

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
ECOLE NATIONALE POLYTECHNIQUE



DÉPARTEMENT D'ÉLECTRONIQUE

Projet de fin d'études

En vue de l'obtention du diplôme d'Ingénieur d'Etat en Electronique

Thème :

**Implémentation d'une commande ANN SHE
PWM sur une carte FPGA pour un véhicule
électrique**

Encadré par :

LARBES Chérif
A.GUELLAL

Réalisé par :

BOUCHEKOUK Oussama
NEKKACHE Abdessalem

Soutenu publiquement le **16 Juin 2015** devant le jury composé de :

M.HADDADI	Professeur ENP	Président
C.LARBES	Professeur ENP	Rapporteur
A.GUELLAL	CDER	Rapporteur
I.SAADAOU	Maitre de conférences	Examineur

ENP 2015

ملخص

الغرض من هذا العمل هو دراسة وإنجاز مغير السرعة للمحرك اللاتزامني على أساس مبدأ V/f ثابت و المراقبة المستمرة للحفاظ على تدفق كهرومغناطيسي ثابت. طريقة التحكم المستعملة هي تقنية تعديل عرض النبضة PWM مع القضاء الانتقائي للاتوافقيات. يتم احتساب زوايا تبديل الإشارات المتحكممة بواسطة تقنية تقوم على الشبكات العصبية الاصطناعية لتنفيذ هذا النظام في الوقت الحقيقي على بطاقة FPGA. بدأنا بالمحاكاة والدراسة الطيفية للإشارات المختلفة على Matlab. بعد ذلك قمنا ببرمجة هذه التقنية على دائرة مبرمجة من نوع XILINX Spartan 3E FPGA باستخدام لغة البرمجة VHDL. أخيراً، قمنا بالتجارب في طوابق مختلفة من هذا المغير و قدمنا تفسيرات للنتائج التي تم الحصول عليها. **كلمات مفتاحية :** تحكم السرعة، محرك لاتزامني، PWM، الشبكات العصبية FPGA، VHDL.

Résumé

Le but de ce travail est l'étude et la réalisation d'un variateur de vitesse pour un moteur asynchrone en se basant sur le principe de la commande scalaire V/f constant afin de garder un flux constant. Ce variateur est piloté par une commande MLI avec élimination sélective des harmoniques. Les angles de commutation des signaux de cette commande seront calculés par une technique basée sur les réseaux de neurones afin de pouvoir implémenter cette commande en temps réel sur une carte FPGA. Nous avons commencé par la simulation et l'étude spectrale des différents signaux sur Matlab. Ensuite, nous avons implémenté cette commande sur une carte FPGA Spartan 3E en utilisant le langage VHDL. Enfin, Nous avons présenté les tests des différents étages de ce variateur et les interprétations des résultats obtenues.

Mots clés : Variateur de vitesse, temps réel, moteur asynchrone, MLI, harmoniques, réseaux de neurones FPGA, VHDL, onduleur triphasé.

Abstract

The purpose of this final project is the study and the realizing of a variable speed drive for asynchronous motor based on the scalar V/f control. This drive is controlled by a PWM control with selective elimination of harmonics. The switching angles of the control signals will be calculated by a technique based on artificial neural networks to implement this command in real time "On line" on a FPGA board. We started with simulation and spectral analyzing of the different signals on Matlab. Then we implemented this command on a "Spartan 3E" FPGA using VHDL. Finally, we presented the tests on different floors of this drive and interpretations of the results obtained.

Keywords: Speed controlle, on line, asynchronous motor, PWM, harmonics, neural networks, FPGA, VHDL, three-phase inverter.

Dédicace

Je dédie ce travail à :

Mes chers parents

Mes frères et ma sœur

Toute ma famille

Ainsi qu'à tous mes amis.

Oussama

Je dédie ce mémoire :

À mes très chers parents qui m'ont toujours encouragé durant mes années d'étude. Je vous souhaite ! Longue vie cher père chère mère.

À mes frères Abderrahim, Younes, Tarek et mon adorable sœur, auxquels je souhaite un meilleur avenir.

À mes grands-parents, À mes tantes et oncles et toute ma famille de proche ou de loin.

