopfield qui permet la dynamique des états de ce réseau, afin d'éviter les minima locaux, en choisissant la position des minima, ou les formes à mémoriser. Ce type de réseau se retrouve sur la machine de Boltzman, dont les neurones ont des caractéristiques probabilistes .

III.5.2 Les réseaux non bouclés :

Le réseau non bouclé est constitué de couches, telles que les deux couches extrêmes correspondent, d'une part, à la couche qui reçoit ses entrées du milieu extérieur et, d'autre part à la couche qui fournit le résultat du traitement effectué (couche de sortie), les couches intermédiaires sont appelées les couches cachées, leur nombre est variable. Les neurones qui appartiennent à une même couche ne sont pas connectés entre eux, chacune des couches reçoit des signaux de la couche précédente et transmet le résultat de ses traitements à la couche suivante. [66]

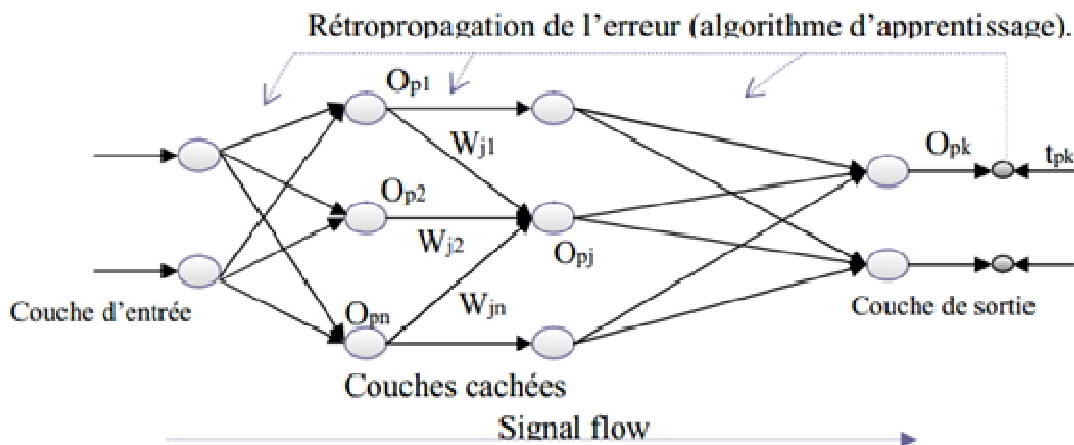


Figure (III.8) : Réseau de neurones multicouches (non bouclés)

III.6 Propriété fondamentale des réseaux de neurones : L'approximation universelle

La propriété d'approximation universelle, justifie dans une large mesure, l'utilisation des réseaux de neurones dans les domaines des sciences de l'ingénieur. La partie précédente montre que tout réseaux peuvent être présenté sous une forme canonique comprenant un réseau de neurones statiques. Les propriétés d'un réseau bouclé ou non bouclé, dépendent donc des propriétés de neurones non bouclés.

La propriété d'approximation universelle des réseaux de neurones peut s'énoncer comme suit :

Pour toute fonction déterministe suffisamment régulière, il existe au moins un réseau de neurone non bouclé, possédant une couche de neurones cachés et un neurone de sortie linéaire, qui réalise une approximation de cette fonction et ces dérivées successive, au sens de moindres carrées, avec une précision arbitraire.

Ce théorème alors ne s'applique que si les fonctions d'activités des neurones cachés sont continuées et bornées, telles que les fonctions sigmoïdes et les fonctions à base radiale.

Nous signalons, que cette propriété n'est pas spécifique aux réseaux de neurones. Les polynômes, les séries de Fourier possèdent aussi cette particularité. La différence avec les réseaux de neurones réside dans leur parcimonie. En effet, pour obtenir une approximation d'une précision donnée, les réseaux de neurones utilisent moins de paramètres que les approximateurs usuels. En particulier, le nombre de paramètres varie essentiellement de manière linéaire en fonction du nombre de variable de la fonction que l'on cherche à approcher, alors qu'il varie beaucoup plus rapidement avec la dimension de l'espace des entrées dans le cas des approximateurs usuels.

En pratique, les RNA sont donc avantageux, par rapport à d'autres approximateurs conventionnels, surtout pour des systèmes à plus d'une variable. Les réseaux de neurones offrent donc les avantages suivants :

- l'estimation des fonctions est d'autant plus rapide que le nombre de paramètres à calculer est petit.
- avec un minimum d'exemple à l'entrée du réseau, il est possible d'approximer une fonction délicate, et difficile à traiter analytiquement. Le fait d'utiliser moins de coefficients que les méthodes classiques de régression, permet donc une économie en nombre d'exemples d'entrées, ce qui peut être particulièrement important lorsque l'acquisition des exemples est coûteuse ou lente.

Il en résulte de tout ça, que :

- les RNA sont des outils statiques de régression qui permettent l'approximation, au sens des moindres carrés, de toute fonction non linéaire suffisamment régulière.
- cette approximation est caractérisée par une économie, en nombre de coefficients.

III.7 L'apprentissage des réseaux de neurones

On peut considérer les réseaux de neurones comme une boîte noire contenant l'information qu'elle doit apprendre et mémoriser. Mais au démarrage lorsqu'on choisit notre réseau, la boîte noire est vide et ne contient aucune information, ni aucune connaissance sur son sujet, c'est pourquoi un apprentissage est nécessaire. L'enseignement que doit subir le réseau de neurones est un apprentissage qui est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

L'apprentissage des réseaux de neurones consiste à adapter ses différents paramètres (poids) d'après un algorithme itératif d'ajustement ou d'adaptation lui permettant de prendre en considération toutes les données (exemples) qui lui sont fournies à son entrée et ainsi ajuster ses paramètres pour trouver le juste milieu permettant de prendre en charge n'importe quel exemple ou donnée apparaissant à son entrée provenant de son environnement [61] [67].

Les algorithmes d'apprentissages donnent des meilleurs résultats lorsqu'on leur fournit des exemples multiples et variés ; ainsi le réseau peut assimiler toutes les connaissances. Il existe différentes règles d'apprentissage parmi lesquelles on peut distinguer [61] :

- la règle de Widrow-Hoff, la règle de Hebb, la règle du perceptron et la règle de Grossberg, etc...

III.7.1 Apprentissage supervisé

Les réseaux multicouches avaient déjà été définis par Rosenblatt, mais on ne savait pas comment faire l'apprentissage. Avec la découverte de l'algorithme de rétropropagation de l'erreur (RP) par Rumelhart, on a commencé à faire de l'apprentissage des réseaux de neurones multicouches à partir d'exemples. Cette méthode de détermination des poids est appelée apprentissage supervisé [68].

L'apprentissage supervisé, repose sur le fait que les exemples sont des couples (entrée, sortie désirer). C'est à dire que l'on suppose l'existence d'un expert qui prend en charge la sortie de notre réseau en lui fournissant une sortie désirée et les associe aux sorties réelles fournies par le réseau d'après les données à l'entrée. Le réseau adapte ses paramètres en fonction de la différence qui existe entre la sortie réelle et la sortie désirée en prenant compte de tous les exemples de l'environnement [61] [67].

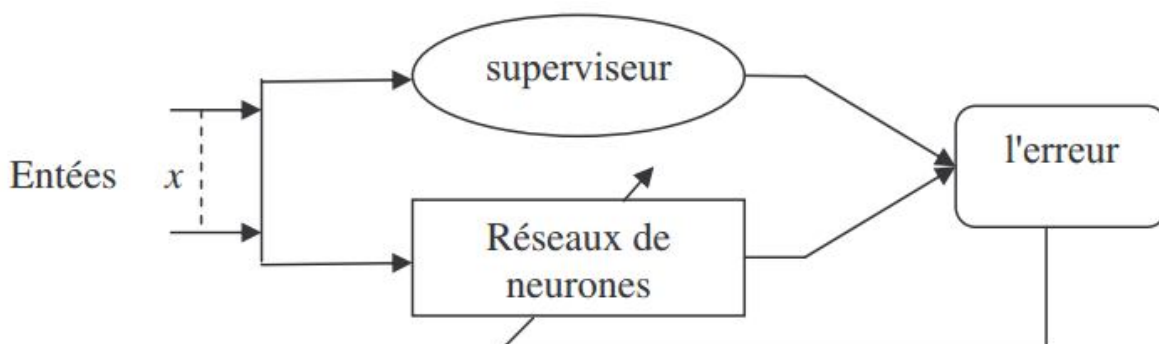


Figure (III.8) : Illustration de l'apprentissage supervisé

III.7.2 Apprentissage non supervisé

La différence majeure entre l'apprentissage supervisé et non supervisé peut être résumée dans le fait que le deuxième type d'apprentissage est autodidacte qui n'a pas besoin d'expert pour le guider à adapter ses paramètres qu'il ne dispose que des valeurs entrées. Remarquons cependant que les modèles d'apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de

labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision [61] [67].

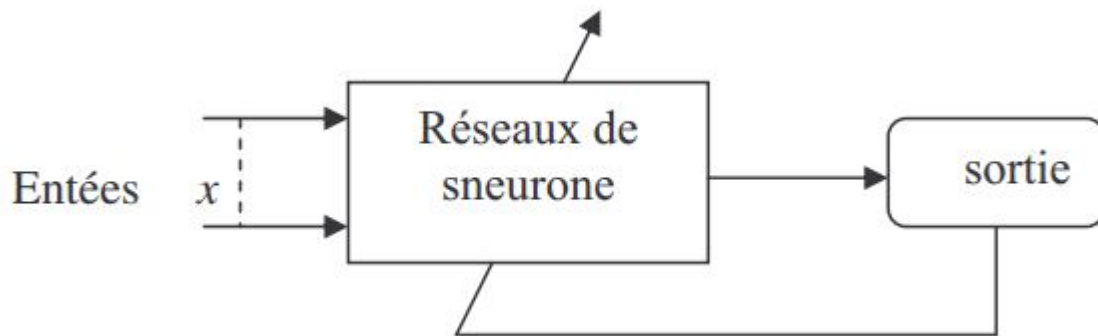


Figure (III.9) : Illustration de l'apprentissage non supervisé

III.7.3 Apprentissage par renforcement

L'idée de base de l'apprentissage par renforcement est inspirée des mécanismes d'apprentissage chez les animaux, dans ce type l'apprentissage on suppose qu'il n'existe pas de maître qui peut fournir la réponse correcte, mais le système à entraîner est informé, d'une manière indirecte, sur l'effet de son action choisie. Cette action est renforcée si elle conduit à une amélioration des performances du système entraîné et les éléments qui contribuent dans la génération de cette action sont soit récompensés ou punis [69].

III.8 Principales règles d'apprentissage

Les stratégies de modification des poids synaptiques sont dérivées des règles générales suivantes :

- La règle de Hebb ou le poids des connexions entre deux processeurs élémentaires est renforcée si les deux processeurs élémentaires sont activés simultanément.
- La règle Widrow-Hoff ou règle Delta règle où le poids synaptique est adapté pour obtenir la diminution de l'erreur entre la sortie réelle du processeur élémentaire et la sortie désirée.
- L'apprentissage compétitif dans lequel tous les processeurs élémentaires sont en compétition basée sur une mesure de similarité entre l'exemple de référence associé à chaque processeur et l'exemple d'entrée.

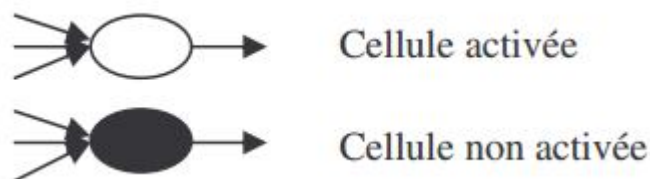
- Les expressions précises des règles dépendent très étroitement de la nature des fonctions d'entrée totale, des fonctions d'activation et de la nature continue ou discrète des informations continues dans les cellules des neurones [63].

III.8.1 Règle de Hebb

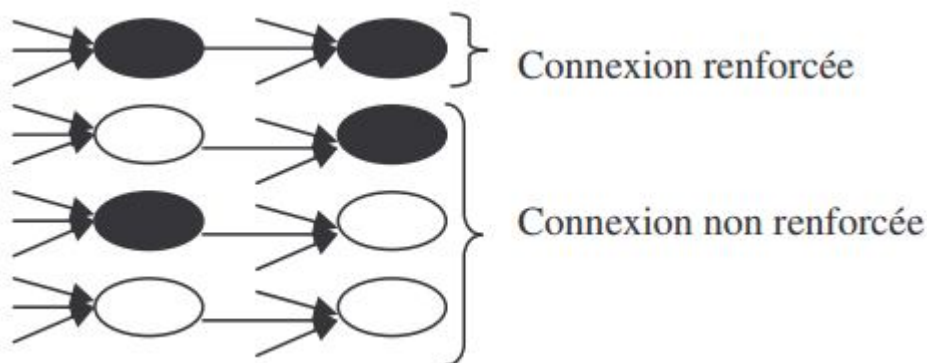
La règle de Hebb est le premier mécanisme d'évolution proposé sur les synapses. Son interprétation pour les réseaux de neurone formels est la suivante.

On considère que si deux neurones connectés entre eux sont activés aux mêmes moments, la connexion qui les relie doit être renforcée et elle n'est pas modifiée, dans le cas contraire. C'est-à-dire que le poids w_{ij} d'une connexion entre un neurone i et un neurone j augmente quand les deux neurones sont activés en même temps et il n'est pas modifié, dans le cas contraire.

Si nous prenons, à titre d'exemple, les connexions suivantes



La règle de Hebb donne alors :



Ceci se traduit par :

Lorsque la connexion entre deux cellules est très forte, quand la cellule émettrice s'active, la cellule réceptrice s'active aussi, il faut augmenter le poids de cette connexion lors de l'apprentissage.

Si la cellule émettrice s'active sans que la cellule réceptrice ne le soit, ou si la cellule réceptrice s'active alors que la cellule émettrice ne s'était pas activée, cela traduit bien le fait que la connexion

entre les deux n'est pas prépondérante dans le comportement de la cellule réceptrice on peut donc dans la phase d'apprentissage laisser un poids faible à cette connexion [68].

En cas se basant sur ce principe, Hebb a donné la règle d'apprentissage suivante :

$$w_{ij}(t + \partial t) = w_{ij}(t) + \mu A_i A_j \quad (\text{III.7})$$

Avec :

$w_{ij}(t)$ et $w_{ij}(t + \partial t)$: les poids de la connexion entre le neurone i et le neurone j aux instants "t" et "t + ∂t ".

A_i et A_j : l'activation du neurone i et l'activation de neurone j ($\mu > 0$), sont des paramètres de l'intensité de l'apprentissage.

III.8.2 Règle de Widrow Hoff.

La règle de Widrow Hoff ou règle delta proposée en 1960, consiste à modifier, à chaque pas, les poids et les biais afin de minimiser la somme des carrés des erreurs en sortie en utilisant la règle suivante [70] :

$$w(k+1) = w(k) + \eta(t_k - y_k) x_k^T \quad (\text{III.8})$$

A chaque pas d'apprentissage k, l'erreur en sortie est calculée comme la différence entre la cible recherchée t et la sortie y du réseau.

$$E_k = e_k^T e_k = (t_k - y_k)_k (t_k - y_k)_k = \frac{1}{2} (t_k^T t_k - y_k^T y_k - 2y_k^T t_k) \quad (\text{III.9})$$

Le gradient de cette quantité par rapport à la matrice de poids w est donné par :

$$\frac{\nabla E_k}{w} = \frac{1}{2} \cdot \frac{\nabla (y_k y_k^T - 2y_k^T t_k)}{w} \quad (\text{III.10})$$

Le gradient se calcul comme suit :

$$\frac{\nabla E_k}{w} = \frac{\partial E_k}{\partial w} = \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w} \quad (\text{III.11})$$

D'après l'expression de E_c et avec $y_k = wx_k + b$ les dérivées partielles sont :

$$\frac{\partial E_k}{\partial y_k} = y_k - t_k \quad (\text{III.12})$$

$$\frac{\partial (wx_k + b)}{\partial w} = x_k^T \quad (\text{III.13})$$

La mise à jour des poids se fait par l'équation :

$$w(k+1) = w(k) - \eta \cdot \frac{\nabla E_k}{w} \quad (\text{III.14})$$

Avec η : le gain d'apprentissage ($0 < \eta < 1$).

De même, on obtient l'expression de la modification du biais

$$b(k+1) = b(k) + \eta(t_k - y_k) = b(k) - \eta \cdot \frac{\nabla E_k}{w} \quad (\text{III.15})$$

III.9 Modèles de perceptron du réseau de neurones

Le mécanisme perceptron fut inventé par le psychologue Frank Rosenblat à la fin des années 50. Il représentait sa tentative d'illustrer certaines propriétés fondamentales des systèmes intelligents en général. Le réseaux dans ce modèle est formé de trois couches : Une couche d'entrée (la rétine), fournissant des données à une couche d'association (intermédiaire), chargée des calculs, cela en fournissant la somme des impulsions qui lui viennent des cellules auxquelles elle est connectée, et elle répond généralement suivant une loi définie avec un seuil, elle fournit une réponse suivant la loi du tout-ou-rien (1 ou 0), elle-même connectée à la couche de sortie (couche de décision), qui interprète le résultat finale et représentant les exemples à mémoriser. Seule cette dernière couche renvoie des signaux à la couche intermédiaire, jusqu'à ce que leurs connexions se stabilisent. Figure (III.10).

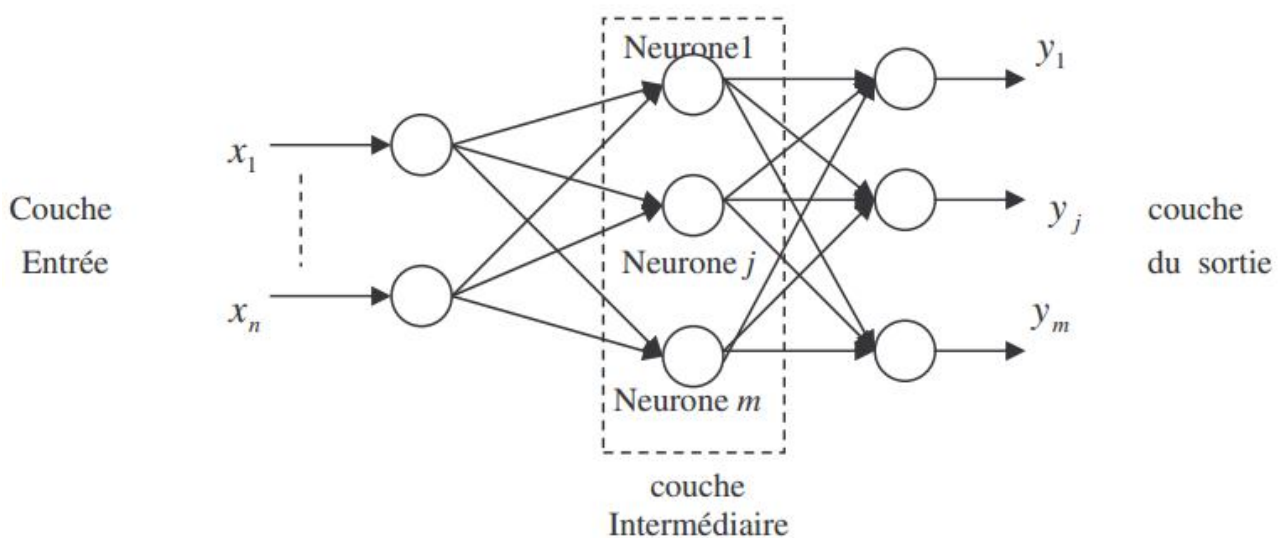


Figure (III.10) : Le modèle de perceptron

La fonction d'activation de ce type de réseaux est définie par l'équation linéaire suivante :

$$S_j = \sum_{i=1}^n w_{ij} x_i + \theta_i \quad ; j=1, \dots, m \quad (\text{III.16})$$

La sortie est définie par la fonction seuil :

$$y_j = f(S_j) = \begin{cases} 1 & S_j > 0 \\ 0 & S_j \leq 0 \end{cases} \quad (\text{III.17})$$

L'apprentissage de perceptron est de type supervisé. Cela consiste à présenter une série d'exemples des entrées x et des sorties désirées y . L'adaptation des poids de ce réseau n'est autre qu'un cas particulier de l'application de la méthode de la descente du gradient présentée ci-dessous

$$w(t+1) = w(t) + \Delta w(t) \quad (\text{III.18})$$

Où :

$$\Delta w(t) = \eta (y_d(t) - y(t)) x(t) \quad (\text{III.19})$$

Avec :

S_j : représente sortie d'un neurone,

w_{ij} : les poids synaptiques en provenance du $i^{\text{ème}}$ neurone de la couche d'entrée vers le $j^{\text{ème}}$ neurone de sortie.