À mes amis et à toutes les personnes qui ont une place spéciale dans ma vie.

Abdesslem

Remerciement

Nous tiendrons à remercier dieu de nous avoir donné la patience de terminer ce travail.

Nos remerciements les plus particuliers vont à notre promoteur monsieur C. LARBES pour son suivi, son aide et son soutien matériel, ses précieux conseils et sa disponibilité.

Nos remerciements vont également à monsieur A.GUELLAL pour son aide précieuse, son suivi en programmation et sa disponibilité.

Nous adressons nos sincères remerciements les membres de jury :

Mr : M. HADDADI, Mr : I. SAADAOUI d'avoir accepté de juger ce travail.

Nous voudrions exprimer nos profonds respects à tous les Enseignants qui nous ont encadrés durant notre étude.

Liste des figures :

Figure 1.1 : Composantes d'un véhicule électrique.....	3
Figure 1.2 : Véhicule tout électrique.....	4
Figure 1.3 : Architecture du véhicule hybride série.....	5
Figure 1.4 : Architecture du véhicule hybride parallèle.....	5
Figure 1.5 : Architecture du véhicule hybride mixte.....	6
Figure 1.6: Création d'un courant induit dans un conducteur fermé.....	8
Figure 1.7 : Positionnement des enroulements dans les machines asynchrones triphasées.....	9
Figure 1.8 : Eclaté d'un moteur asynchrone.....	10
Figure 1.9: Le couple électromagnétique en fonction de glissement pour une fréquence fixe.	13
Figure 1.10 : Caractéristique couple vitesse en fonction de la fréquence d'alimentation.....	14
Figure 1.11 : Un système de commande V/f constant pour une machine asynchrone.....	14
Figure 1.12 : Schéma de l'onduleur monophasé.....	17
Figure 1.13 : Schéma de principe d'un onduleur triphasé de tension.....	17
Figure 1.14 : Schéma synoptique de la loi de modulation triangulo-sinusoidale.....	18
Figure 2.1: Un signal MLI à deux niveaux avec une symétrie demi onde.....	21
Figure 2.2: Un neurone avec son arborisation dendritique.....	30
Figure 2.3: Structure d'un nœud de RNA.....	31
Figure 2.4 : Topologie d'un réseau multicouche (MLP).....	32
Figure 2.5 : Représentation de la couche de sortie d'un réseau de neurones.....	34
Figure 2.6 : Centrage de la base de données.....	36
Figure 2.7 : Architecture de réseau de neurones pour l'MLI.....	37
Figure 2.8 : Utilisation de la fonction "nn_train_tool" pour le calcul des paramètres de réseau ANN-1.....	39
Figure 2.9 : Signal MLI pour $i_m=0.6$ et $m=7$	42
Figure 2.10 : Le spectre de signal MLI pour $i_m=0.6$ et $m=7$	43
Figure 3.1 : Classification des circuits numériques.....	46
Figure 3.2 : Architecture interne du FPGA.....	48
Figure 3.3 : Une cellule SRAM.....	49
Figure 3.4 : Différentes utilisation de cellules SRAM.....	49

Figure 3.5 : Exemple de LUT.....	50
Figure 3.6 : Structure de base d'un composant programmable.....	51
Figure 3.7 : Structure d'un bloc d'entrée/sortie.....	51
Figure 3.8 : Schéma montrant structure d'un code VHDL.....	54
Figure 3.9 : Organisation fonctionnelle de développement d'un projet sur circuit FPGA.....	57
Figure 3.10 : Quelques schémas électroniques équivalents générés par le synthétiseur, visualisés par l'outil «RTL Schematic Viewer».....	59
Figure 3.11 : Organigramme de l'algorithme de commande.....	60
Figure 3.12 : Résultat de simulation des signaux des sorties pour les trois phases.....	61
Figure 3.13: Visualisation des signaux ANN SHE PWM.....	62
Figure 4.1 : Le schéma bloc de variateur de vitesse.....	64
Figure 4.1 : Structure de MOSFET.....	67
Figure 4.3 : Transistor High-side et Low-side du bras de l'onduleur.....	68
Figure 4.4 : Principe de la technique de bootstrap.....	69
Figure 4.5: Caractéristique $R_{th(j-c)}$ en fonction du rapport cyclique D et de temps de conduction du MOSFET.....	71
Figure 4.6 : Le Schéma et les pins de HCPL220.....	72
Figure 4.7 : Le montage de l'HCPL 2200.....	73
Figure 4.8 : Le circuit d'isolation.....	73
Figure 4.9 : Schéma interne et les pins d'IR2111.....	74
Figure 4.10: Utilisation typique d'IR2111.....	75
Figure 4.11: Dimensionnement de capacité de bootstrap.....	76
Figure 4.12 : Blocs de commande approchée pour les trois phases.....	77
Figure 4.13 : Le signal au sortie de l'étage de commande et son spectre.....	78
Figure 4.14 : La tension entre une phase et la masse.....	79
Figure 4.15 : Tension ente deux phases.....	80

Liste des tableaux :

Tableau 2.1 : Les angles de commutation en fonction de im.....	29
Tableau 2.2 : Les nombres des angles en fonction d'indice de modulation im.....	38
Tableau 2.3 : Paramètre de réseau ANN-1 obtenus.....	39
Tableau 2.4 : Les angles de commutations calculées par le réseau ANN-1.....	40
Tableau 2.5 : L'erreur entre les valeurs calculées par ANN-1 et les valeurs exactes.....	41

Abréviations :

MLI : Modulation à Largeur d'Impulsion

MCC : Moteur à Courant Continu

MS : Moteur Synchrone

MAS : Moteur Asynchrone

DSP : Digital Signal Processor

MLI ESH : Modulation à Largeur d'Impulsion avec Elimination Sélective d'Harmonique

SHE PWM: Selective Harmonics Elimination Pulse –Width Modulation

TMG : Taufik, Mellitt et Goodmanla

RNA : Réseau de Neurones Artificiel

ANN: Artificial Neural Network

LSI: Large Scale Integration

SSI: Small Scale Integration

VLSI: Very Large Scale Integration

ABEL: Advanced Boolean Expression Language

CLB: Configurable Logic Bloc

EEPLD: Electrically EPLD

EEPROM: Electrically EPROM

EPLD : Erasable PLD

EPROM: Erasable PROM

FPGA : Field Programmable Gate Arrays

IEEE: Institute of Electrical and Electronics Engineers

ISE: International Society of Electrochemistry

LCA: Logic Cells Arrays

IOB: input output bloc

LS I: Large Scale Integration

LUT: Look Up Table

MOS: Metal Oxyde Semiconductor

MSI: Medium Scale Integration

Mux : Multiplexer

PAL : Programmable Array Logic

PLA: Programmable Logic Array

PLD: Programmable Logic Device

ROM: Read Only Memory

RTL :Register Transfer Level

SRAM: Static Random Access Memory

SSI: Small Scale Integration

VHDL: VHSIC Hardware Description Language

VHSIC: Very High Speed Integrated Circuit

VLSI : Very Large Scale Integration

MOSFET: Metal–Oxide–Semiconductor Field-Effect Transistor

IGBT: Insulated-Gate Bipolar Transistor

Nomenclature:

Symbole	Correspondance	Unité
N	Vitesse de champ tournant	tr/min
N_s	Vitesse de synchronisme.	tr/min
f_0	Fréquence de courant d'alimentation.	Hz
p	Nombre de pair de pôles	//
g	Glissement	//
C	Couple moteur	N.m
I_s	Courant du stator	A
I_r	Courant du rotor	A
R_s	Résistance du stator	Ω
R_r	Résistance du rotor	Ω
N_r	Inductance de fuite rotorique	H
ω	Pulsation du stator	rad/s
M'	Inductance mutuelle entre le stator et le rotor	H
L	Inductance propre d'une phase statorique	H
Φ_{rd}	Flux rotorique	weber
Φ_{rq}	Flux statorique	weber
I_{rq}	Courant induit rotorique dans l'axe q	A
I_{sd}	Courant statorique dans l'axe d	A
α	Angle de commutation	$^\circ$
i_m	Indice de modulation	//
k	Indice de kieme itération	//
M	Nombre des angles de comutation	//
α^*	Angle de commutation exacte	$^\circ$