θ_i : paramètre qui agit sur le seuil en sortie.

n : le nombre d'élément du vecteur d'entrée.

m : nombre de sorties.

η : paramètre qui contrôle le taux d'apprentissage.

$y_d(t)$: la sortie désirée.

$w(t)$: le vecteur de poids.

Le réseau de ce perceptron est capable de résoudre des opérations logiques simples comme AND ou OR. En 1969, Papert et Minsky, démontrent les limites de ce perceptron. Ce dernier n'est efficace que pour les problèmes linéairement séparables. Ceci montre l'intérêt à généraliser l'algorithme du perceptron à un réseau à plus d'une couche.

III.9.1 Perceptron multicouches

Comme nous l'avons déjà dit le cerveau humain est composé des milliers et des milliers de neurones, alors il est évident qu'un simple neurone et seul ne peut rien faire à lui tout seul il lui faut la coopération d'autres neurones. En suivant ce raisonnement il est évident qu'il vaut trouver une architecture qui relie les neurones entre eux, qui crée une liaison entre les neurones pour créer un réseau de neurones.

En s'inspirant du perceptron monocouche, une architecture plus complexe englobant plusieurs neurones a été mise au point.

Cette nouvelle architecture est le perceptron multicouches (ou MLP pour Multi Layer Perceptron en anglais). L'apparition de cette architecture a permis de résoudre les problèmes de classification non linéaire du perceptron et de dépasser les limites principales de celui-ci. L'idée principale est de grouper des neurones dans une couche. En plaçant ensuite bout à bout plusieurs couches et en connectant complètement les neurones de deux couches adjacentes. Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche [61] [71].

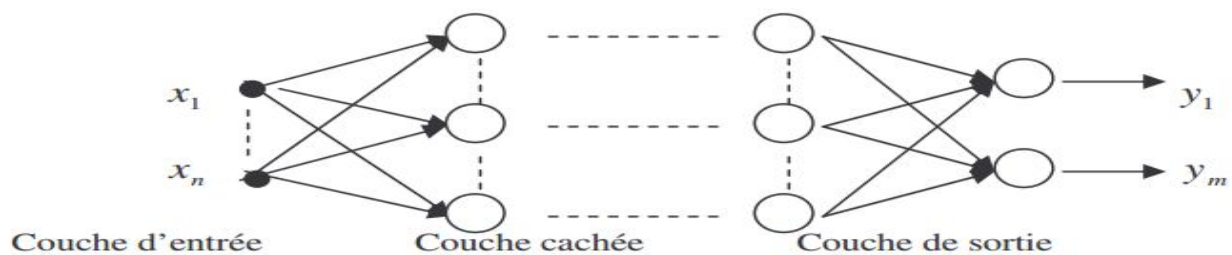


Figure III.11 : Structure du perceptron multicouches

Le perceptron multicouche est très utilisé en identification et en contrôle. Avec une couche cachée, il constitue un approximateur universel. De récentes recherches montrent qu'il peut être entraîné de manière à approximer n'importe quelle fonction entrées-sorties sous réserve de mettre suffisamment de neurones dans la couche cachée et d'utiliser des sigmoïdes comme fonctions d'activation [72].

L'un des problèmes de l'utilisation des réseaux de neurones consiste dans le choix de sa topologie. Par exemple, il n'existe pas une règle générale qui donne le nombre de neurones à retenir pour la couche intermédiaire. Ce choix est spécifique à chaque application. Pour cela, un nombre aléatoire est toujours introduit au début. Ensuite, l'exécution consécutive de l'algorithme d'apprentissage et avec l'observation de l'erreur aide à sélectionner le nombre qui donne la précision souhaitée.

Une fois que l'architecture est déterminée, on passe à l'étape de l'apprentissage. Au cours de cette phase est utilisé un algorithme pour assurer l'ajustement des poids synaptiques, afin de conférer au réseau le comportement entrée-sortie désiré.

Quant aux réseaux multicouches, leur apprentissage est assuré généralement par des algorithmes tel que (Rétropropagation du gradient, Gauss Newton, Levenberg Marquardt, ...).

III.10 Méthodes d'apprentissage du réseau de neurones

L'apprentissage consiste à adapter la connaissance du RNA au problème posé. La connaissance étant définie par l'ensemble des pondérations sur les liens du réseau ainsi que par sa topologie, l'apprentissage est donc étroitement lié au réglage adéquat des poids d'interconnexion, ainsi qu'à l'adaptation de la topologie.

Il existe divers algorithmes d'apprentissage pour ajuster les poids d'un réseau neuronal, et éventuellement sa topologie. Généralement l'architecture du réseau est définie par l'utilisateur, puis éventuellement optimisée par l'ordinateur. Tandis que l'ajustement des poids est totalement pris en charge par le système.

L'apprentissage peut s'effectuer de diverses manières. Il peut être supervisé, non supervisé ou par renforcement, Comme nous l'avons dit précédemment. Cette partie emploie le réseau (perceptron) multicouche, il existe plusieurs types d'algorithme d'apprentissage.

III.10.1 Méthode d'apprentissage retenue et développée

Parmi les algorithmes de la famille quasi-Newton, la méthode de Levenberg- Marquardt est un standard pour l'optimisation de l'erreur quadratique due à ses propriétés de convergence rapide et de robustesse. Elle s'appuie sur les techniques des moindres carrés non-linéaires et de l'algorithme de Gauss-Newton à voisinage restreint [73].

Le problème résolu par cet algorithme est le suivant : soit f une fonction, dépendant d'un vecteur de paramètres p . On ne connaît pas l'expression de f , mais on sait calculer numériquement quelques caractéristiques de cette fonction (valeur ou dérivée en un point par exemple). On cherche, numériquement également, le vecteur p qui minimise la quantité $f(p)$.

Dans le contexte des réseaux de neurones, f est la fonction de performance, généralement l'erreur quadratique moyenne calculée sur une base d'apprentissage, et p est un vecteur regroupant

tous les poids du réseau. Un algorithme d'optimisation permet d'effectuer un apprentissage du réseau, c'est à dire de trouver les poids qui minimisent l'erreur quadratique moyenne

L'algorithme ci-après ne trouvera qu'un minimum local de la fonction. Si f possède plusieurs minima locaux, celui vers lequel convergera l'algorithme dépend du point de départ choisi et des paramètres de l'algorithme. On ne peut garantir de converger vers un minimum global.

Une autre manière de diminuer le nombre d'itérations d'un algorithme d'optimisation est d'utiliser les dérivées secondes de f . En effet le gradient donne une direction vers laquelle se déplacer pour trouver le minimum, mais ne donne pas le pas. Dans la descente de gradient classique ce pas est un coefficient fixe, et dans la variante adaptative il peut varier à chaque itération. Mais la dérivée seconde de f est liée au rayon de courbure de la fonction, et permet donc de déterminer ce pas de manière plus fine.

En effet si l'on suppose que f est une fonction quadratique :

$$f(p) = a + b^T p + p^T C p \quad (\text{III.20})$$

L'équation (IV.20) représente le Modèle quadratique de la fonction f

Où b^T est la transposée du vecteur b et C est une matrice symétrique, on peut trouver l'extremum de la fonction, il s'agit du point auquel la dérivée de f s'annule.

$$\nabla f = 0 \Leftrightarrow b + 2C.p = 0 \quad (\text{III.21})$$

Soit :

$$p = -2C^{-1}.b \quad (\text{III.22})$$

Équation (III.21) : Recherche du minimum du modèle quadratique.

Équation (III.22) : Expression du minimum du modèle quadratique.

A condition que C soit inversible. Pour une fonction f quelconque, il est possible de l'approximer localement en un point p_i par une fonction quadratique, en utilisant ses dérivées première et seconde, et avec l'équation (III.23) déterminer le vecteur pour l'itération suivante d'un algorithme d'optimisation plus évolué que la descente de gradient. Mais le calcul des dérivées secondes peut être très long, tout d'abord parce que le nombre de dérivées secondes est le carré de celui des dérivées premières, et également parce que la dérivée seconde de f peut être assez complexe. De nombreux algorithmes, peut-être abusivement appelés algorithmes d'ordre 2, utilisent en fait une approximation des dérivées secondes calculées à partir de dérivées premières. Cependant ils gardent l'avantage de nécessiter beaucoup moins d'itérations qu'une descente de gradient.

L'algorithme de Levenberg Marquardt [74] fait partie de ces algorithmes, et s'applique au cas particulier où f est une erreur quadratique moyenne. On peut donc l'exprimer sous la forme :

$$f(p) = \langle (g(x, p) - y)^2 \rangle \quad (\text{III.23})$$

L'équation (III.23) est l'expression de f sous la forme d'une erreur quadratique moyenne.

Où g désigne une fonction de deux vecteurs x et p et $\langle \dots \rangle$ désigne la moyenne calculée sur un ensemble de couples (x, y) . L'on se place dans le cas où g est une fonction scalaire afin de simplifier la notation, mais la même démarche peut être faite si g est une fonction vectorielle.

Dans la suite de cette section toutes les dérivées sont en fonction du vecteur p . C'est en effet uniquement ce vecteur que l'on fait varier afin de trouver le minimum de f .

On suppose, comme pour la descente de gradient, que l'on se trouve à une itération numéro i , et que l'on cherche à calculer un nouveau vecteur p_i en fonction de p_{i-1} , tel que $f(p_i)$ se rapproche plus d'un minimum local de f . Pour cela on calcule une approximation quadratique $\hat{f}$ de f à partir d'une approximation linéaire $\hat{g}$ de g autour du point p_{i-1} . En déterminant le point p auquel le gradient de $\hat{f}$ s'annule, on obtient :

$$p = p_{i-1} - H^{-1}d \quad (\text{III.24})$$

Équation (III.24) : Expression du vecteur qui minimise l'approximation de f

Avec :

$$\begin{cases} d = \langle (g(x, p_{i-1}) - y) \nabla g(x, p_{i-1}) \rangle \\ H = \langle (\nabla g(x, p_{i-1}) - y) \nabla g(x, p_{i-1})^T \rangle \end{cases} \quad (\text{III.25})$$

À condition que H soit inversible. La matrice H est une approximation du Hessien de f , calculée à partir du gradient de g . L'équation précédente pourrait servir dans un algorithme d'optimisation, qui permet de calculer p_i à partir de p_{i-1} au cours de l'itération i . Mais ceci n'est efficace en pratique que si g est effectivement proche d'une droite autour du point p_{i-1} . Dans le cas contraire cet algorithme donne de très mauvais résultats.

L'idée de Levenberg est donc d'utiliser cette approche quadratique dans les zones où g est quasi-linéaire, et une descente de gradient dans les autres cas. Le pas d'une itération de cet algorithme est calculé de la manière suivante :

$$p = p_{i-1} - (H + \lambda I)^{-1} d \quad (\text{III.26})$$

L'équation (III.26) calcule le pas de Levenberg.

Lorsque λ est faible, cette équation est équivalente à (III.25), et le nouveau vecteur de paramètres est déterminé avec l'approximation quadratique de f . Lorsque λ est grand, cette équation est équivalente à :

$$\begin{aligned} p_i &= p_{i-1} - \frac{1}{\lambda} d \\ &= p_{i-1} - \frac{1}{\lambda} \langle (g(x, p_{i-1}) - y) \nabla g(x, p_{i-1}) \rangle \\ &= p_{i-1} - \frac{1}{2\lambda} \nabla f(x, p_{i-1}) \end{aligned} \quad (\text{III.27})$$

L'équation (III.27) donne l'équivalence entre l'algorithme de Levenberg et la descente de gradient pour λ grand.

Ce qui correspond bien à une descente de gradient. Pour des valeurs intermédiaires de λ l'algorithme est un mélange entre la descente de gradient et l'approche quadratique basée sur l'approximation linéaire de g . Ce coefficient λ est modifié à chaque itération, comme pour la descente de gradient adaptative. Si $f(p_i)$ diminue au cours de l'itération, on diminue λ (en le divisant par 10 par exemple), et l'on se rapproche ainsi de la méthode quadratique. Au contraire si $f(p_i)$ augmente, cela signifie que nous nous trouvons dans une région dans laquelle g n'est pas très linéaire, et donc on augmente λ (en le multipliant par 10 par exemple) afin de se rapprocher de la descente de gradient.

Cet algorithme a ensuite été amélioré par Marquardt, le pas de l'itération étant défini cette fois par :

$$p_i = p_{i-1} - (H + \lambda \text{diag}(H))^{-1} d \quad (\text{III.28})$$

L'équation (III.28) calcule le pas de Levenberg-Marquardt

La matrice identité a été remplacée par la diagonale de H . Le but est ici de modifier le comportement de l'algorithme dans les cas où λ est grand, c'est à dire lorsque l'on est proche d'une descente du gradient. Avec cette modification l'on se déplace plus vite dans les directions vers lesquelles le gradient est plus faible, afin d'éviter de passer de nombreuses itérations sur un plateau. Ceci est appelé l'algorithme de Levenberg Marquardt.

En pratique cet algorithme, en particulier dans le cas des réseaux de neurones, permet de converger avec beaucoup moins d'itérations. Mais chaque itération demande plus de calculs, en particulier pour l'inversion de la matrice H , et son utilisation se limite donc aux cas où le nombre de paramètres à optimiser n'est pas très élevé. En effet le nombre d'opérations nécessaires à l'inversion d'une matrice est proportionnel à N^3 , N étant la taille de la matrice, et ici également la taille du vecteur p .

III.10.2 Avantage de la règle LM

La méthode LM est un condensé de deux techniques exposées précédemment. En effet, cette méthode tend vers la méthode de Newton pour une valeur de μ_k petite mais est équivalente à la méthode du gradient simple pour un pas $\eta = \frac{1}{\mu_k}$ pour une valeur de μ_k grande. Le Hessien est toujours défini positif ce qui assure la convergence vers un minimum de la solution.

III.11 L'algorithme de rétropropagation

L'algorithme d'apprentissage de rétropropagation est un algorithme de gradient itératif désigné pour minimiser l'erreur quadratique moyenne entre la sortie actuelle du réseau en amont et la sortie désirée. On suppose que la fonction sigmoïde est utilisée de la manière suivante :

Etape 1 : initialisation des poids et des seuils :

Attribuer de faibles valeurs aux poids et aux seuils.

Etape 2 : présentation des entrées et des sorties :

Présenter un vecteur d'entrée en valeur continue et spécifier les sorties désirées. Si le réseau est utilisé comme un classifieur, alors toutes les sorties sont mises à zéro sauf celle correspondant à la classe d'entrée dont la sortie désirée est 1. L'entrée pourrait être nouvelle pour chaque essai ou des échantillons d'un apprentissage peuvent être présentés cycliquement jusqu'à la stabilisation des poids.

Etape 3 : calcul des sorties actuelles :

On utilise la fonction sigmoïde non linéaire, la forme du réseau est celle représentée dans la figure (III.8).

Etape 4 : Adaptation des poids :

On utilise l'algorithme récursif sur les unités de sortie, on effectue un retour sur la première couche cachée et on ajuste les poids comme suit :

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \quad (\text{III.29})$$

Où : $\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}$

Dans cette équation $w_{ji}(t)$ est le poids, η est le terme de gain appelé coefficient d'apprentissage, et un terme d'erreur pour le neurone j.

Si l'indice j correspond à une unité de sortie, alors δ_{pj} peut être calculée par :

$$\delta_{pj} = f'_j(s_{pj}) \sum_k \delta_{pk} w_{kj} \quad (\text{III.30})$$

Les biais sont adaptées d'une manière similaire, ce sont des poids de connexion sur les entrées auxiliaires des neurones qui ont des valeurs constantes. La convergence est quelquefois rapide quand un terme de **momentum** est ajouté et les variations des poids sont lissés.

$$b_{pj}(t+1) = b_{pj}(t) + \eta \sum \delta_{pj} o_{pi}$$

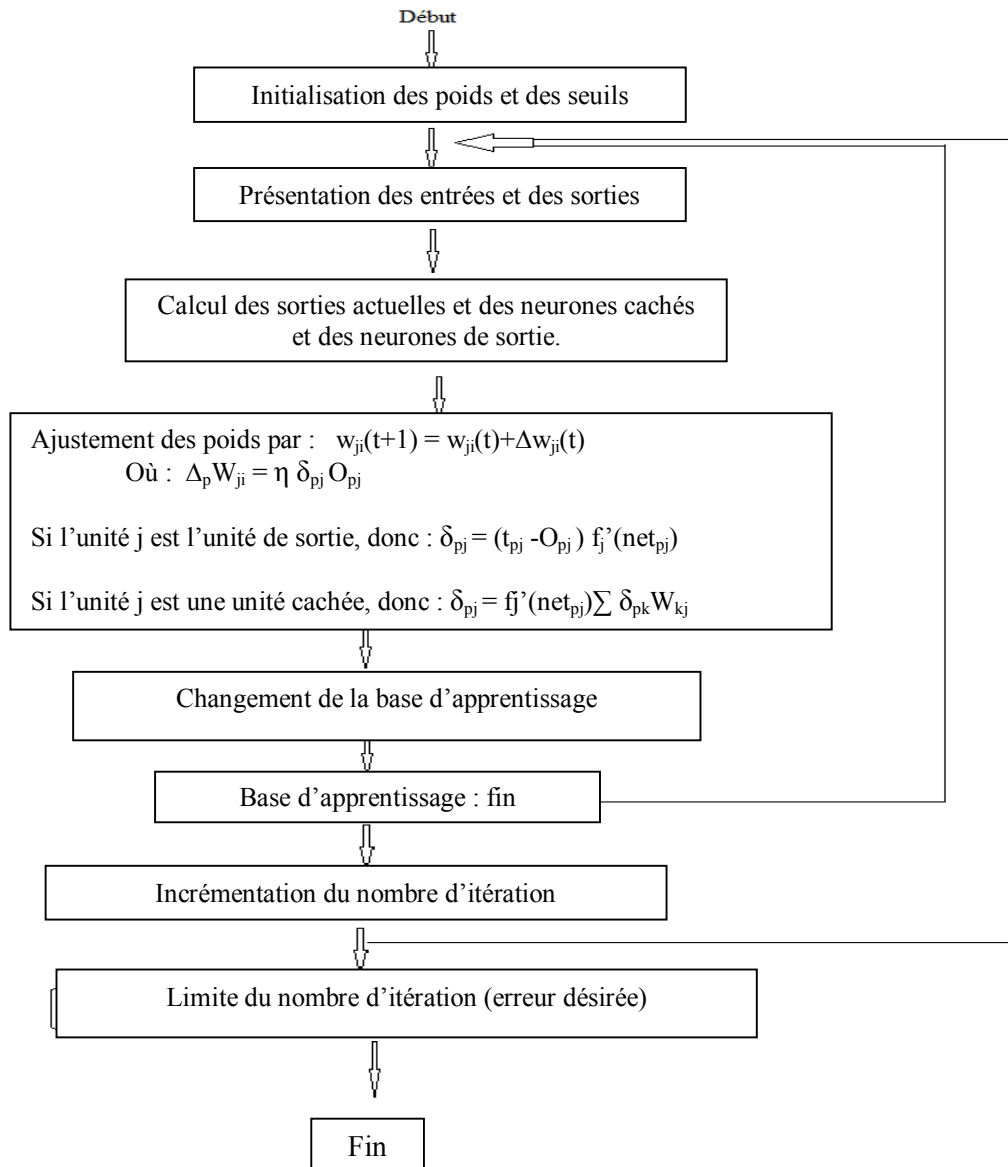


Figure III.12 : Algorithme d'apprentissage par rétropropagation

III.12 La commande MLI neuronal Introduction

L'objectif, ici, est de construire un réseau de neurones multicouches qui génère à ses sorties, les angles de commutation d'un signal MLI. Ces derniers doivent être proches des angles exacts calculés par la technique de Patel et Hoft. afin de commander un onduleur qui commandé la machine asynchrone.

III.12.1 Architecture du système

Pour simplifier l'architecture, on choisit un réseau de neurones composé à une couche d'entrée et une couche de sortie.

Pour la couche d'entrée, on prend Cinq neurones parce que le système possède trois entrées, qui représentent la tension d'alimentation. La fonction d'activation est la fonction sigmoïde, concernant la couche de sortie, elle est constituée de trois neurones, les trois sorties qui représentent l'état pour chaque interrupteur de l'onduleur. Comme monter la figure suivante

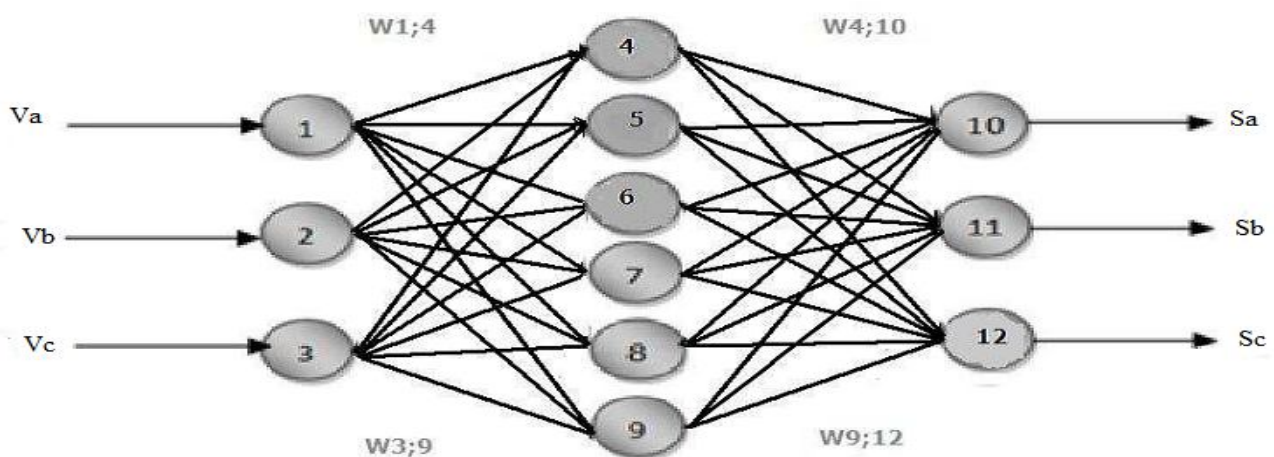


Figure II.13 : Structure interne d'un MLI neuronal.

III.12.2 Apprentissage :

Pour déterminer les paramètres du système neuronal (poids et seuils) de notre réseau, on écrit un programme dans le logiciel MATLAB, cette étape elle besoin une base des données ces sont deux matrices la première matrice est reprisent les défèrent valeur des tensions d'alimentation, et de la sortie est reprisent l'état des interrupteurs qui équivalent chaque tension d'alimentation. Par exemple :

La matrice d'entre :

$$\begin{bmatrix} 199.9754 & -97.2685 & -102.7069 \\ 0.1593 & 173.1254 & -173.2847 \\ . & . & . \\ -102.1727 & -97.8114 & 199.9841 \end{bmatrix}$$

et la matrice de sortie :

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ . & . & . \\ 1 & 0 & 1 \end{bmatrix}$$

III.12.3 Résultats d'essais de réseau

On a effectué un apprentissage automatique à l'aide du logiciel Matlab jusqu'à où on obtenu une erreur quadratique plus petite de « 1.6097-010 » (voir Figures III.14) on a obtenu la plus petite erreur après 120000 itérations.

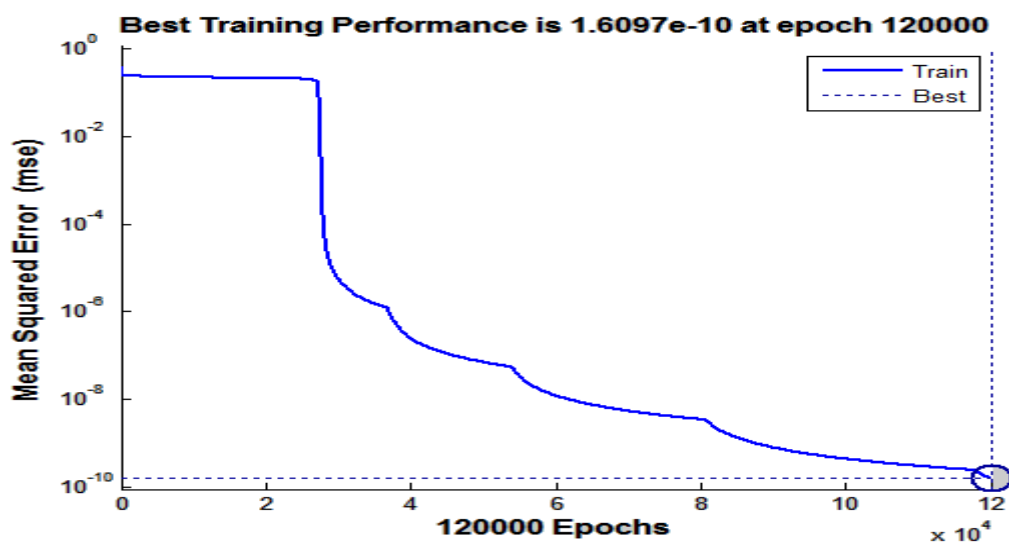


Figure III.14 : Evaluation de l'erreur quadratique en fonction du nombre d'itérations

III.12.4 Conception de RNA sous Simulink

La conception de RNA sous Simulink est représentée sur la figure (Figure III.15)

La figure suit monter la structure de neurone d'entre et le neurone de sortie :

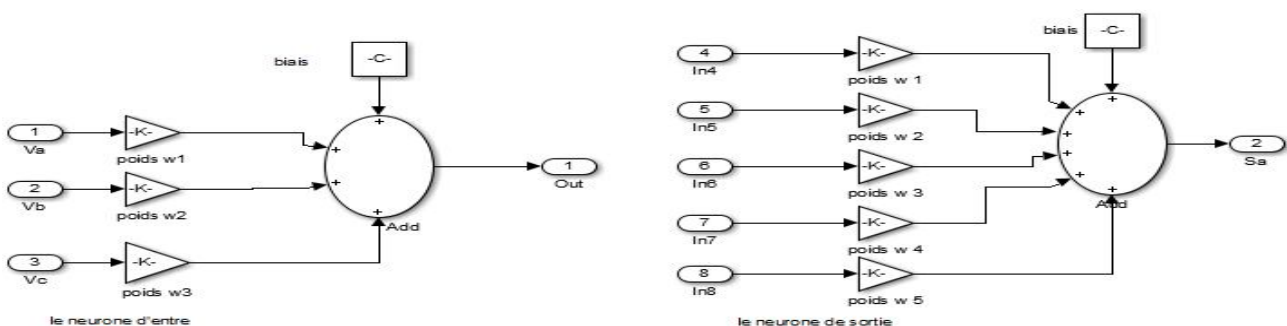


Figure III.15 : la structure de neurone d'entre et le neurone de sortie

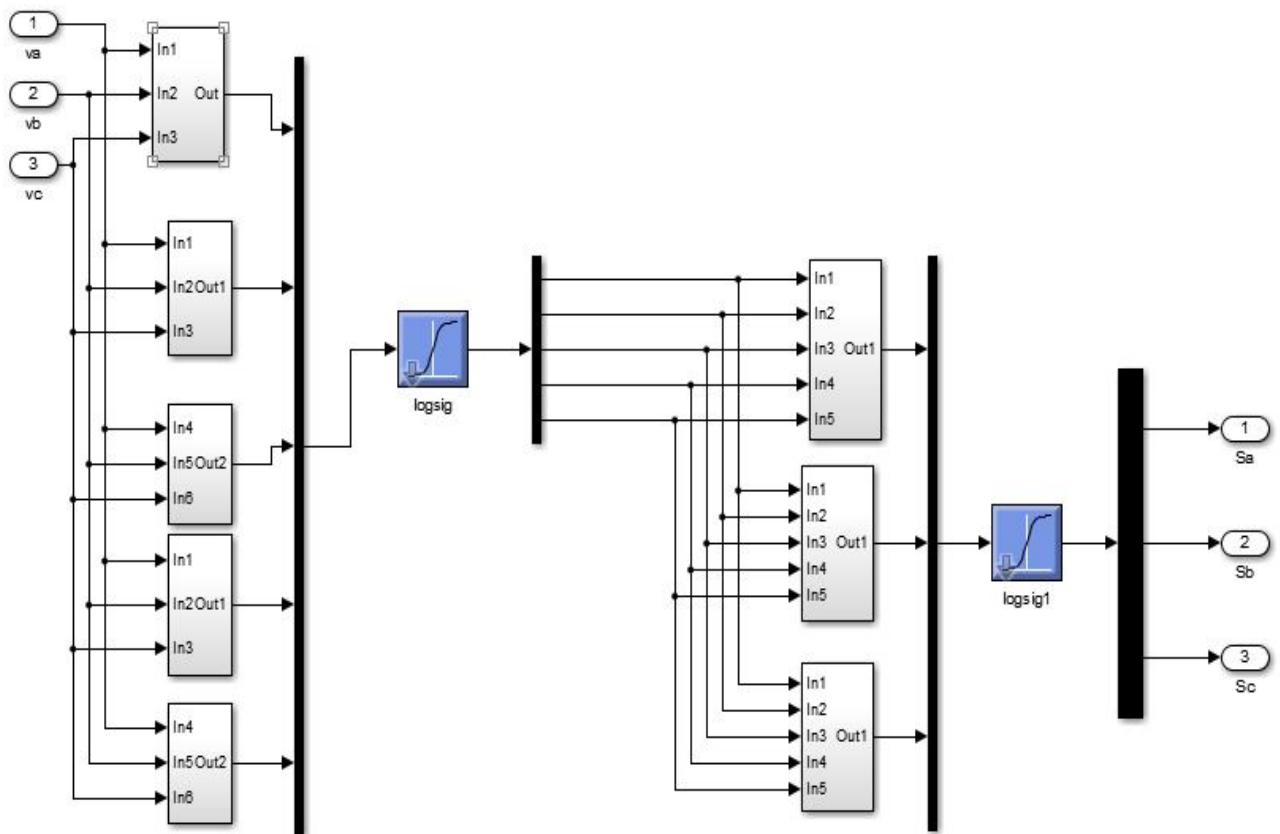


Figure III.16 : Architecture du MLI neural

III.12.3 Comparaissent entre MLI et MLI neural :

La figure (III.17) monter les courbes de vitesse qui générée avec la technique MLI la technique de RNA par le MLI neural :

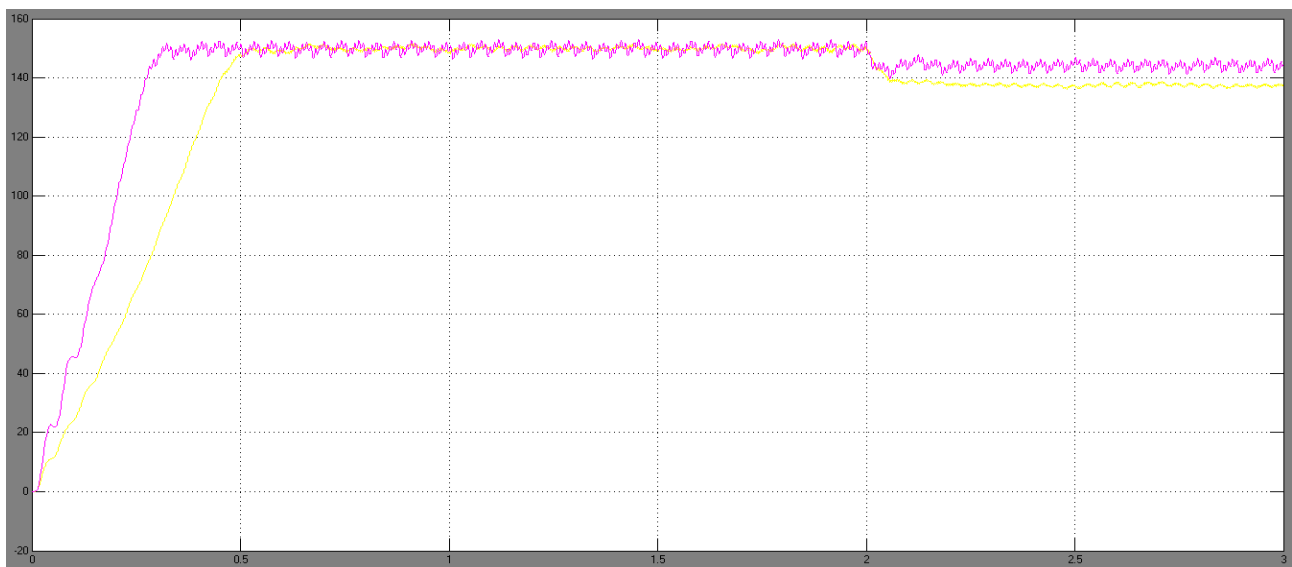


Figure III.17 : les courbes de vitesse

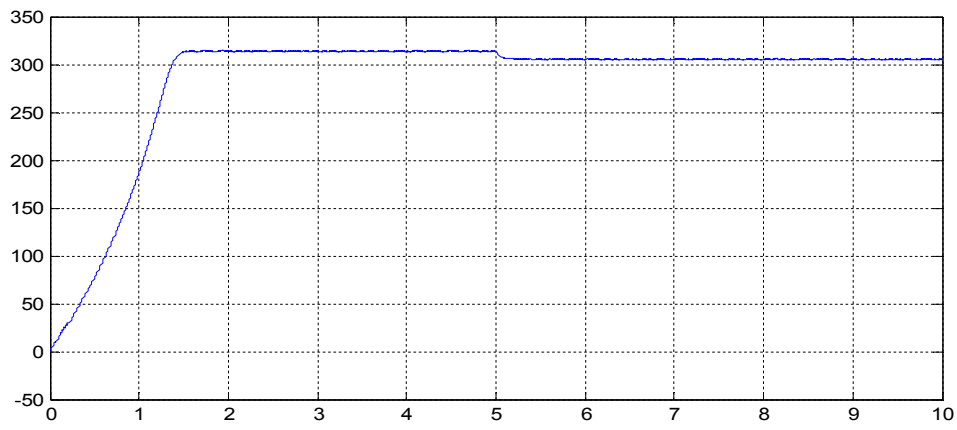


Figure III.18 : La courbe de la vitesse

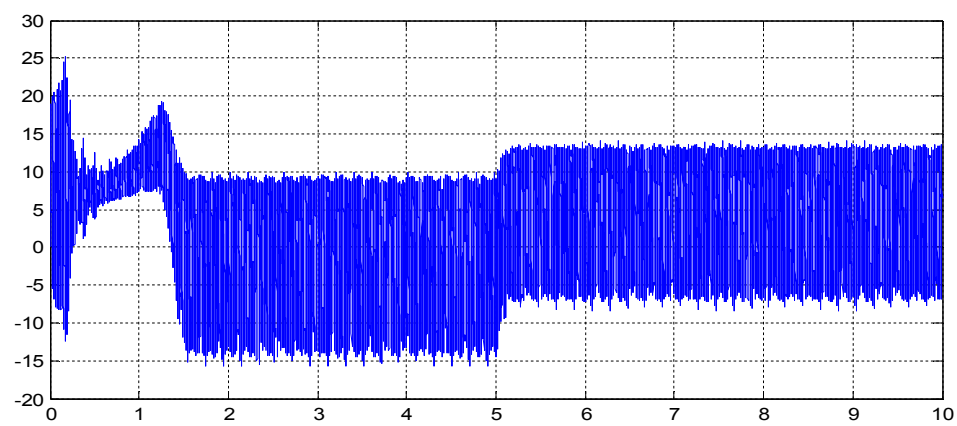


Figure III.19 : La courbe de couple par MLI neural

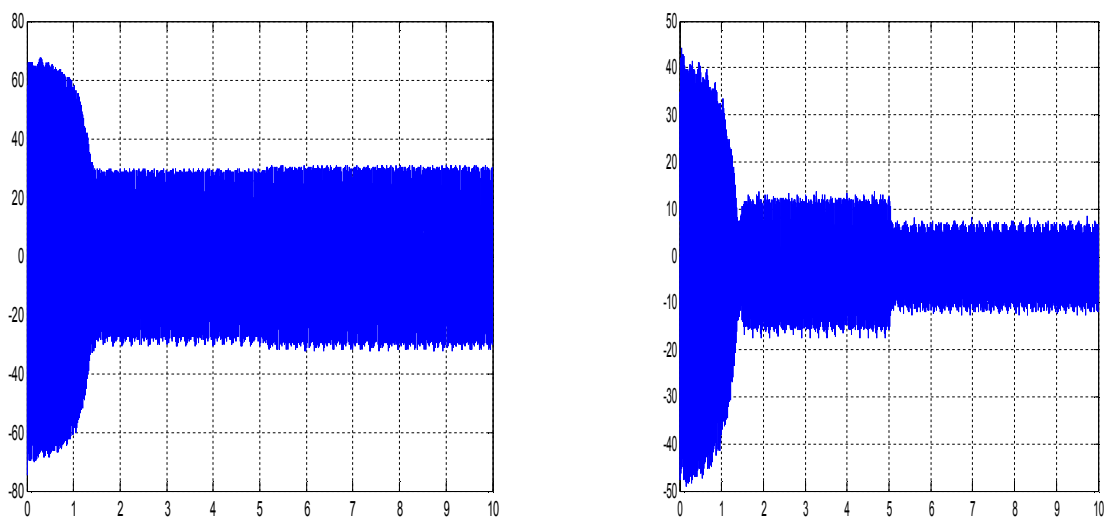


Figure III.20 : Les courbes des courants statorique ids iqs successivement

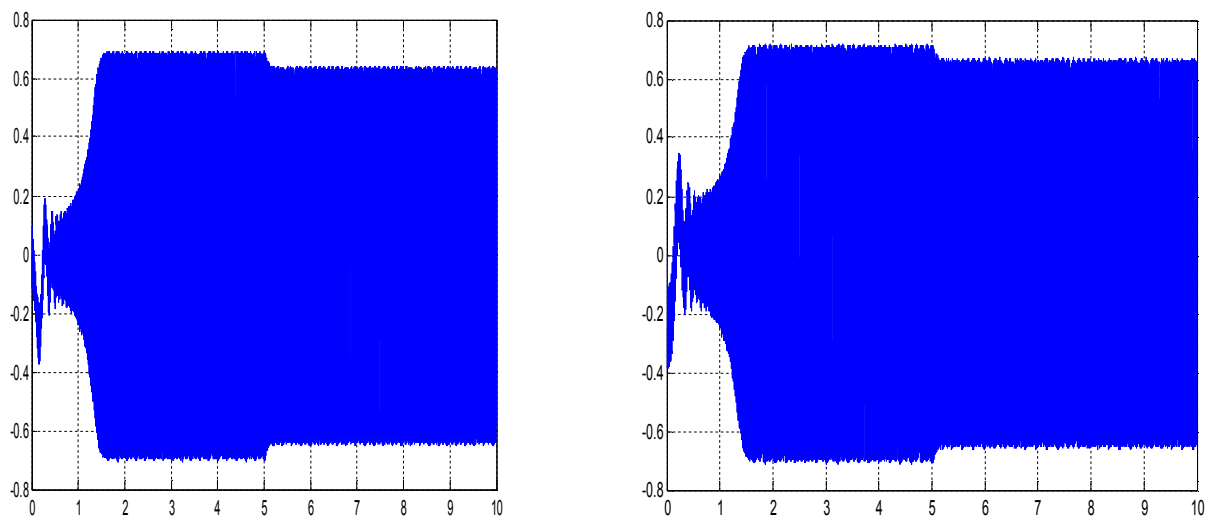


Figure III.21 :les courbes des fréquence rotorique frq et frd successivement

III.12.3.1 Interprétation des résultats :

Dans la figure III.14, on voit la convergence entre la courbe de vitesse qui est générée par MLI neural et la courbe de MLI neural.