$\alpha^{(k)}$	Angle de commutation calculé à la kieme itération	°
$\Delta\alpha^{(k)}$	Erreur entre l'angle de commutation exact et calculé à la kieme itération	°
$E^{(k)}$	Matrice des dérivées premières calculée à la kime itération	//
D	Ensemble de base de données	//
W	Matrice des poids	//
x	Entrée de RNA	//
y^{des}	Sortie désirée	//
J	Gradient	//
y	Sortie calculée	//
N_2	Nombre d'exemple dans la base de données	//
a	Sortie de nœud	//
b	Seuil	//
W_1	Matrice des poids entre la couche d'entrée et la couche cachée	//
W_2	Matrice des poids entre la couche cachée et la couche de sortie	//
b_1	Matrice des poids de la couche cachée.	//
b_2	Matrice des poids de la couche de sortie	//
f_c	Fréquence de commutation	Hz
T_c	Période de commutation	s
P_D	Puissance dissipée	W
$P_{D_conduction}$	Puissance dissipée par conduction	W
P_{D_commu}	Puissance dissipée par commutation	W
R_{DSS}	Résistance Drain-Source en régime statique.	Ω
I_{DSS}	Courant drain-source	A
D	Rapport cyclique de commutation	//
C_{IN}	Capacité d'entrée du MOSFET	F

$R_{th(j-c)}$	Résistance thermique entre la jonction et le boîtier	W/°C
$R_{th(c-r)}$	Résistance thermique entre le boîtier et le radiateur	W/°C
$R_{th(r-a)}$	Résistance thermique radiateur-ambiante	W/°C
T_j	Température de jonction	°C
T_a	Température ambiante	°C
R_{th}	Résistance thermique totale	W/°C
V_{cc}	Tension d'alimentation de la carte de commande	V
V_F	Tension aux bornes de la diode de bootstrap	V
V_{GSmin}	Tension grille source d'amorçage minimale	V
V_{DSon}	Tension drain- source en conduction de MOSFET	V
Q_{TOT}	Charge totale de condensateur de bootstrap	C
Q_G	Charge nécessaire à la grille du MOSFET	C
I_{fuite}	Courant de fuite de driver	A
t_{off}	Temps de blocage du MOSFET	S
C_b	Capacité de condensateur de bootstrap	F

// Sans unité

Table des matières

Introduction générale.....	1
Chapitre I : Les véhicules électriques, les moteurs asynchrones et onduleurs de tension	
1.1 Introduction :.....	3
I.2. Le véhicule électrique	3
I.2.1. Principales Configurations des Véhicules	3
I.2.1.1. Le véhicule tout électrique	4
I.2.1.2. Les véhicules hybrides	4
I.2.2. Choix du moteur électrique	6
I.3. Les moteurs asynchrones	7
I.3.1. Introduction	7
I.3.2. Le principe de fonctionnement des moteurs asynchrones	8
I.3.2.1. Le champ tournant	8
I.3.2.2. La vitesse de synchronisme	9
I.3.2.3. Le Glissement	9
I.3.3. Principaux composants de la machine asynchrone	10
I.3.4. Commandes des machines asynchrones	11
I.3.4.1. La commande scalaire V/f constant	11
I.3.4.2. La commande vectorielle	15
I.4. Les Onduleurs	16
I.4.1. Notions sur les onduleurs	16
I.4.2. Classification des onduleurs	16
I.4.2.1 Onduleurs non autonome	16
I.4.2.2 Onduleur autonome	16
I.4.3. Types d'onduleurs autonomes de tension	16
I.4.3.1 Les onduleurs monophasés	16
I.4.3.2 Les onduleurs triphasés.....	17
I.4.4. Commandes d'onduleurs.....	17
I.4.4.1. Types de commandes	17
I.4.4.2. Différentes techniques MLI.....	18
I.5. Conclusion	19