De la figure III.17, on voit que la forme du courant dans la charge est très proche d'une forme sinusoïdale.

D'après ces figures, on constate que les angles de commutation de l'OND approximatés par notre réseau de neurones sont très proches des angles exacts qui sont générés par MLI classique.

III.13 Conclusion

Nous avons étudié dans ce chapitre, les aspects théoriques des réseaux de neurones, ainsi que leurs caractéristiques principales. Nous y sommes basés pour élaborer un réseau de neurones artificiel qui génère une commande MLI. Les résultats de simulation montrent les performances de notre algorithme MLI neuronale qui donne une précision dans l'estimation des angles de commutation de l'OND.

Chapitre 04 : Implémentation des RNA sur Circuit FPGA

Chapitre 04 : Implémentation des RNA sur Circuit FPGA

IV.1 Les circuits programmables

La plupart des circuits numériques programmables sont issues de la célèbre architecture « Architecture Von-Neumann » proposée par John Von Neumann en 1945 et porte son nom. Ensuite une autre architecture qui vient pour compenser les lacunes de la précédente dans certains domaines et afin d'améliorer la cadence de calcul où le facteur temps d'exécution est le plus important. Cette architecture nommée « Architecture Harvard » qui porte le nom de l'université américaine qui la propose sachons que parallèlement l'architecture « Von-Neumann » n'est pas figée mais en évolution. Mais le besoin croissant de composants très rapides a orienté les chercheurs à développer une autre solution qui sera complètement différente des deux précédentes architectures. Cette solution réside dans le mode de programmation qui est devenu architectural à logique câblée inversement aux deux premières architectures précédentes où la programmation est séquentielle.

D'une manière générale, il existe deux alternatives ou solutions qui sont :

- ❖ **Une solution logicielle** : Elle est nommée aussi solutions programmables du type « Processeur » où un traitement séquentiel relativement lent et programmation dépendante du composants (DSP, Microprocesseur et Microcontrôleur...).
- ❖ **Une solution matérielle** : Elle est nommée aussi solutions programmables du type « Logique » où un traitement parallèle en temps réel et une programmation architecturale avec un langage de description matériel HDL (Méthodologie de conception CAO) indépendante du composant (ASIC et FPGA ...)

Les caractéristiques de ces circuits sont :

- **Les circuits du type DSP/Microprocesseurs** : Un rapport performance/coût faible, un temps de conception très court et une grande souplesse d'utilisation.
- **Les circuits du type spécialisé ASIC** : Très performants mais avec un cycle de conception longue et une architecture figée.
- **Les circuits du type FPGA** : Des performances proches des ASIC un coût unitaire intermédiaire et un cycle de conception moyen et une architecture modifiable.

Voici un tableau récapitulatif de comparaison des différentes solutions numériques :

	ASIC	FPGA	DSP
Performances	Très élevées	Elevées	Faibles
Taille	Faible	Moyenne	Elevée
Consommation	Faible	Moyenne	Très élevées
Intégration	Système sur puce	Système sur puce	Composants supplémentaires
Souplesse	Fonctions figées	Reconfigurable	Programmable
Mise en œuvre	Complexe	Complexité moyenne	Complexité moyenne
Coût du composant	Très élevé	Moyen	Faible

Tableau (IV.01) : Comparaison des différentes solutions numériques

IV.2 Les circuits logiques programmables du type FPGA

IV.2.1 Critères de choix du circuit programmable FPGA

Les FPGA sont développés récemment grâce aux progrès de la technologie VLSI l'apparition de ce type de circuits est une révolution des systèmes digitaux et ouvrants des perspectives de traitement numérique inaccessibles auparavant. La fin des années 80 a vu l'apparition des premiers circuits FPGA qui sont des circuits intégrés que l'on peut configurer en un temps relativement court pour réaliser n'importe quelle fonction logique « câblée » à bas coût par une programmation de ses cellules logiques et ses interconnexions avec une restriction de ne pas épuiser les ressources du FPGA. Typiquement, un circuit FPGA haute densité peut contenir jusqu'à plusieurs millions d'éléments programmables. Pour réussir une application à base d'FPGA et afin d'obtenir un système plus performant, consommant un minimum de puissance, il est nécessaire de respecter un certain nombre de règles comme :

- Bien connaître les caractéristiques du FPGA ciblé pour assurer son adéquation avec les besoins du projet.
- Elaborer une méthodologie de conception.
- Maîtriser les outils d'implémentation et de choisir des outils de synthèse de qualité.

La conception sur les circuits FPGA est un challenge dans lequel l'objectif est de trouver le bon compromis entre densité, flexibilité et performances temporelles.

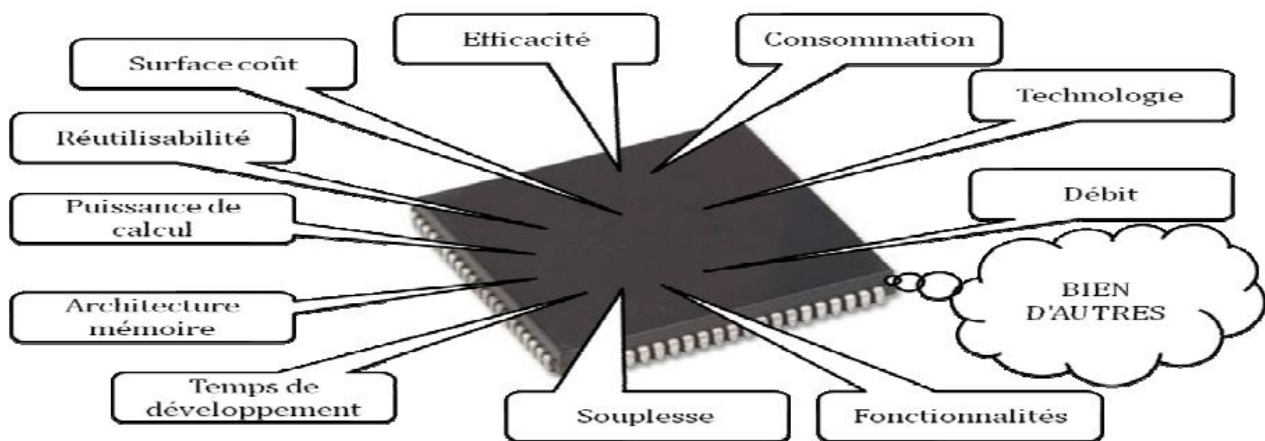


Figure (IV.1) : Critères de choix du circuit logique programmable FPGA.

IV.2.2 Avantages et inconvénients des FPGA

IV.2.2.1 Avantages

- Un circuit reprogrammable : L'avantage du FPGA est de pouvoir être reprogrammable contrairement aux circuits intégrés de type ASIC. Ce qui rend cette solution modulable et donne la possibilité de modifier le programme générique de base afin de le rendre spécifique au circuit utilisé.

- Une solution de validation utilisant le FPGA peut alors convenir à beaucoup de projets et donc diffusée à plusieurs équipes.

- Un investissement rentable dans la durée : Cela est dû à sa reprogrammation, ce qui implique une réutilisation à destination d'autres projets, malgré un prix à l'achat supérieur à un circuit ASIC.

- Une Reprogrammation quasi-instantanée du circuit. Une fois le programme validé cela ne prend que quelques minutes à l'implémenter. A titre de comparaison, la fabrication d'un circuit ASIC peut prendre plusieurs semaines.

Cependant, le FPGA n'est pas le seul composant reprogrammable du marché. Le DSP « Digital Signal Processor », processeur de signal numérique, permet également d'émuler un montage numérique. Le DSP est programmable grâce au langage C, le FPGA utilise quant à lui le VHDL. Le format de description machine est généré automatiquement par les logiciels de développement des concepteurs. Il est possible alors de vérifier les circuits sans avoir à les concevoir par nous-mêmes. L'importation de leur fichier dans notre programme est suffisante. Par ailleurs, le FPGA peut disposer d'un DSP sous forme d'IP incluse dans le système du circuit.

IV.2.2.2 Inconvénients

- Performances non optimisées.
- Temps de réponse long par rapport aux ASIC

IV.3 Architecture interne des FPGA :

L'architecture interne des FPGA est différente d'un fabricant à un autre, mais leurs ressemblances peuvent être représentées dans le schéma de la figure suivante [75] :

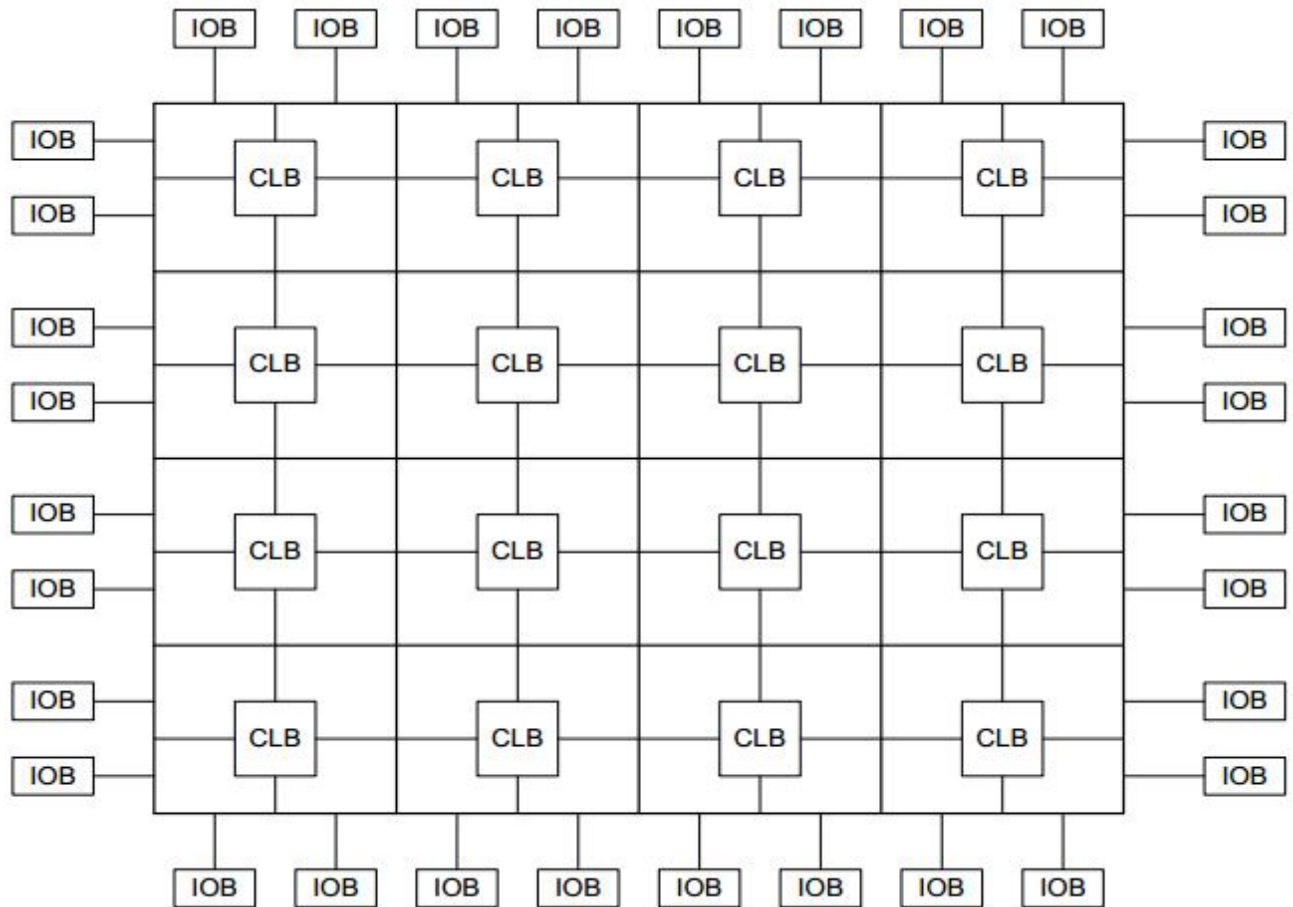


Figure (IV.2) : Architecture interne d'un FPGA.

Les macro-cellules internes sont appelées :

- a) CLB : « Configurable Logic Block », signifiant bloc logique configurable.
- b) IOB : « Input Out put Block », signifiant bloc d'entrée sortie.
- c) PIP : « Programme Interconnect Point », signifiant l'ensemble des points de connexion.

La granularité des FPGA par les macro-cellules CLB nous permet d'implémenter des fonctions logiques combinatoires ou séquentielles complexes par chaque CLB.

IV.3.1 Structure des CLB

Les CLB représentent la ressource principale à implémenter pour la logique séquentielle tout comme pour la logique combinatoire, chaque CLB est constitué de SLICES interconnectées, ces slices sont disposés en paires et chaque paire est disposée en colonne. La paire de gauche est appelé SLICEI et celle de droite SLICEO.

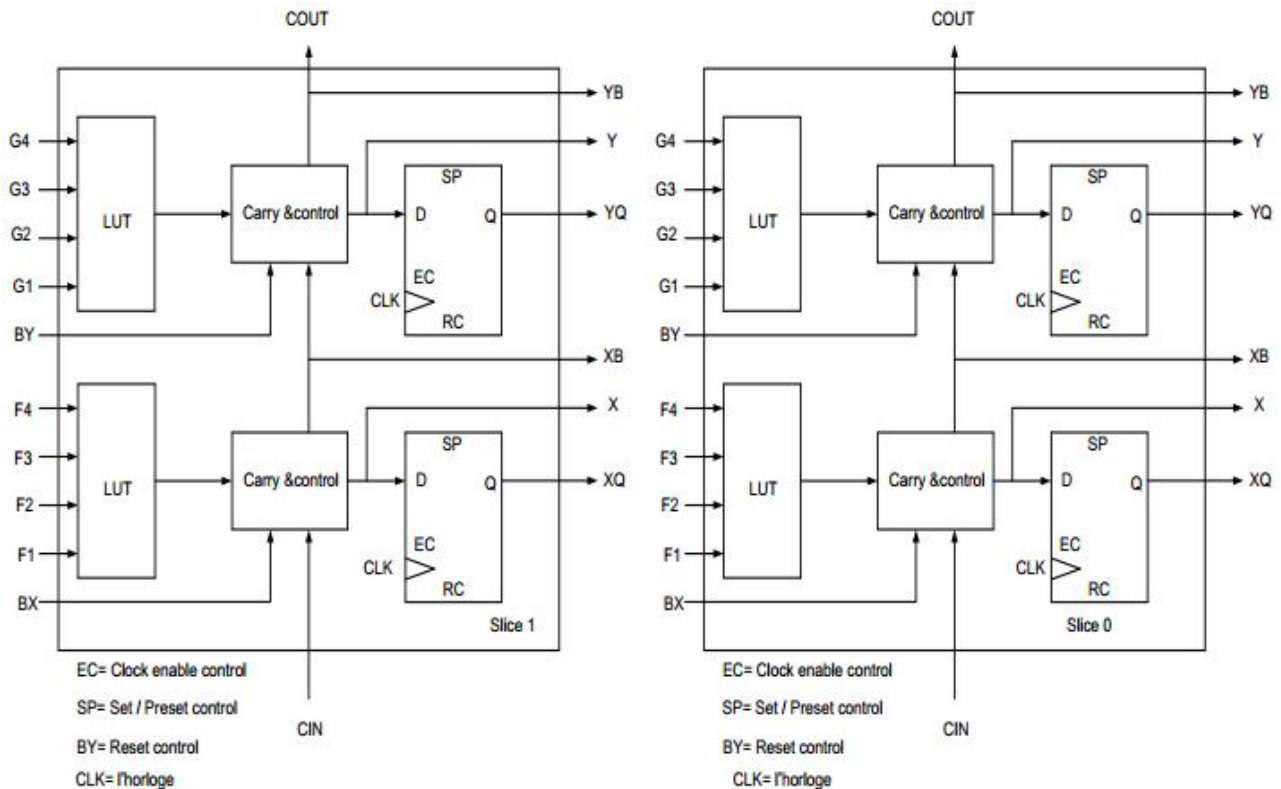


Figure (IV.3) : Architecture d'un CLB de famille Virtex [75].

L'architecture d'un SLICE est comme suit :

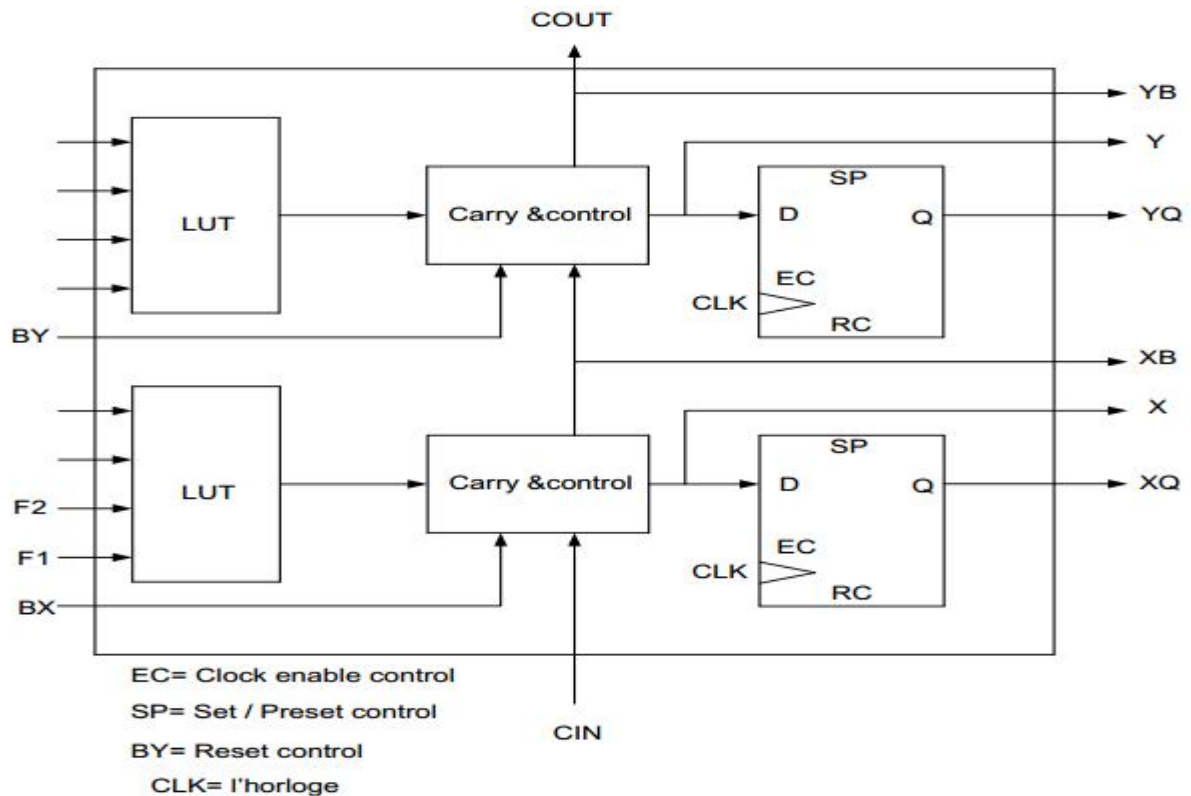


Figure (IV.4) Architecture d'un SLICE de famille Virtex [5].