Chapitre II : Algorithme MLI avec élimination des harmoniques et réseau de neurones	
II.1 Introduction	20
II.2. La méthode MLI ESH (Patel-Hoft)	21
II.2.1. Principe de Patel-Hoft	21
II.2.2. Calcul de valeurs exactes des angles de commutation par la Méthode de Newton-Raphson	25
II.2.2.1. Algorithme TMG	26
II.2.2.2. Algorithme de calcul des angles	26
II.2.3. calcul des angles de commutation	28
II.2.4. Interprétation des résultats	29
II.3. Réseau de neurones artificiels.....	30
II.3.1. Généralité sur les réseaux de neurones artificiels	30
II.3.1.1. Définitions	30
II.3.1.2. Modélisation mathématique	31
II.3.2. Méthode de gradient	33
II.3.3. Centrage de données	36
II.4. Une commande MLI à base des réseaux de neurones	36
II.4.1. Architecture de l'RNA	37
II.4.2. Calcul des paramètres des réseaux	38
II.4.3. Calcul des angles de commutation	40
II.4.4. Interprétation des résultats	40
II.4.5. Simulation	42
II.5. Conclusion	44
Chapitre III : Implémentation de la commande ANN_SHE_PWM sur une carte FPGA	
III.1. Introduction.....	45
III.2. Les circuits logiques programmables PLD	46
III.3. Les circuits FPGA.....	47
III.3.1. Définition	47
III.3.2. Application des FPGA	47
III.3.3. Architecture des FPGA	47
III.3.4. Réseau mémoire SRAM	49
III.3.5. Les CLB	50
III.3.6. Les IOB	51
III.3.7. Les interconnexions	52

III.3.8. Les outils de développement des FPGA	52
III.4. Langage de description VHDL.....	53
III.4.1. Définition	53
III.4.2. Pour quoi un langage de description	53
III.4.3. Spécification	53
III.4.4. Simulation	53
III.4.5. Conception	54
III.4.6. Structure d'une description VHDL simple	54
III.4.6.1 Déclaration des bibliothèques	55
III.4.6.2 Déclaration de l'entité et des entrées/sorties	55
III.4.7. Les deux modes de travail en VHDL	55
III.4.7.1. Le mode combinatoire	55
III.4.7.2. Le mode séquentiel	56
III.5. Etapes nécessaires au développement d'un projet sur FPGA	56
III.5.1. Saisie du texte VHDL	57
III.5.2. Vérification des erreurs.....	57
III.5.3. Synthèse.....	57
III.5.4. Simulation	59
III.5.5. Optimisation, placement et routage	59
III.5.6. Programmation du composant et test	60
III.6. Implémentation de l'algorithme de commande sur FPGA	60
III.6.1. Description de l'algorithme	60
III.6.2. Simulation.....	61
III.6.3. Test de signaux de sortie	61
III.6.4. Interprétation	62
III.7. Conclusion	63
Chapitre IV : Réalisation d'un variateur de vitesse piloté par une commande ML	
IV.1. Introduction	64
IV.2. Principaux blocs du système	64
IV.2.1. L'étage de puissance	65
IV.2.1.1. Choix de type de transistor	66
IV.2.1.2. Les caractéristiques générales des MOSFET.....	67
IV.2.1.3. Conception de l'étage de puissance	70
IV.2.2. L'étage de commande	72
IV.2.2.1. Le bloc d'isolation	72

IV.2.2.2. Le bloc de commande approchée.....	74
IV.3. Résultats expérimentaux	78
IV.3.1. Les signaux aux sorties de l'étage de commande	78
IV.3.2. Les signaux aux phases de l'onduleur	79
IV.4. Conclusion.....	81
Conclusion Générale.....	82
Bibliographie	
Annexe	

Introduction Générale

Les constructeurs automobile considèrent que le véhicule électrique est devenu une solution inévitable pour le futur, pour plusieurs raisons parmi lesquelles la minimisation des émissions des gaz à effet de serre ainsi que la réduction de consommation des ressources énergétiques non renouvelables. Cependant la mise en œuvre des véhicules électriques capables de concurrencer les véhicules à moteur à combustion avec les performances croissantes de ces derniers n'est pas du tout évidente. En effet, les véhicules électriques existants présentent de nombreux inconvénients tels que la faible autonomie de batterie, de faibles performances, une longue durée de charge, l'indisponibilité des stations de charge et beaucoup d'autres points faibles.