IV.3.2 Structure des IOB

Les blocs d'entrées/sorties disposent aussi de bascules de contrôle à trois états :

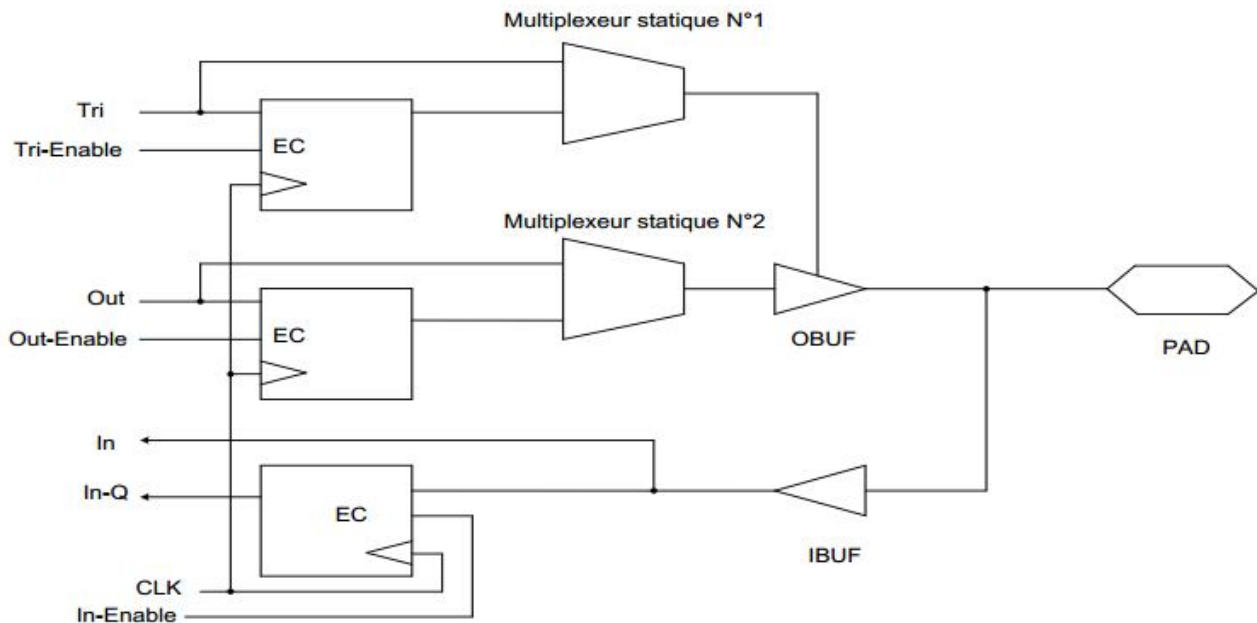


Figure IV.5 : Architecture d'un IOB de famille Virtex [75].

IV.3 Langage VHDL

IV.3.1 Définition

Le VHDL est un langage de description de matériel qui est utilisé pour la spécification (description du fonctionnement), la simulation et la preuve formelle d'équivalence de circuits. Ensuite il a aussi été utilisé pour la synthèse automatique. L'abréviation VHDL signifie VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (langage de description de matériel pour circuits à très haute vitesse d'intégration).

IV.3.2 Pourquoi utiliser le VHDL

Le VHDL est un langage normalisé, cela lui assure une pérennité. Il est indépendant d'un fournisseur d'outils. Il est devenu un standard reconnu par tous les vendeurs d'outils EDA. Cela permet aux industriels d'investir sur un outil qui n'est pas qu'une mode éphémère, c'est un produit commercialement inévitable. Techniquement, il est incontournable car c'est un langage puissant, moderne et qui permet une excellente lisibilité, une haute modularité et une meilleure productivité des descriptions. Il permet de mettre en œuvre les nouvelles méthodes de conception.

Il est à noter toutefois un inconvénient qui est la complexité du langage. En effet, ce langage s'adresse à des concepteurs de systèmes électroniques, qui n'ont pas forcément de grandes connaissances en langages de programmation.

D'autres fausses idées circulent sur le langage VHDL. Celui-ci n'assure pas la qualité du résultat, la portabilité et la synthèse des descriptions. Une méthodologie est indispensable pour combler ces lacunes.

IV.3.3 Unité de conception (module VHDL)

L'unité de conception est un ensemble d'éléments VHDL avec lesquels nous allons décrire un système numérique. Celui-ci peut être constitué d'une simple porte logique jusqu'à un système complexe. Nous parlerons aussi de module VHDL. L'unité de conception est constituée d'une entité (définit l'interface), une ou plusieurs architectures (défini le fonctionnement), des bibliothèques et de la configuration. Les bibliothèques regroupent un ensemble de définition, déclarations, fonctions, etc ... nécessaire à tous les modules. La configuration est optionnelle. Elle ne sera pas utilisée dans ce manuel.

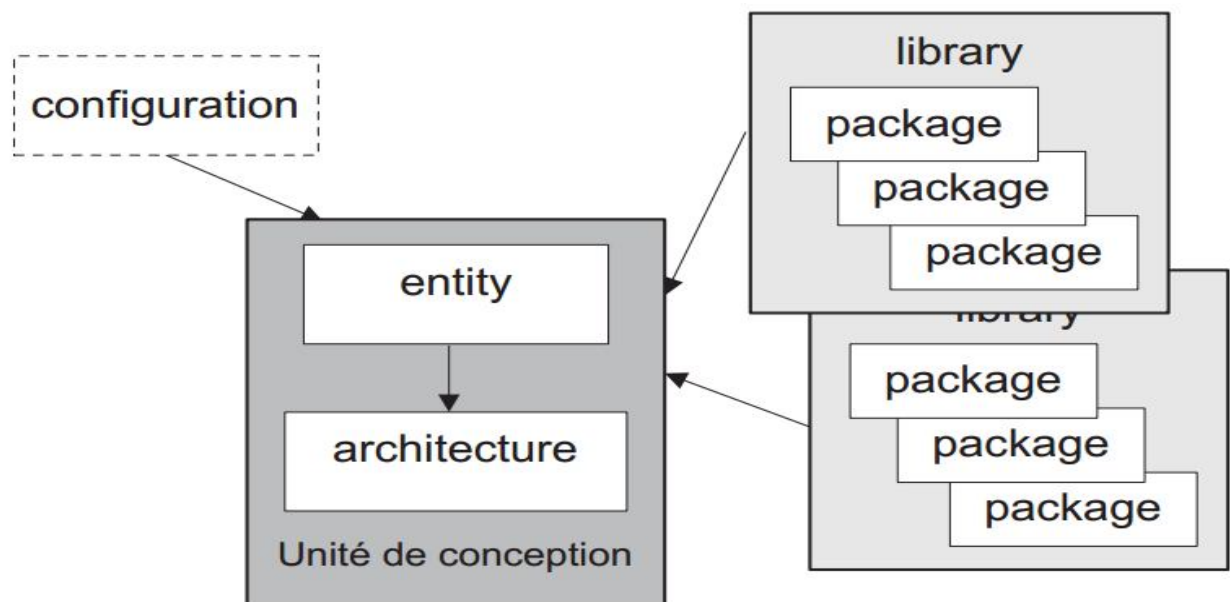


Figure IV.6 : Représentation unité de conception

IV.3.3.1 L'entité

L'entité est une vue externe du module, elle définit toutes les entrées et sorties.

Syntaxe générale de l'entité:

```
entity Norn_Du_Module is
  port ( Norn_Entrée_1 :intype_Du_Signal;
        Norn_Entrée_2 :intype_Du_Signal;
        ...
        Norn_Sortie_1 : outtype_Du_Signal;
        Norn_E_S_1   : inouttype_Du_Signal );

end Norn_Du_Module;
```

Remarque : Les entrées et les sorties d'un module sont toujours représentées par un port, il en existe 4 différents types, qui sont :

- **in** : port d'entrée.
- **out** : port de sortie.
- **inout** : port bidirectionnel (utilisation pour des bus).
- **buffer** : port de sortie qui peut être relu (ne pas utiliser)

Un signal de type buffer ne peut pas être connecté directement avec un signal de type out (le VHDL est typé). Il est nécessaire d'utiliser un cast. Cela n'est pas très pratique. Nous déconseillons l'utilisation de ce type.

IV.3.3.2 L'architecture

L'architecture décrit le comportement du module. Le langage VHDL permet de décrire le comportement de différentes façons. Dans ce manuel nous allons traiter les descriptions synthétisables et les bancs de test.

Nous allons commencer par les descriptions synthétisables. Dans la littérature, ce niveau de description est appelé RTL (Register Transfert Level). Nous distinguerons différentes possibilités de descriptions synthétisables, soit :

- **Logique** : Equations logiques, description du fonctionnement directement avec des équations logiques.
- **TDV** : Table de vérité, description du fonctionnement avec une table de vérité (utilisation de l'instruction with ... select).
- **Flot_Don** : Flot de données, description du fonctionnement très proche du fonctionnement logique (utilisation des instructions concurrentes <=, when ... el se ou with ... select)
- **Comport** : Comportementale, description du fonctionnement en décrivant le comportement (exemple : description avec un process et de l'instruction if ... then ... el se)
- **M_Etat** : Machine d'état, description du fonctionnement à l'aide d'une machine d'état
- **Struct** : Structurale, description du module en donnant les interconnexions entre différents sous modules (description hiérarchique avec l'instanciation de composant)

IV.3.3.3 Syntaxe générale d'une architecture

Voici la syntaxe d'une architecture :

```
architecture Type_D_Architecture of Nom_Du_Module is
--Zone de déclaration
begin
```

```
Instructions_Concurrentes;  
process ( Liste_De_Sensibilité)  
begin  
    Instructions_Séquentielles;  
end process;  
  
end Type_D_Architecture;
```

Dans un circuit logique, toutes les portes fonctionnent simultanément. On dit alors que les portes fonctionnent de manière concurrente, c'est à dire que l'ensemble des opérations se déroulent en parallèle. Il est simple de comprendre que toutes les parties d'un circuit fonctionnent simultanément.

Le langage VHDL a été créé dans le but de pouvoir décrire le comportement d'un circuit. Le langage dispose donc d'instructions concurrentes. Cette notion est particulière au langage de description de matériel. Elle n'existe pas dans les langages de programmation conventionnel.

Parfois, il est plus performant de décrire le comportement en utilisant un algorithme en utilisant une instruction tel que le `if .. then .. else`. C'est la raison de l'existence, dans le langage VHDL, d'une instruction concurrente particulière, l'instruction `process`. Celle-ci permet de décrire le comportement du circuit avec des instructions séquentielles. A l'intérieur de cette instruction le déroulement des instructions est séquentiel.

IV.3.3.4 Les paquetages

Le langage VHDL doit son succès grace aux notions de bibliothèque et de paquetage. Il est ainsi possible de fournir un ensemble de fonctionnalités rassemblées dans un paquetage. C'est le cas par exemple des définitions des types standards indispensables en synthèse, soit :

```
Library IEEE;  
use IEEE.Std_logic_1164.all;
```

Il sera aussi possible de définir son propre paquetage afin de regrouper des définitions, des sous-programmes, ...

IV.4. La configuration des FPGA

IV.4.1 De l'algorithmique à la conception CAO

L'origine du mot algorithme provient du nom latinisé d'Al-Khawarizmi (Abou Jafar Muhammed Ibn Musa al-Khawarizmi médecin arabe du moyen âge). Un algorithme est une suite ou séquence de raisonnements réalisés par un nombre fini d'opérations en termes de temps et de support matériel afin de fournir une solution à certains problèmes. Un algorithme est présenté sous forme d'une prescription qui peut avoir différentes formes (textuelle, graphique, formule mathématique,

diagramme de séquence... etc.). La plupart des algorithmes existants sont orientés vers une implémentation logicielle ce qui est contraire à notre application qui s'agit d'une implantation matérielle. L'objectif des algorithmes sont récapitulés comme suit :

- Transmettre un savoir faire.
- Décrire les étapes à suivre pour réaliser un travail.
- Expliciter clairement les idées.

Les techniques de conception CAO (Conception Assistée par Ordinateur) sont aujourd'hui très éprouvées et largement employées afin de concevoir des circuits électroniques nécessaires à mettre en pratique les connaissances algorithmiques. L'approche moderne pour la conception des circuits (logiques) électriques et la manière d'introduire une fonctionnalité sur un support physique sont confiées aux outils CAO. Les outils CAO sont utilisés pour générer le fichier de configuration des FPGA qui s'appelle (bit-Stream) à partir d'une description de haut niveau. Les principaux rôles confiés aux outils CAO sont : la description, la simulation, la synthèse, le placement et le routage. Un design peut être conçu à l'aide d'un éditeur schématique ou d'un outil de traitement de textes. [76]

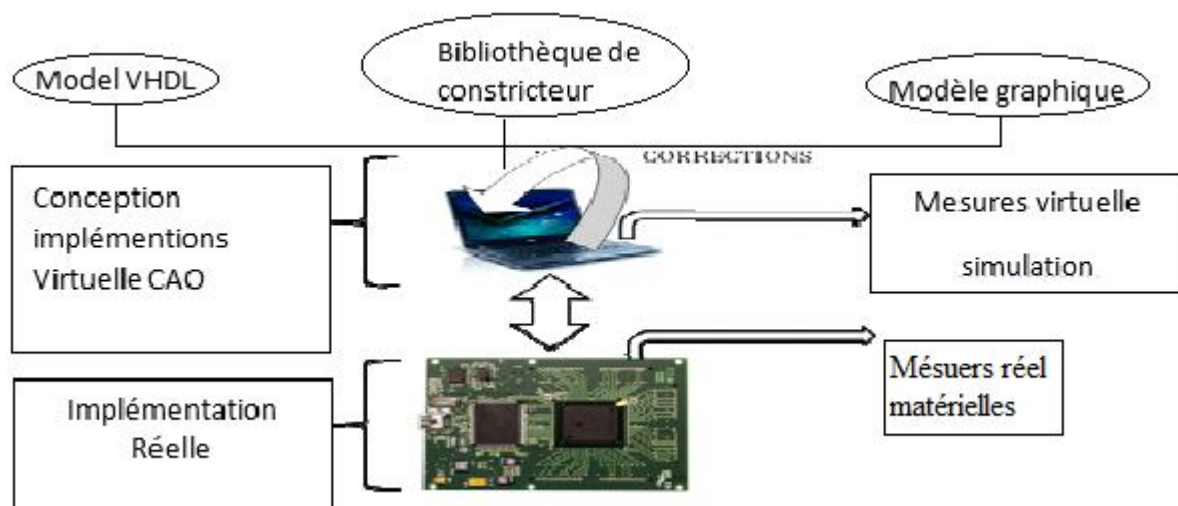


Figure IV.7 : Etapes de conception sur FPGA

La conception de circuit simple peut se faire par l'approche schématique mais dans les circuits complexes elle cède le champ de conception à l'approche textuelle.

IV.4.2 Méthodologie de conception

Le flot de conception d'un système sur puce regroupe plusieurs niveaux d'abstraction. Dans chaque niveau, le concepteur s'intéresse à la résolution d'un problème. Les outils de CAO sont utilisés intensivement et assurent la transition entre les différents niveaux d'abstraction. Nous pouvons traiter un système complexe de deux manières qui sont :

- L'approche dite « descendante » (ou « top-down » en anglais).
- L'approche dite « ascendante » (ou « bottom-up » en anglais).

Le développement d'une application sur FPGA par des outils CAO suit l'enchaînement des étapes suivantes :

IV.4.2.1 Spécification du design

- Le nombre de broches d'entrée-sortie et leur localisation dans la puce FPGA.
- La spécification de la fréquence d'horloge du système.
- La spécification de la mémoire requise pour l'application.

IV.4.2.2 Développement du design

- Spécification de la méthodologie de design (Outil de développement utilisé).
- La saisie du circuit Codage RTL (VHDL, Verilog ...)
 - Graphique (Machine à états).
 - Saisie HDL (Hardware Description Language).
- La simulation (Pré et Post synthèse).

IV.4.2.3 Synthèse

La synthèse est le processus qui convertit la représentation du design à partir du code *HDL* fourni pour produire une représentation au niveau porte logique. Elle s'occupe de déterminer quelles sont les structures susceptibles de répondre au cahier des charges étudié et de produire un code booléen unique sous forme d'un fichier.

IV.4.2.4 Placement et routage

A partir des fichiers de synthèse, l'outil de conception procède au placement et routage. Un algorithme de routage est sensé de faire l'aiguillage des données qu'il reçoit vers leurs destinations par action sur les nœuds de routage ce qui est équivalent à définir les chemins qui relient l'ensemble des **CLB** contenus dans la fonction désirée. Ces algorithmes de routage sont différents d'un concepteur à un autre. Plusieurs traitements sont nécessaires pour obtenir un fichier de configuration :

- **Partitionnement** : Les équations logiques sont partitionnées en un autre ensemble équivalent d'équations. Chaque équation de ce nouvel ensemble peut être implantée dans un seul bloc logique du composant cible **FPGA**.

- **Placement** : Des blocs logiques sont sélectionnés dans la matrice et affectés au calcul des nœuds du réseau booléen.

- **Routage** : Les ressources d'interconnexion sont affectées à la communication de l'état des nœuds du réseau vers les différents blocs logiques qui en ont besoin.

- **Génération des données numériques de configuration** : Les informations abstraites de routage, de placement et les équations implantées dans les blocs sont transformées en un ensemble de valeurs numériques, qui seront chargées sur le composant *FPGA*.

IV.5. Intégration et implémentation

L'implémentation est la réalisation proprement dite qui consiste à mettre en œuvre l'algorithme sur l'architecture du circuit configurable cible, c'est-à-dire à compiler, charger, puis lancer l'exécution sur un ordinateur ou calculateur. C'est une étape de programmation physique et de tests électriques qui clôture la réalisation du circuit. La figure suivante résume un peu l'ensemble de ces étapes.

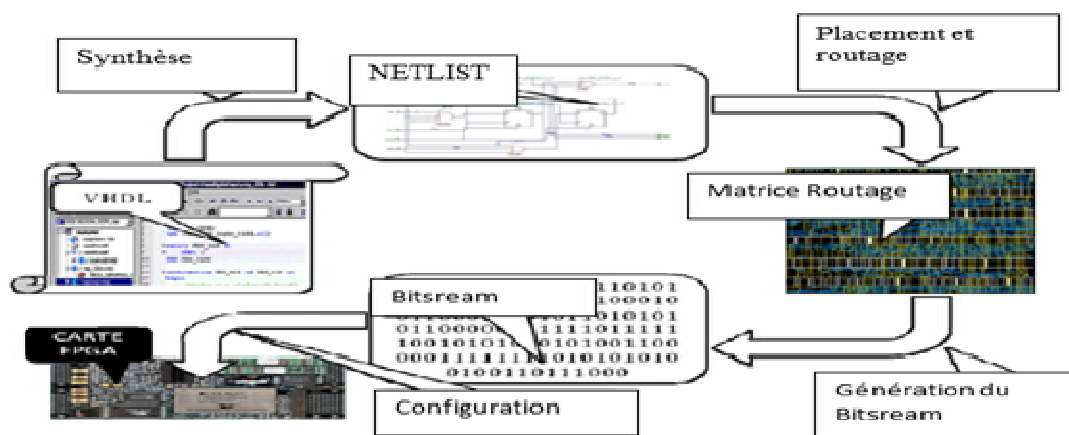


Figure IV.8 : Etapes de conception sur FPGA

IV.6 La structure matérielle de couplage (FPGA /machine)

La recherche dans le domaine de conception des circuits numériques de commande pour les systèmes automatisés en temps réel est difficile, car elle nécessite non seulement une parfaite maîtrise des technologies employées mais aussi une très bonne connaissance des caractéristiques d'application et de la nature de son environnement. La structure générale de couplage FPGA/Moteur est schématisée dans la figure suivante :

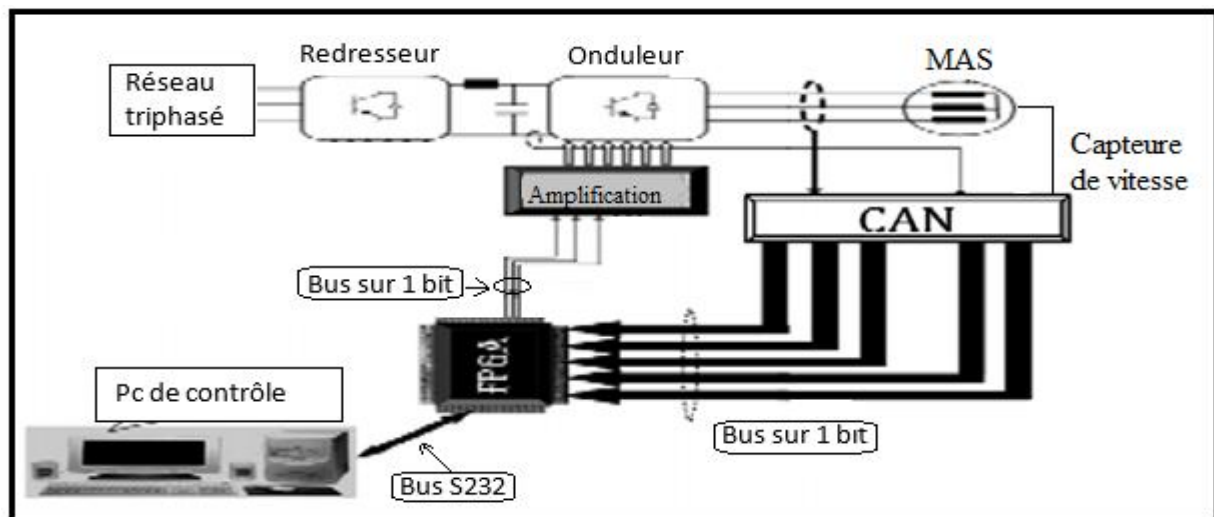


Figure IV.9 : Structure de couplage FPGA/Moteur triphasée.

IV.7 Synthèse de RNA et simulation sur XILINX :

IV.7.1 Génération du code VHDL de bloc RNA

D'après la construction de RNA sous Xilinx le System Generator permet également de générer le code VHDL. 1. On place tout le contenu de chaque couche du bloc de RNA dans le même subsystem avec un seul system Generator (voir Annexe B), ensuite on génère le code VHDL avec les caractéristiques suivantes Compilation :

- HDL Netlist.
- Part : Virtex4 xc4vsx35-1011668.
- Synthesis Tool: XST
- Target Directory: Votre choix de répertoire
- Create Testbench: Checked
- FPGA System Clock Period (ns): 10

2. On ouvre le bloc System Generator:

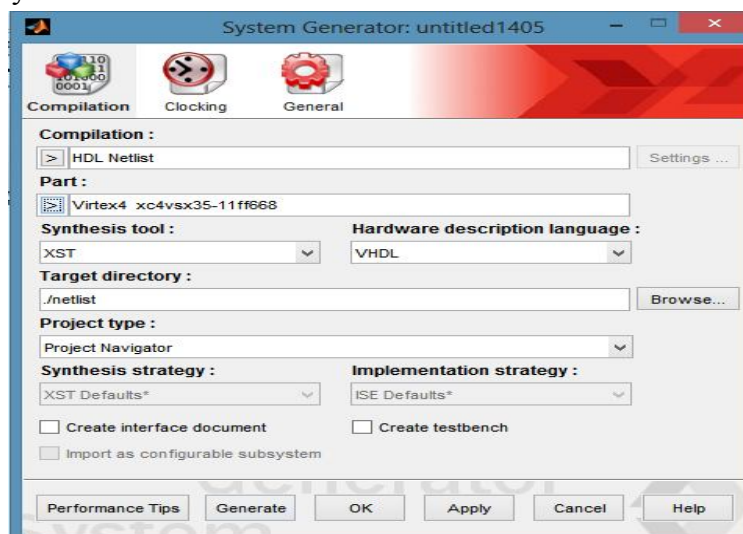


Figure IV.10 : La fenêtre de system Generator.

3. On choisi un répertoire de travail désire dans Target Directory. On met « ./netlist » qui va créer un sous répertoire dans le répertoire courant.
4. On Choisi les paramètres suivant dans le reste de la fenêtre :

- Compilation : HDL Netlist.
- Part: Virtex4 xc4vsx35-1011668.
- Synthesis Tool: XST.

5. On met 10 dans FPGA System Clock Period (ns). Ceci sera la contrainte d'horloge pour le logiciel ISE.

6. A la fin ont Cliqué sur Generate, qui donne le programme en VHDL.

La Figure (IV.11) présente le résultat obtenu :

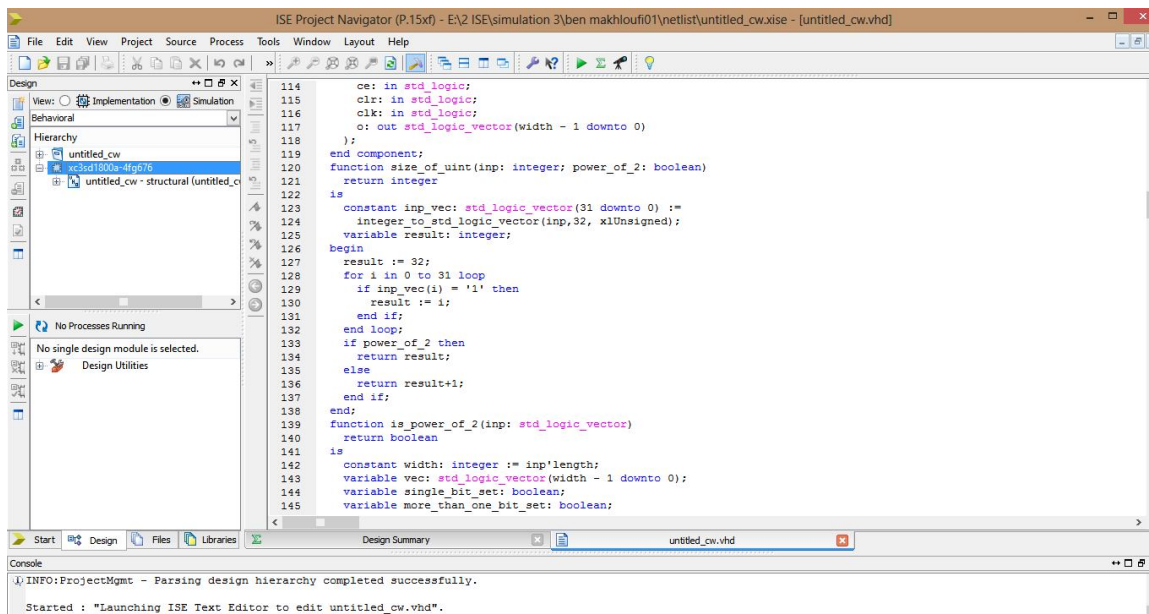


Figure IV.11 : Le résultat de code VHDL obtenu.

IV.7.2 Implémentation de RNA sur FPGA :

Cette partie est consacrée à la description de l'implantation de RNA décrit sur FPGA.

Pour ce faire, nous utilisons le System Generator de Xilinx.

Synthésise : ont Cliqué sur Synthésise pour connaître l'espace des ressources occupées par le code VHDL voir la Figure (IV.11).

Le résultat de Synthésise est affiché dans Le tableau (IV.2).

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	6,633	30,720	21%	
Number of 4 input LUTs	11,264	30,720	36%	
Logic Distribution				
Number of occupied Slices	6,613	15,360	43%	
Number of Slices containing only related logic	6,613	6,613	100%	
Number of Slices containing unrelated logic	0	6,613	0%	
Total Number of 4 input LUTs	11,726	30,720	38%	
Number used as logic	11,264			
Number used as a route-thru	462			
Number of bonded IOBs	115	448	25%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Number of RPM macros	62			

Tableau IV.2 : Le résultat de Synthésise

IV.7.3 Discussions des résultats

D'après les résultats obtenus sur le tableau on remarque que le code VHDL occupe un espace de 21% sur FPGA, signifie que le circuit FPGA de type **Virtex4 xc4vsx35-1011668** supporté ce code VHDL.

IV.8 La co-simulation

Dans le cadre de la co-simulation, on doit pouvoir connecter cette entité avec d'autres simulateurs permettant de reproduire l'environnement externe du composant et ce, quelque soit le niveau d'abstraction fait sur les entrées et sorties. Intégrer le FPGA dans un environnement de co-simulation consiste alors à mettre en place ses interfaces. En effet, la co-simulation est basée sur l'exécution conjointe de simulateurs, chaque simulateur représentant une partie spécifique du circuit à concevoir ou de son environnement.

Une interface de co-simulation doit permettre l'échange de données entre les simulateurs. Le concepteur doit pouvoir tester son circuit, en reliant celui-ci avec d'autres simulateurs sans que cette interface n'intervienne dans les spécifications de son système. Initialement, la co-simulation qui correspond à la simulation conjointe du logiciel et du matériel ne fait intervenir que des composants numériques. La connexion entre les simulateurs se fait de façon particulière en introduisant la notion de " bus de co-simulation". Dans le but d'utiliser la co-simulation, il est nécessaire de construire une interface spécifique entre un simulateur FPGA et un logiciel permettant de simuler le procédé à commander qui, dans notre cas, correspond au logiciel Matlab. [77]

IV.4.1 Définition d'un bus de co-simulation :

La communication entre deux simulateurs peut se faire selon deux modes différents
Le premier consiste à utiliser une relation de type maître-esclave entre les deux simulateurs (ils

s'exécutent alors au sein d'un même processus au sens informatique du terme). Dans ce cas, l'un des simulateurs (par convention défini comme maître) appelle l'autre (alors défini comme esclave) par le biais d'une procédure. Ce type de communication peut s'avérer complexe car il est nécessaire que le simulateur esclave puisse sauver l'ensemble du contexte à la fin de son appel pour pouvoir repartir du même état (en rechargeant le contexte) au prochain appel. Le lancement du simulateur esclave, la sauvegarde et la restitution peuvent s'avérer très coûteux en termes de ressources et temps de simulation. [77]

Le second mode consiste à utiliser un système de simulation distribuée. Les deux simulateurs fonctionnent en parallèle et s'échangent des informations via un canal de communication (Figure IV.12). C'est ce deuxième mode de communication qui est généralement utilisé en co-simulation.

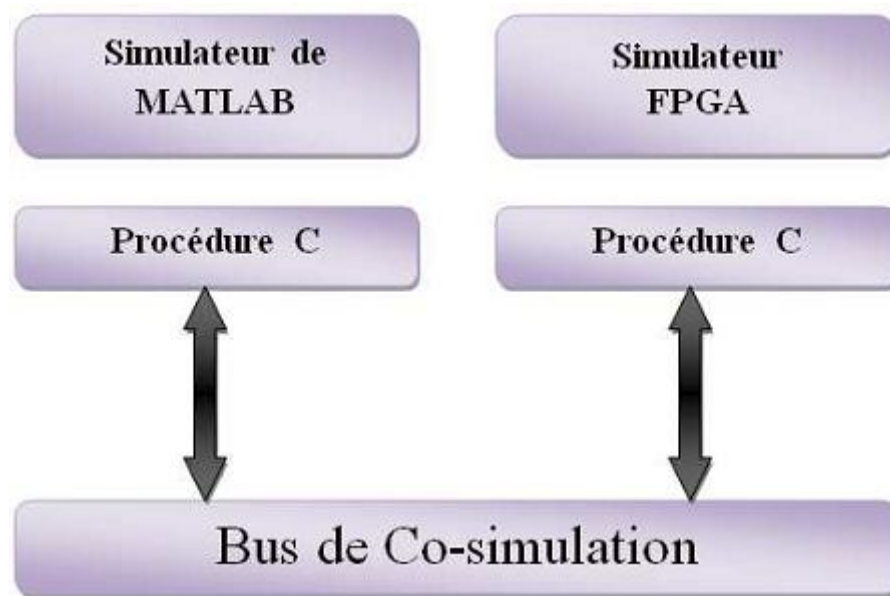


Figure IV .12 : Utilisation d'un bus de co-simulation.

IV.4.2 Les différents types de communication Inter-Processus :

La mise en place d'un canal de communication peut être faite en utilisant différentes technologies informatiques : on peut répertorier les mécanismes de communication interprocessus du système Unix, ainsi que le mécanisme des sockets et des **RPC**. Les sockets peuvent être utilisés sur une station de travail unique mais il est aussi possible de mettre en place des communications réseaux (**Ethernet**) utilisant des protocoles tels que **UDP** ou **TCP**. [77]

IV.4.3 Application de la Co-simulation pour la commande

IV.4.3.1 Etape de la co-simulation :

On a travaillé avec les quatre étapes suivantes :

1. construire le bloc en Xilinx Simulink

2. tester le bloc en XilinxSimulink
3. Générer le bloc de la co-simulation
4. Implémenter le bloc de la co-simulation

IV.4.3.1.1 Construire le bloc en Xilinx Simulink

La figure (IV.13) est le schéma bloc de la fonction sigmoïde sous Xilinx Simulink.

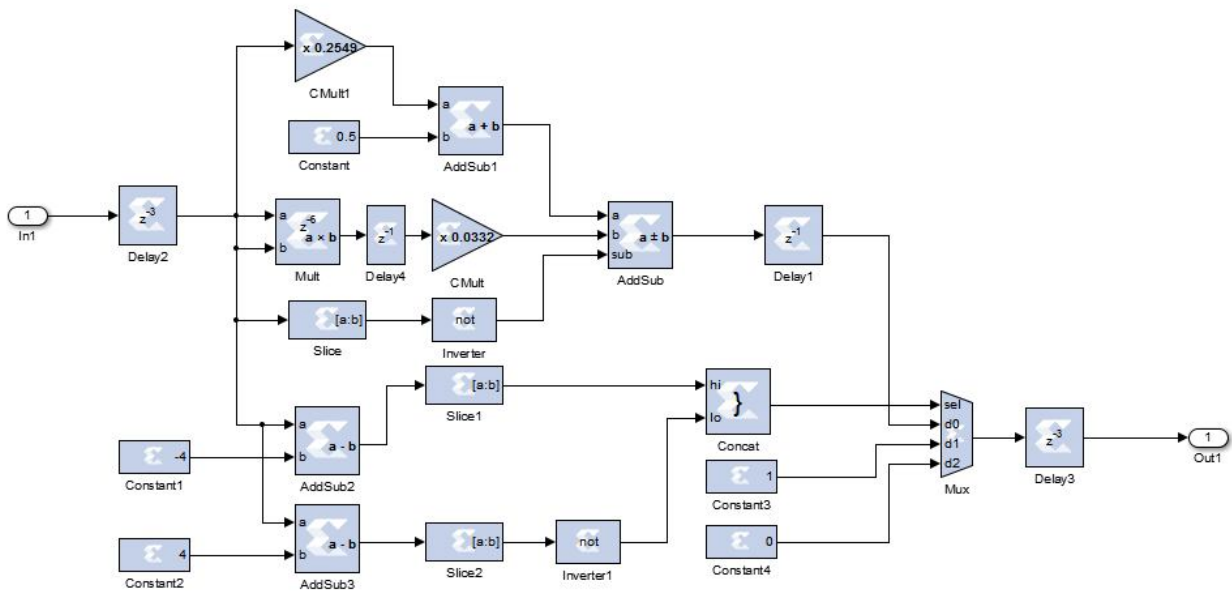


Figure IV.13 : fonction sigmoïde sous Xilinx Simulink

IV.4.3.1.2 Test de réseau fait par Xilinx Simulink

Le teste se fait de la même manière que precedament (test de RNA fait par simulink), on a suivi les conditions du Tableau IV.1.

IV.4.3.1.3 Génération du bloc de la co-simulation

On a appliqué la co-simulation sur la fonction d'activation sigmoïde tel qu'on a suivi les étapes suivantes :

1. Premièrement, on ouvre le bloc System Generator sous Simulink suivant la Figure (IV.14), (voir l'approximation de la fonction sigmoïde dans l'annexe H)
2. On choisit un répertoire de travail désire dans Target Directory. On met. /netlist qui va créer un sous répertoire dans le répertoire courant.
3. On Choisit les paramètres suivant dans le reste de la fenêtre de la Figure (IV.14) :