La commande intelligente associée avec une gestion optimale de l'énergie électrique dans le véhicule électrique peuvent améliorer les performances. Dans ce contexte notre projet de fin d'étude introduit une commande basée sur les réseaux de neurones pour la commande du moteur asynchrones d'un véhicule électrique.

Dans ce projet nous considérons la partie commande du moteur électrique. Le moteur le plus adapté aux véhicules électriques est le moteur asynchrone à cause de sa facilité de construction et sa maintenance, son coût minimal et son rendement élevé. Tandis que la variation de vitesse nécessite une source d'alimentation alternative de fréquence et amplitude variable. Nous avons choisi la commande scalaire V/f constant qui implique un flux constant. Cette alimentation est assurée par un onduleur de tension piloté par une commande MLI (PWM). La tension de sortie de ce dernier est riche en harmoniques, d'où le fonctionnement de moteur sera accompagné des pulsations de couple et d'échauffement de celui-ci à cause des pertes. Ces pertes influent directement sur l'autonomie du véhicule et les pulsations de couple causent la défaillance du moteur à long terme. La commande MLI avec élimination sélective d'harmoniques (SHE PWM) permet d'éliminer à volonté les harmoniques proche au fondamentale en se basant sur le principe de Patel et Hoft. Les angles de commutation seront calculés en se basant sur le principe des réseaux de neurones (ANN).

L'objectif du ce projet de fin d'études peut être résumée comme suite :

- Implémenter la commande ANN SHE PWM dans une carte FPGA.
- Réaliser un variateur de vitesse pour un moteur asynchrone piloté par cette commande.

Afin de réaliser ces objectifs. Nous avons organisés ce mémoire en quatre chapitres :

Le premier chapitre donne une présentation des généralisées sur le véhicule électrique, les moteurs asynchrones et les onduleurs de tension ainsi que les commandes des onduleurs.

Le deuxième chapitre est consacré pour la commande MLI basé sur le principe de Patel-Hoft, en prenant deux méthodes différentes pour calculer les angles de commutation du signal MLI. La première c'est la méthode itérative de Newton-Raphson et la deuxième s'agit d'un nouvel algorithme basé sur les réseaux de neurones (ANN SHE PWM).

Le Troisième chapitre comporte les connaissances et les étapes suivis pour l'implémentation de la commande ANN SHE PWM sur une carte FPGA. Les signaux générés par la carte FPGA sont accompagnés d'une analyse spectrale pour pouvoir confirmer l'élimination des harmoniques non désirés.

Le quatrième chapitre regroupe l'étude et les étapes suivis pour réaliser un variateur de vitesse piloté par les signaux de commande générés par la carte FPGA. Les signaux de sorties de variateur sont également analysés dans l'espace fréquentiel.

Chapitre I

LES VEHICULES ELECTRIQUES, LES MOTEURS ASYNCHRONES ET LES ONDULEURS DE TENSION

1.1 Introduction :

Dans ce chapitre, on va présenter quelques généralités sur le véhicule électrique et les moteurs asynchrones, ensuite on va voir les types d'onduleurs qui existent et la technique MLI consacrée pour leurs commandes.

1.2. Le véhicule électrique :

Le véhicule électrique se compose de deux éléments essentiels : une batterie et un moteur électrique. La batterie est l'élément de stockage de l'énergie. Elle fournit l'électricité nécessaire au fonctionnement du moteur. Elle est reliée au moteur électrique par un convertisseur commandé de façon à régler l'intensité du courant qui actionne le moteur. Le convertisseur utilisé dépend du type du moteur utilisé, qui peut être à courant continu ou alternatif.

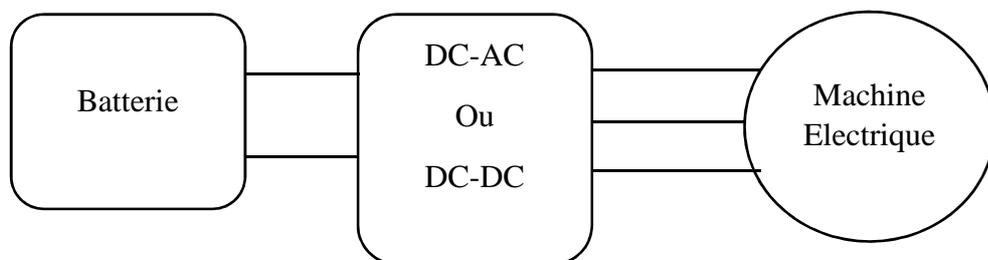


Figure 1.1 : Composantes d'un véhicule électrique.