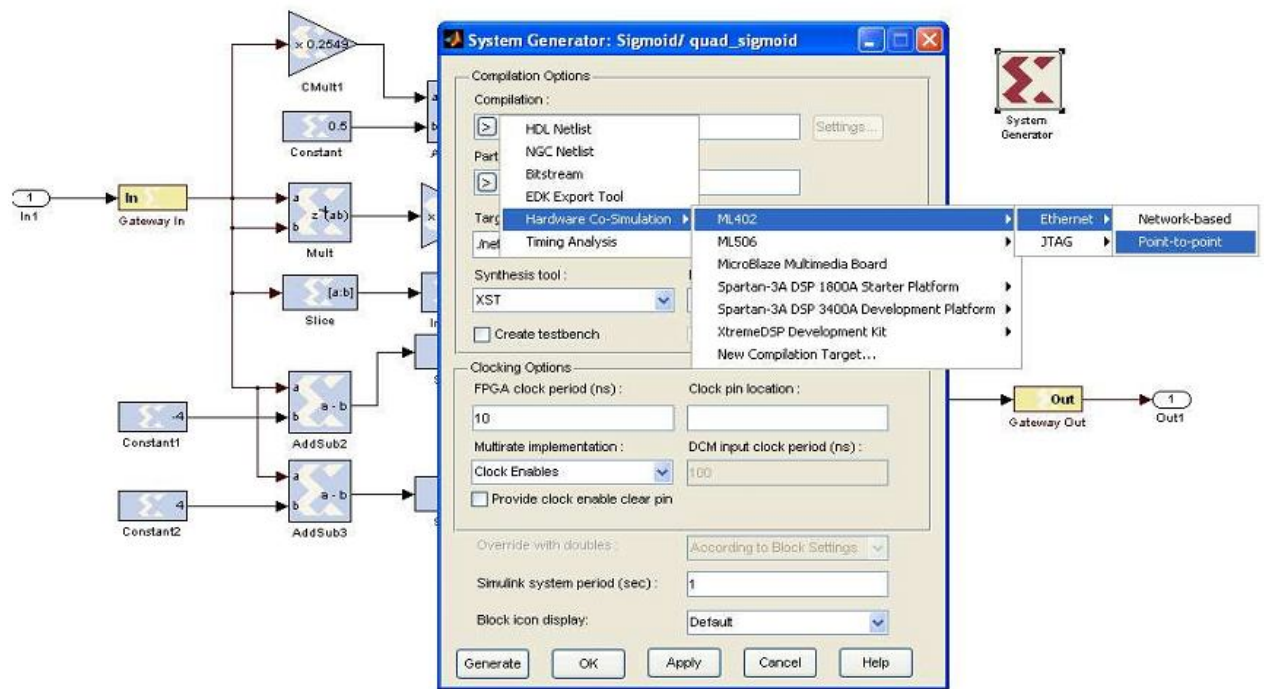


Figure IV.14 : Sous fenêtre de system Generator

4. A la fin ont Cliqué sur Generate, qui donne le bloc de Co-simulation. La Figure (IV.15) présente le résultat obtenu :

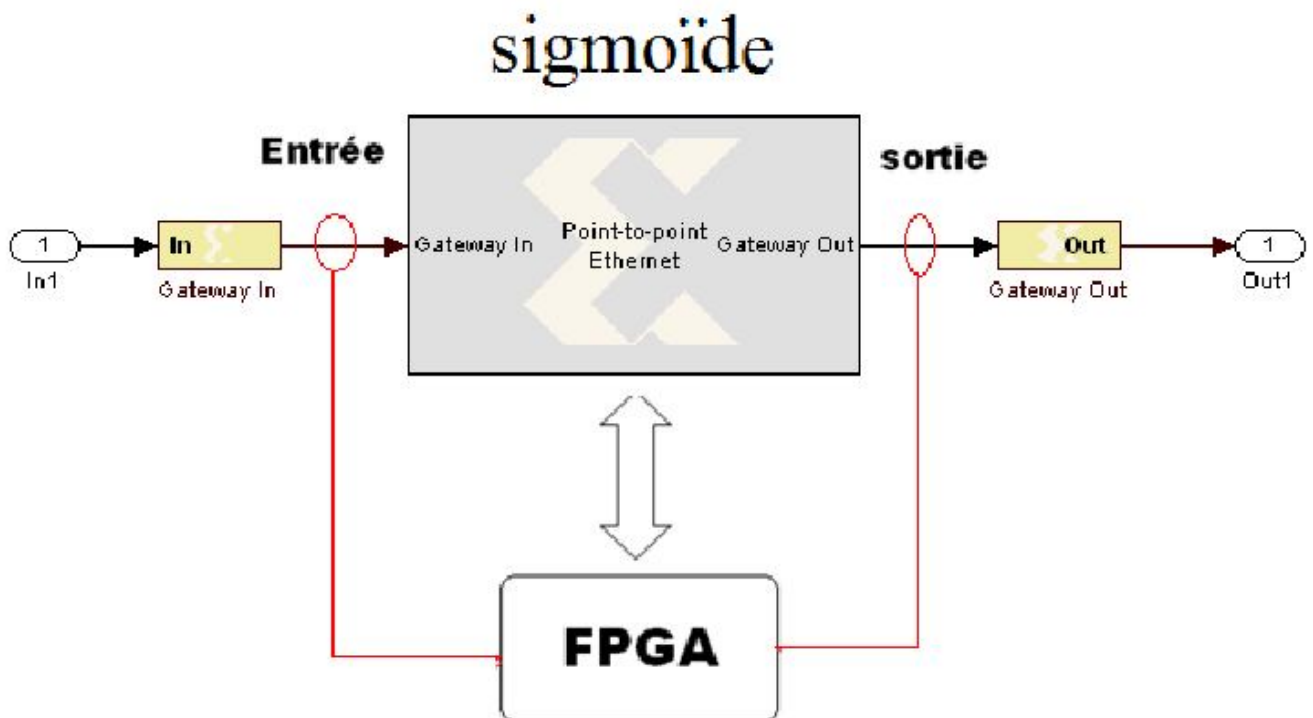
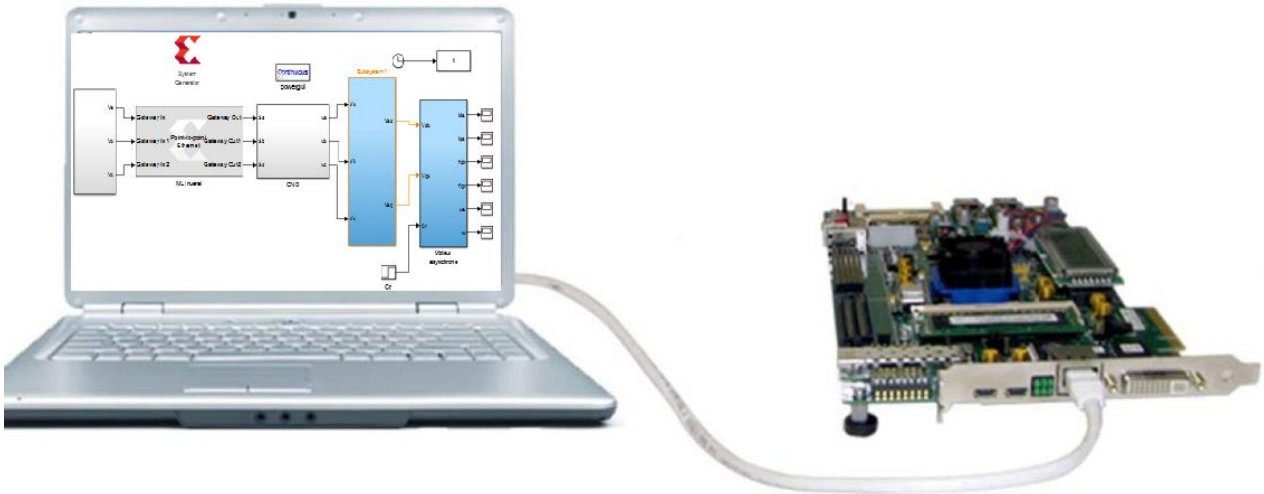


Figure IV.15 : Principe de co-simulation du bloc sigmoïde

IV.4.3.1.4 Implémentation du bloc de la co-simulation

Pour cette dernière phase, on a implémenté le bloc sur un circuit FPGA **Virtex4xc4vsx35-1011668** (voir Annexe G). A travers d'un câble de connexion (ou Ethernet), voir la figure (IV.16).



FigureIV.16 : la co-simulation

IV.5 Résultats de co-simulation

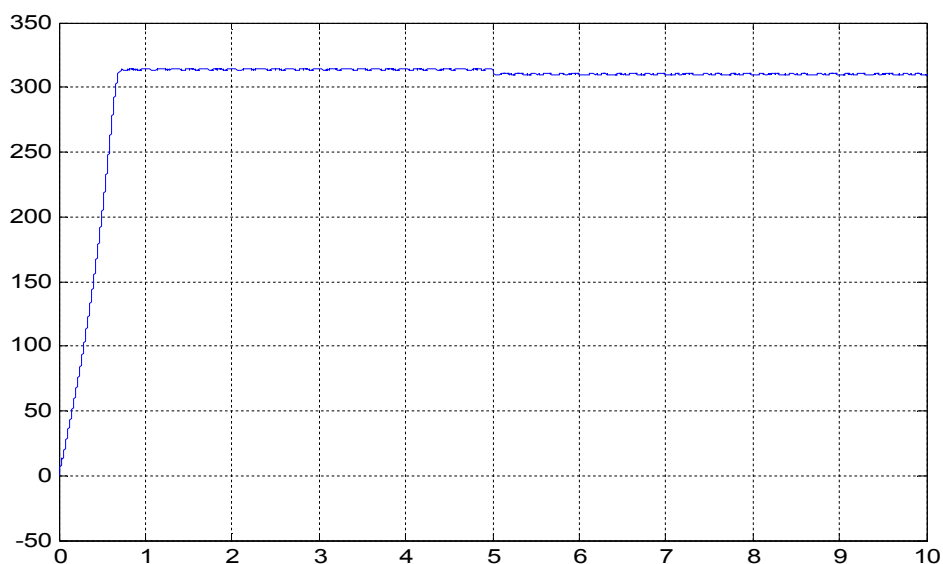


Figure IV.17 : La courbe de la vitesse par la co-simulation

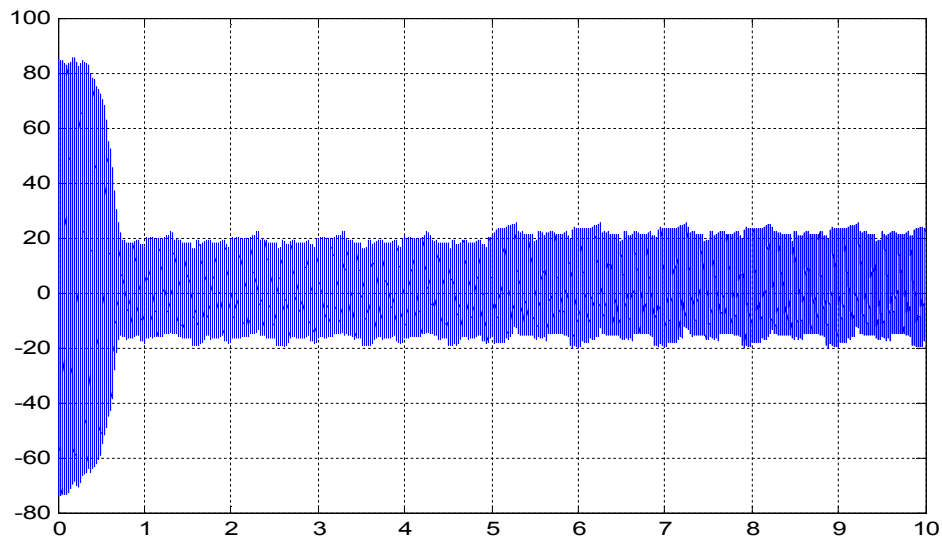


Figure IV.18 : La courbe de courant statorique par la co-simulation

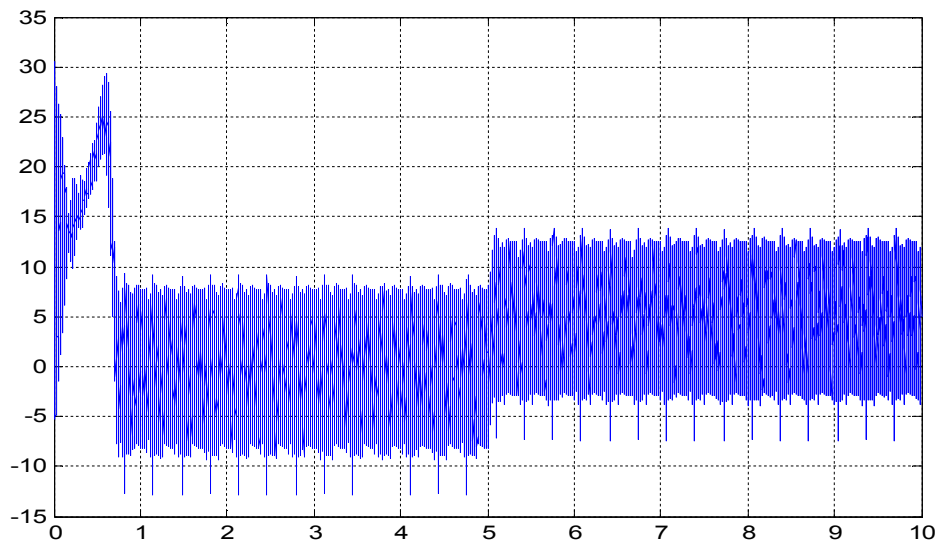


Figure IV.19 : La courbe de couple électromagnétique par la co-simulation

IV.5.1 Interprétation des résultats

Il est appa  t que le test a   t   donn   des bons r  sultats et il est presque le m  me que le teste fait par RNA sous Matlab/simulink.

L'  utilisation de l'outil de conception haut niveau 'System Generator' est tr  s b  n  fique pour la v  rification du comportement de l'algorithme sur Simulink. Les simulations effectu  es pour la fonction sigmo  de ainsi que pour le r  seau de neurones montrent que les r  sultats obtenus sous Xilinx ont donn   les m  mes performances que la fonction sigmo  de obtenus sous Simulink ainsi que pour le r  seau de neurones   tudi  .

IV.6 CONCLUSION :

Dans ce chapitre, nous avons proposé un algorithme simple pour l'implémentation de la RNA. La synthèse matérielle de l'algorithme proposé, est effectuée par le System Generator. Le routage et l'implantation sur FPGA de type **Virtex4** a été effectué par l'ISE foundation.

Une implémentation sur FPGA a été réalisée dans le but de vérifier expérimentalement la génération des signaux de commande. Les résultats d'implémentation montrent une similitude entre les signaux générées par le circuit FPGA et ceux obtenus par Simulink MATLAB.

L'utilisation de l'outil de conception haut niveau 'System Generator' est très bénéfique pour la vérification du comportement de l'algorithme sur Simulink. Les simulations effectuées pour la fonction sigmoïde ainsi que pour le réseau de neurones montrent que les résultats obtenus sous Xilinx ont donné les mêmes performances que la fonction sigmoïde obtenus sous Simulink ainsi que pour le réseau de neurones étudié.

V. Conclusion générale

Sous l'impulsion de progrès rapide et extraordinaire du numérique, l'amélioration de la qualité et des performances a toujours été une préoccupation constante chez les concepteurs de circuits numériques et c'est dans ce contexte que nous exploitons les *FPGA* afin de faire une contribution à l'amélioration de la commande des machines en temps réel. Les contraintes majeures pour la commande des machines sont la satisfaction du compromis rapidité /précision d'une part et un taux de calculs élevé d'autre part. Une approche efficace pour résoudre ce genre de problème est aujourd'hui disponible et concrétiser par ces circuits logiques *FPGA*. L'introduction des *FPGA* est un remède à la complexité des algorithmes de commande ainsi qu'à la vitesse de traitement.

Avant l'apparition des *FPGA* qui sont disponibles aujourd'hui, les méthodes classiques utilisées avec *DSP* et microcontrôleur, permettent l'obtention d'un temps de cycle moyen proche de 100 (us) ce qui est équivalent à une fréquence de commutation aux alentours de 1-5 kHz. Avec l'introduction des *FPGA* à des prix raisonnables, la fréquence de commutation est devenue aux alentours de 10-15 (KHz) et aujourd'hui d'ordre de 50(KHz). Il est même possible d'obtenir des fréquences aussi élevées que 100 (KHz) mais malheureusement les limites des dispositifs d'électronique de puissance sont atteintes sachons que l'énergie dissipée dans ces convertisseurs est proportionnelle à la fréquence de commutation ce qui représente une contrainte très pesante.

Le *FPGA* assure toute la partie algorithmique de la commande grâce à ces caractéristiques notamment le parallélisme de traitement. Parmi les caractéristiques de ce circuit que le contrôle des Machines électriques peut se bénéficier réside dans : la possibilité d'implémenter des fonctions avancées irréalisables dans le domaine analogique, aucun impact des perturbations externes sur les algorithmes, la réalisation de systèmes sûrs et efficaces avec précision, la reprogrammation sur site sans changer de composant ni câblage et un encombrement minimal où tous les algorithmes de contrôles sont intégrés sur une puce de quelques millimètres carrés. L'ensemble de ces caractéristiques est un acquis pour la commande des moteurs par une optimisation du rendement des convertisseurs statiques.

Finalement l'apport des *FPGA* sur la commande des machines peut être résumé dans les quelques points suivant :

- Les *FPGA* sont des solutions numériques qui permettent d'approcher les avantages de l'analogique et de garder au même temps les avantages du numérique avec l'implantation d'algorithmes complexes, un temps de calcul réduit à quelques microsecondes et une bonne précision.
- Eviter l'inconvénient majeur des solutions analogiques classiques qui réside dans l'influence des variations paramétriques engendré par la sensibilité aux perturbations externes comme la chaleur.
- Pas d'entretien qui nécessite du temps et des pertes d'ordres économique à l'inverse des solutions analogiques.

- Implémentation de fonctionnalités supplémentaires qui ne sont pas réalisable en continu.
- Augmentation de la bande passante vis à vis des autres solutions numériques comme les *DSP* et microcontrôleurs ou microprocesseur traditionnels.
- L'intégration sur une seule puce de plusieurs algorithmes de contrôle grâce à la configuration dynamique avec une grande flexibilité pour un changement de la structure de contrôle
- La possibilité de réduire fortement les délais de développement et de commercialisation.
- L'utilisation des *FPGA* dans le contrôle des machines ne nécessite pas d'espace ce qui est équivalent à un encombrement minimal car c'est une technologie embarquée hautement intégrée avec une consommation d'énergie ultra-basse.

Annexe

Annexe A

L'index de modulation

Il parait utile de définir l'index de modulation comme le rapport de la valeur crête de l'amplitude du fondamentale de la tension désirée à la valeur crête de l'amplitude du fondamentale de la tension de sortie de l'onduleur (I-25), cette dernière est normalisée en mode pleine onde ou six-étapes (*six-stepmode*) [78], qui vaut $\frac{2}{\pi} V_{dc}$ (figure.A.1).

$$m = \frac{V_1}{V_{sex_step}} \quad (A.1)$$

Donc :

$$m = \frac{\pi V_{ref}}{2 V_{dc}} \quad (A.2)$$

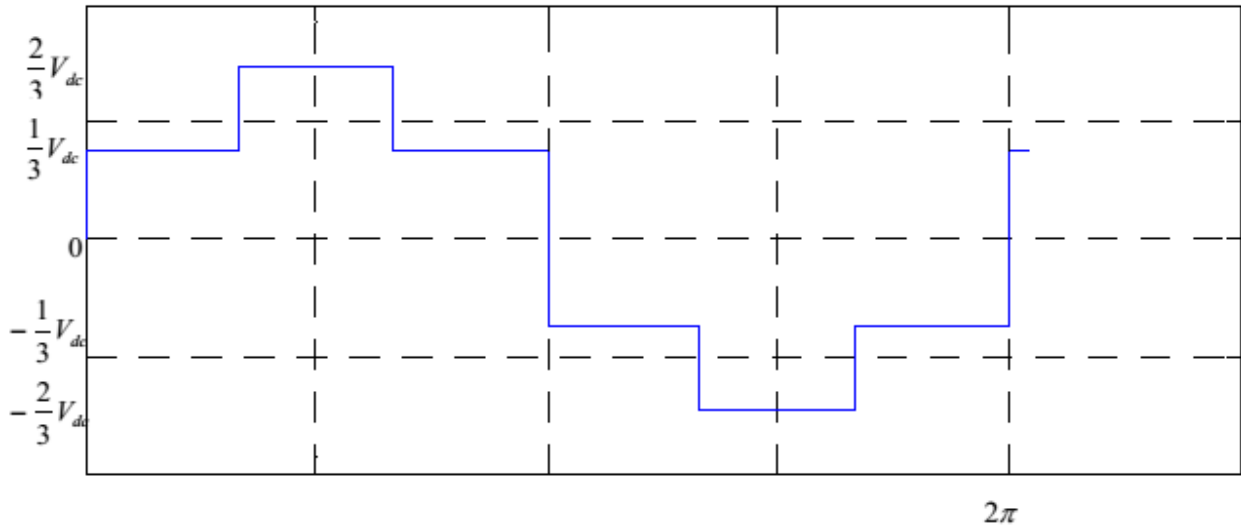


Figure A.1 : Tension Van (six-étapes) aux bornes de la charge

La figure A.2, montre la trajectoire en cercle du vecteur V_{ref} qui par son rayon $\frac{V_{dc}}{2}$ fixe la valeur maximale qu'il peut atteindre par la technique *STPWM*. On voit bien la limite du cercle par rapport à ceux de l'hexagone.

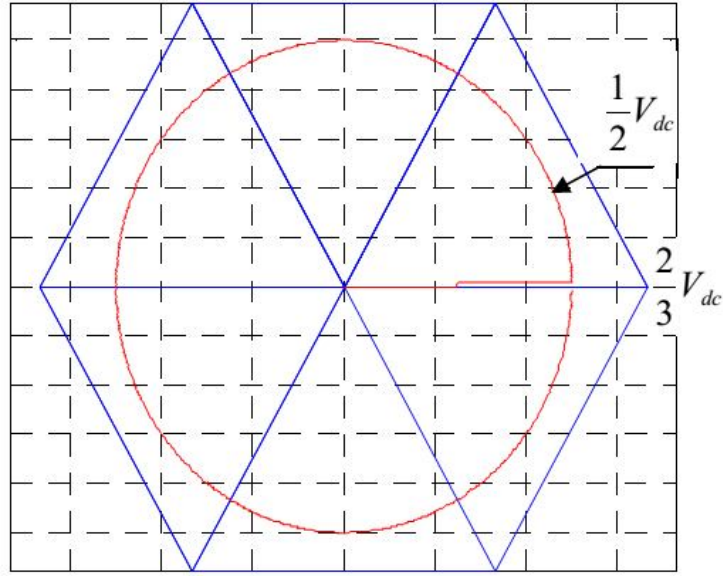


Figure A.2 : Technique STPWM : la tension décrit un cercle de $(V_{dc}/2)$ V_{dc} de rayon

Alors :

$$m_{\max_spwm} = \frac{\frac{V_{dc}}{2}}{(\frac{2}{\pi}) V_{dc}} = 0.7854 \quad (A.3)$$

Pour la *SVPWM*, la tension de référence maximale est calculée au milieu de deux extrémités d'états de tensions actifs, là où le cercle est tangent à l'hexagone (voire figure.A.3), par :

$$V_{ref} = \frac{2}{3} V_{dc} \cos(\frac{\pi}{6}) = \frac{1}{\sqrt{3}} V_{dc} \quad (A.4)$$

Donc le maximum de l'index de modulation est :

$$m_{\max_spwm} = \frac{\frac{V_{dc}}{\sqrt{3}}}{(\frac{2}{\pi}) V_{dc}} = 0.9069 \quad (A.5)$$

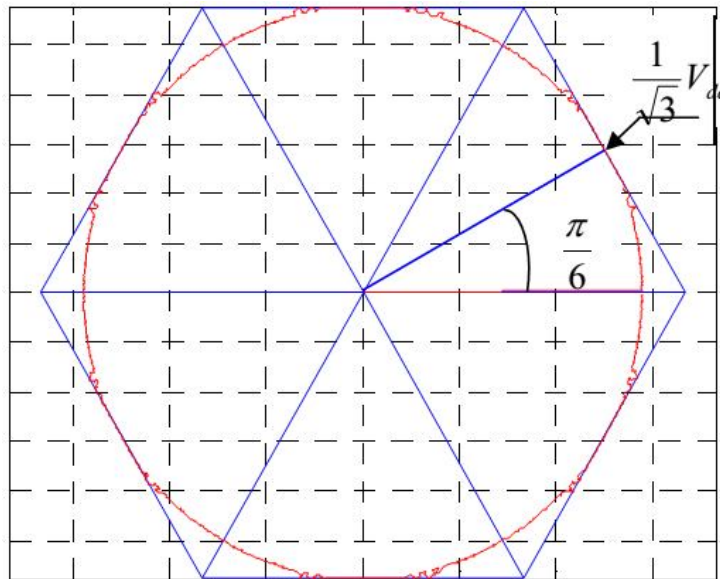


Figure A.3 : Technique SVPWM : la tension décrit un cercle de rayon $(V_{dc}/\sqrt{3})$

Annexe B

B1. Approximation de la fonction sigmoïde

Nous avons l'intention d'approximer la fonction sigmoïde sous forme polynomiale selon la matrice de Vangove, où la fonction (B.1) devient (B.2) : [79]

$$f(x) = \frac{1}{1 + e^{-cx}} \quad (\text{B.1})$$

$$f(x) = c + bx + ax^2 \quad (\text{B.2})$$

Pour la détermination des coefficients c, b, a on applique tout d'abord :

$$\begin{cases} a = \frac{1}{1 + e^{-a^3}} \\ b = \frac{1}{1 + e^{-a^2}} \\ c = \frac{1}{1 + e^{-a^1}} \end{cases} \quad (\text{B.3})$$

avec :

$$\vec{a} = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \text{ est un vecteur qui obtenue a partir de : } \vec{a} = \vec{V} / \vec{F}.$$

Où :

$$\vec{a} = \begin{bmatrix} 1 & 0^1 & 0^2 \\ 1 & 2^1 & 2^2 \\ 1 & 4^1 & 4^2 \end{bmatrix}, \vec{F} = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix}$$

Références

- [01] M. Lamine, A. Traoré, « Modélisation et simulation d'une machine Asynchrone à cage à l'aide du logiciel Matlab / Simulink ».
- [02] J. Caron and J. Hautier. Modélisation et commande de la machine asynchrone. technip Ed., 1995
- [03] J. Lesenne, F. Notelet, and G. Séquier. Introduction à l'électrotechnique approfondie. Technique et documentation, 1981.
- [4] P. Barret. Régimes transitoires des machines électriques tournantes. Paris, France, eyrolles Ed., 1982.
- [05] J. Chatelain and J. Neirynek. Machines électriques. Presses Polytechniques romandes, 1983.
- [06] S. V. Ustun, M. Demirtas "Optimal tuning of PI coefficients by using fuzzy-genetic for V/f controlled induction motor", Elsevier, Expert Systems with Applications 34, pp. 2714–2720, 2008.
- [07] Zhiwen Ma, Zheng T., Fei Lin "Stability Improvement of V/Hz Controlled PWM Inverter-fed Induction Drives", Industrial Electronics and Applications, 2006 1ST IEEE Conference on Volume, Issue, pp.1-4. 24-26 May 2006.
- [08] Y-A- Chapuis, D. Roye, S. Courtine "Commande directe du couple d'une machine asynchrone par le contrôle direct de son flux statorique", Journal De Physique DL T 5, N° 6, June 1995.
- [09] G. Guidi, H. Umida "A Novel Stator Resistance Estimation Method for Speed-Sensorless Induction Motor Drives", IEEE Transactions on Industry Applications, Vol. 36, No. 6, Nov/Dec 2000.
- [10] A. Mezouar, M. K. Fellah, S. Hadjeri, O. Touhami, Y. Sahalim "Robust Direct Field Oriented Control of Induction Motors Using Adaptive Observer", IEEE ISIE 2006, July 9-12, 2006, Montreal, Quebec, Canada.
- [11] A. Sabanovic, B. V. Izoimov "Application of sliding modes to induction motor control", IEEE Trans. Idust. Applic. Vol.IA 13.N°1, pp.41, 1981.
- [12] Mitzova D., Mira C., de Fornel B. "Speed Control of an Asynchronous Motor by Using Variable Structure Control", Systems Engineering in the Service of Humans apos,, Conference Proceedings., International Conference on Volume , Issue , vol.4. pp.447 – 452, Oct 1993.
- [13] N. Madani, M.F. Benkhoris, S. Siala, M.O. Mahmoudi "Sliding Mode Control of an Asynchronous Motor Drive", Power Electronics and Variable Speed Drives, Conference Publication No. 456 IEE, Sept. 1998.
- [14] Mohamed M. Negm "Integral VSC and Preview Control of Efficiency and Speed for a DC Drive" Power SystemTechnology, 2002. Proceedings. International Conference on Volume 4, Issue , 2002.
- [15] Mahieddine-Mahmoud S., Ramanou R., Kefsi L., Chrifi-Alaoui L. "Sliding mode control: a new contribution using an integral action in the boundary layer", Industrial Technology. IEEE International Conference on Volume, Issue, pp.681-686, Dec. 2006

Références

- [16] A. Hazzab, I. K. Bousserhane, P. Sicard “Fuzzy Soft-Switching Law of an Adaptive Sliding Mode Controller for Induction Motor Speed Control”, IEEE ISIE, Montreal, Quebec, Canada, Jul. 2006
- [17] Tian-Jun Fu, Wen-Fang Xie “A novel sliding-mode control of induction motor using space vector modulation technique”, The Instrumentation, Systems, and Automation Society (ISA) Transactions 44. pp. 481–490, 2005.
- [18] Zhiwen Ma, Trillion Zheng, Fei lin, Xiaojie You “A New Sliding-mode Current Controller for Field Oriented Controlled Induction Motor Drives”, Industrial Electronics Society (IECON). 31st Ann. Con. of IEEE Publication Nov. 2005.
- [19] Alkorta P., Barambones O., Garrido A. J., Garrido I. “SVPWM Variable Structure Control of Induction Motor Drives”, Industrial Electronics, ISIE. IEEE International Symposium on Volume, Issue, pp.1195 – 1200, June 2007.
- [20] A. Arias, L. Romeral, E. Aldabas, M. Jayne “Stator flux optimised Direct Torque Control system for induction motors”, ELSEVIER, Electric Power Systems Research 73. pp.257–265. 2005.
- [21] F. Zidani, R. Nait Said “Direct Torque Control of Induction Motor with Fuzzy Minimization Torque Ripple”, Journal of Electrical Engineering, Vol. 56, No. 7-8, pp.183–188, 2005.
- [22] D. Casadei, G. Grandi, G. Serra, A. Tani “Effects of Flux and Torque Hysteresis Band Amplitude in Direct Torque Control of Induction Machines”, IEEE, IECON, Bologna, Italy, sep. 1994.
- [23] Yi Wang, Heming Li, Xinchun Shi “Direct Torque Control with Space Vector Modulation for Induction Motors Fed by Cascaded Multilevel Inverters”, IEEE Industrial Electronics, 32nd Ann. Conf. pp. 1575 – 1579, Nov. 2006.
- [24] Jun Zhang, Rahman M.F. “Analysis and Design of a Novel Direct Flux Control Scheme for Induction Machine”, IEEE International Conference, Electric Machines and Drives, pp. 426 – 430, May 2005.
- [25] D. Casadei, G. Serra, A. Tani “Implementation of a Direct Torque Control Algorithm for Induction Motors Based on Discrete Space Vector Modulation”, IEEE Transactions on Power Electronics, Vol. 15, NO. 4, Jul. 2000.
- [26] Toufouti Riad “Contribution à la Commande Directe du Couple de la machine asynchrone ”, thèse de Doctorat de l'Université de Constantine 2008.
- [27] Lin Chen, Kang-Ling Fang, Zi-Fan Hu “A Scheme of Fuzzy Direct Torque Control for Induction Machine”, IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, Aug. 2005.
- [28] G. Buja, D. Casadei, G. Serra “Direct Torque Control of Induction Motor Drives”, IEEE Catalog Number: 97TH8280, Guimaras, Portugal, ISIE 97
- [29] D. Casadei, F. Profumo, G. Serra, A. Tani “FOC and DTC: Two Viable Schemes for Induction Motors Torque Control”, IEEE Transactions on Power Electronics, Vol. 17, no. 5, Sept. 2002.
- [30] Ramón Blasco Giménez “High Performance Sensorless Vector Control of Induction Motor Drives”, Thèse de Doctorat Nottingham 1995.

Références

- [31] M. W. Degner, R. D. Lorenz, "Using multiple saliencies for the estimation of flux, position and velocity in AC machines," IEEE Trans. Ind. Applicat., Vol. 34, pp. 1097–1104, Sept./Oct. 1998.
- [32] Sbita Lassaâd, Ben Hamed Mouna "An MRAS - based full Order Luenberger Observer for Sensorless", ICGST-ACSE Journal, Volume 7, Issue 1, May 2007.
- [33] Michael Hilaret "Application des outils du traitement de signal à la commande des machines tournantes ", Thèse de Doctorat, Université de Nantes 2001.
- [34] Miran Rodic, Karel Jezernik, "Speed-Sensorless Sliding-Mode Torque Control of an Induction Motor", IEEE Transactions on Industrial Electronics, Vol. 49, No. 1, Feb. 2002.
- [35] Ahmad Razani Bin Haron "Simulation Of Mras Based Speed Sensorless Estimation Techniques For Induction Machine
- [36] C. Schauder, "Adaptive speed identification for vector control of induction motors without rotational transducers", Proc. IEEE Industry Applications Society Annual Meeting, San Diego, California, pp.493- 499, 1989.
- [37] H. Tajima, Y. Hori, "Speed sensorless field-oriented control of the induction machine", IEEE Trans. Ind. Applicat., vol. 29, pp. 175–180, Jan./Feb. 1993.
- [38] Peng F., Fukao T., "Robust Speed Identification for Speed Sensorless Vector Control of Induction Motors", IEEE Industry Applications Society Annual Meeting, Vol.1, pp. 419-426. 1993.
- [39] Suman Maiti, Chandan Chakraborty, Sabyasachi Sengupta "Adaptive Estimation of Speed and Rotor Time Constant for the Vector Controlled Induction Motor Drive Using Reactive Power", The 33rd Ann. Conf. IEEE Industrial Electronics Society, Taipei, Taiwan Nov. 2007.
- [40] Barambones O., Garrido A.J., Maseda F.J. "A Robust Field Motor with Flux Oriented Control of Induction Observer and Speed Adaptation", Emerging Technologies and Factory Automation, IEEE Conference, vol.1. pp.245– 252, Sept. 2003.
- [41] Marek Jasiński "Direct Power and Torque Control of AC/DC/AC Converter-Fed Induction Motor Drives", Thèse de Doctorat (Faculty of Electrical Engineering) Warsaw – Pologne, 2005.
- [42] Bimal K. Bose "Modern Power Electronics and AC Drives" Edition Prentice Hall PTR 2002, ISBN 0-13-016743-6.
- [43] Timothy L. Skvarenina "The Power Electronics Handbook", Edition CRC Press LLC 2002, ISBN 0-8493-7336-0.
- [44] Pierre Brosselard "Conception, Réalisation et Caractérisation d'interrupteurs (thyristors et JFETs) haute tension (5kV) en carbure de silicium ", Thèse doctorat de l'INSA de Lyon 2004.
- [45] Kwok K.Ng "Complete Guide to Semiconductor Devises", Edition McGraw-Hill ISBN 0-17-113527-8.
- [46] Franck Nallet "SiC pour l'Électronique de puissance du futur", RE 3, Techniques de l'Ingénieur.
- [47] Peter K. Steimer et al. "IGCT-une nouvelle génération de thyristors pour des onduleurs de forte puissance à moindre coût", ABB Industrie AG, Revue ABB 5/98.

Références

- [48] Rashid, Muhammad H. "Power Electronics Handbook." Academic Press Series in Engineering 2001, ISBN: 0-12-581650-2.
- [49] Fairchild Semiconductor "AN9010 MOSFET Basics", April 1999. www.fairchild-direct.com.
- [50] Serge Pittet "Modélisation Physique d'un Transistor de Puissance IGBT -Traînée en Tension à L'enclenchement", Thèse doctorat EPFL Lausanne 2005.
- [51] STMicroelectronics "IGBT Basics, AN1491 - Application Note", 2001. <http://www.st.com>
- [52] Motto, E.R., Donlon, J.F., Majumdar, G., Hatae, S. "A New Intelligent Power Module with Microprocessor Compatible Analog Current Feedback, Control Input, and Status Output Signals", IEEE Thirty-First Industry Applications Conference, vol.3, pp. 1287 – 1291, Oct 1996
- [53] François Bernot "Composants de l'électronique de puissance", E3960 Technique de l'Ingénieur.
- [54] Mitsubishi Electric Corporation "Power Electronics Edition", Mitsubishi Electric Advance, Vol. 97, ISSN 1345-3041, Mar. 2002.
- [55] Analog Devices "Implementing Space Vector Modulation with the ADMC33 (AN331-17)" Analog Devices Inc. Jan 2000.
- [56] Mariusz Malinowski "Sensorless Control Strategies for Three - Phase PWM Rectifiers", Thèse de Doctorat (Faculty of Electrical Engineering), Warsaw – Pologne, 2001.
- [57.6] Mr Benslimane Tarak « thèse de doctorat 2009 université de bomerdes ».
- [58] Réalisation d'un ensemble à vitesse variable de dynamique élevée. Conception de la machine synchrone, mise en œuvre de son alimentation et étude de la boucle d'autopilotage. Thèse de Docteur de L'INPT- 1986.
- [59] : R.Haddouche «Simulation et contrôle d'une colonne d'absorption» Thèse de magistère, ENP, 2006
- [60] D.Nauck.et R.Kurse, "A Neuro-Fuzzy Approach to Obtain Interpretable Systems for function Approximation", IEEE International Conference on Fuzzy Systems, 1998.
- [61] Claude.Touzet, "Les réseaux de neurones artificiels, introduction au connexionnisme, cours, exercices et travaux pratiques", préface de Jeanny Hérault, EC2 éd. 150 pages, Paris, 1992.
- [62] Gilles Zwingelstein, "Diagnostic des défaillances, théories et pratique pour les systèmes Industriels". Collection : Traité des nouvelles technologies, Editeur : Hermes Sciences Publicat, 601 pages, 1995.
- [63] McCulloch, Pitts.W, "A logical calculus of the ideas imminently in nervous activity", Bulletin of Mathematical biophysics, Vol.9, pp. 127-147, 1943.
- [64] Hebb.D.O, "The Organization of Behavior, a Neuropsychological Theory", New York : John Wiley, 1949.

Références

- [65] E. Olasagasti « application des réseaux de neurones à l'identification d'un axe de machine-outil » THESE pour obtenir le grade de DOCTEUR DE L'INPG Spécialité : « Génie électrique » Préparée au Laboratoire d'Electrotechnique de Grenoble dans le cadre de l'Ecole Doctorale « Electronique, Electrotechnique, Automatique, Télécommunication, Signal ». Novembre 2002
- [66] Dr. Khodja djalal eddine «Elaboration d'un Système Intelligent de Surveillance et de Diagnostic Automatique en Temps Réel des Défaillances des Moteurs à Induction », THESE pour obtenir le grade de DOCTEUR Spécialité : « Génie électrique » avril 2007.
- [67] Dave Anderson & George Mc Neill, "Artificial neural networks technology", ADACS State-of-the-Art Report (Data & Analysis Center for Software), Kaman Science Corporation, 1992.
- [68] Jean-François Joduin, "Les réseaux neuromimétiques : modèles et applications", HERME, 1994.
- [69] A. Mahmoud, "Identification d'un arc par les réseaux de neurones ", Mémoire d'ingénieur Université de Setif, 1999.
- [70] Menasria.A et Boualleg.AH, "Commande d'un processus par modèle inverse neural" Mémoire d'ingénieur, Université de Geulma, 2001.
- [71] "Les réseaux de neurones artificiels", [www.up.univmrs.fr/Local/umr_6149/umr/pageperso/Les réseaux de neurones artificiels. Pdf](http://www.up.univmrs.fr/Local/umr_6149/umr/pageperso/Les%20réseaux%20de%20neurones%20artificiels.Pdf)
- [72] Dr. KHODJA Djalal Eddine, Implementation of Optimized Approximate Sigmoid Function on FPGA Circuit to use in ANN for Control and Monitoring. Romania CEAI, Vol.17, No.2 pp. 64-72, 2015
- [73] Saha Saif Allah, "Identification des dispositifs hyper-frequences par réseaux de Neurones" Mémoire d'ingénieur, Setif, 2006.
- [74] D. W. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters", Journal of the Society for Industrial and Applied Mathematics, Vol: 11. N: 2, pp: 431-441, 1963.
- [75] : Z.Ait wali «Application des FPGA à la commande d'un moteur asynchrone» Mémoire de magister UMMTO Tizi-ouzou.
- [76] PHILIPPE POURE, RAFIK KADRI, LUC HEBRARD, FRANCIS BRAUN : « Méthodologie d'intégration de commandes numériques pour dispositifs d'Électronique de Puissance basée sur l'utilisation du langage VHDL-AMS ». Laboratoire d'Électronique et de Physique des Systèmes Instrumentaux Université Louis Pasteur.
- [77] K. Renovell, P. Faure, J.M. Portal, J. Figueras and Y. Zorian, "IS-FPGA: A New Symmetric FPGA Architecture With Implicit SCAN" , IEEE ITC International Test Conference, Paper 33.1, 0-7803-7169-0/2001.
- [78] Analog Devices “Implementing Space Vector Modulation with the ADMC33 (AN331-17)” Analog Devices Inc.Jan 2000.
- [79] Dr. KHODJA Djalal Eddine Sigmoid Function Approximation for ANN Implementation in FPGA Devices. 2010

OPTION : INGENIERIE DES SYSTEMES ELECTROMECHANIQUES

Proposé et dirigé par : **Dr. KHODJA Djalal Eddine**

Présenté par : **BEN MAKHLOUFI Amar**

Thème

Implémentation des RNA sur FPGA en vue de commande de la MAS

Résumé :

Dans ce travail nous proposons l'implémentation des réseaux de neurones artificiels sur FPGA pour la commande scalaire du moteur asynchrone en utilisant la Co-simulation. Cette dernière qui fait partie des applications de « Hardware in the loop » permet d'approcher la simulation au cas réel. Les circuits FPGA conviennent parfaitement à une implémentation optimale du système de la commande des MAS, du fait qu'ils ont un coût réduit et qu'ils sont caractérisés par une grande densité d'intégration et une grande flexibilité avec une structure totalement reconfigurable. Par ailleurs, nous procéderons à l'implémentation des RNA sur un circuit de type FPGA 'Virtex4'. L'algorithme d'implémentation proposé sera basé sur la structure simple des RNA. L'implémentation de cet algorithme sera faite à l'aide d'un outil de haut niveau, à savoir : le System Generator de Xilinx.

Mots clés

Commande scalaire. MAS, MLI, RNA, FPGA, System Generator, Xilinx.

N° d'ordre : ISE 188