1.2.1. Principales Configurations des Véhicules :

Selon le type d'énergie embarquée on distingue deux grandes familles de véhicules électriques : le véhicule tout électrique et le véhicule hybride [1].

I.2.1.1. Le véhicule tout électrique :

Il s'agit d'un véhicule qui possède uniquement un accumulateur comme source d'énergie. La structure est donnée par le schéma de la figure 1.2.

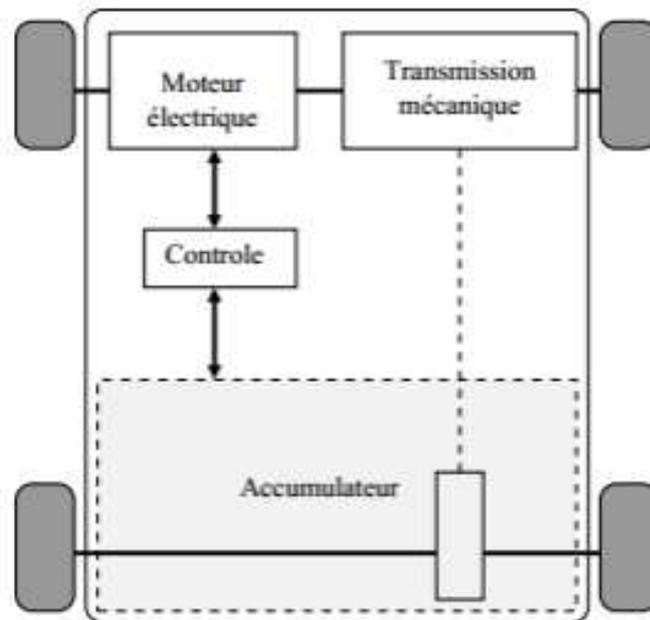


Figure 1.2 : véhicule tout électrique

Les véhicules tout électriques proposés actuellement sont exclusivement urbains, leur autonomie est comprise entre 70 et 120 km avec des technologies de batteries relativement conventionnelles (plomb-acide et cadmium-nikel) et 150 à 200 km avec des technologies plus avancées (nikel-métal-hydrure et lithium). Le freinage récupératif permet d'accroître sensiblement l'autonomie (d'environ 20%), surtout en cycle urbain. Il permet en outre d'obtenir un frein moteur. Pour ces raisons, il est nécessaire d'avoir un système de conversion réversible [1].

I.2.1.2. Les véhicules hybrides :

Les véhicules hybrides utilisent des sources d'énergie différentes pour fournir la force motrice. Généralement, ils combinent un moteur à combustion interne et une ou plusieurs machines électriques (moteur/générateur) associées à un système de stockage. En comparant avec un véhicule électrique de même dimension et de même puissance, un véhicule hybride est plus spacieux, plus léger et offre plus d'autonomie. Par rapport aux véhicules conventionnels, les véhicules hybrides consomment moins de carburant et émettent donc moins de polluants [2].

On distingue alors trois concepts de véhicule hybride :

a. Hybride série

C'est la configuration la plus simple. L'énergie mécanique fournie par le moteur à combustion est convertie en énergie électrique par un alternateur pour alimenter le moteur électrique ou recharger la batterie [3].

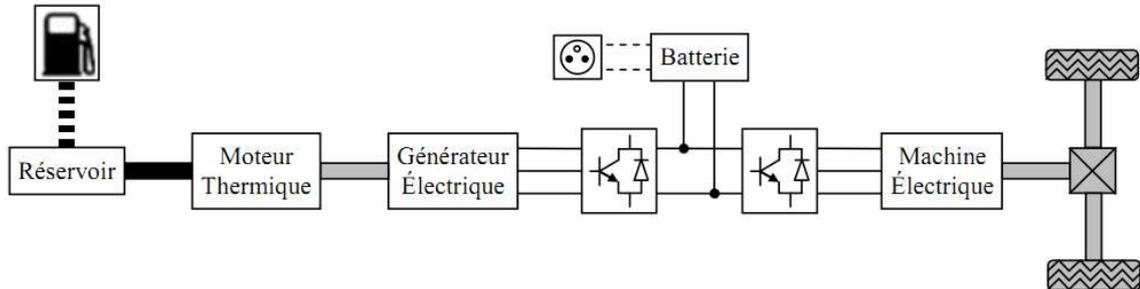


Figure 1.3 : architecture du véhicule hybride série.

Pour ce moteur à essence, le meilleur rendement se situe à l'intérieur de l'iso consommation de 250 g/kWh (34,3 %)

b. Hybride parallèle

Dans cette configuration, le moteur à combustion et le moteur électrique sont mécaniquement couplés au système de transmission, de sorte qu'ils peuvent travailler séparément. [3]

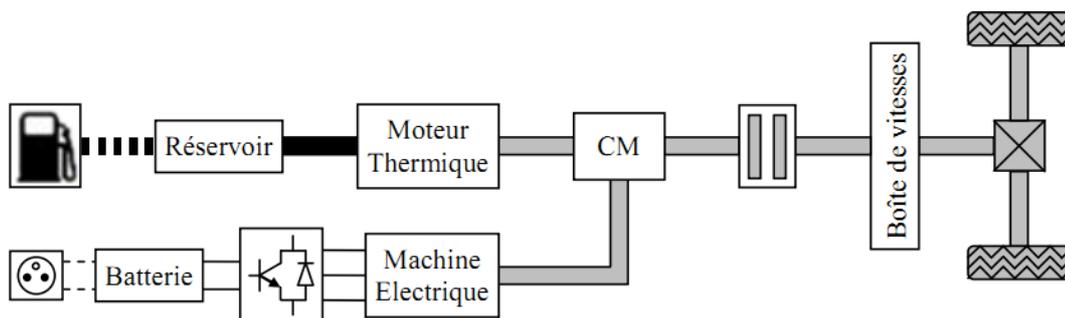


Figure 1.4 : architecture du véhicule hybride parallèle.

c. Hybride mixte (série-parallèle)

Cette structure est basée sur la combinaison d'une structure série et d'une structure parallèle, le but est de maximiser les avantages des deux structure [3].

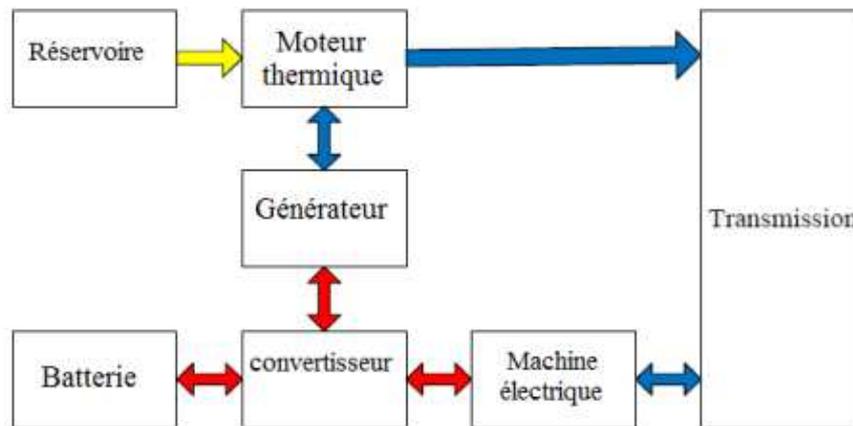


Figure 1.5 : architecture du véhicule hybride mixte.

I.2.2. Choix du moteur électrique [4]

Le choix du moteur électrique repose sur quelques critères : cout, performance, rendement, poids, facilités de maintenance. La relation est étroite entre l'augmentation des moteurs et la réalisation des hautes performances des véhicules électriques en exploitation.

Il existe trois types de moteurs à ce stade (le moteur à courant continu MCC, le moteur synchrone MS et le moteur asynchrone MAS). La solution de référence a été pendant longtemps la motorisation à courant continu. L'avantage principal des MCC réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation : les variateurs de vitesse, voire leur raccordement direct à la source d'énergie : batteries d'accumulateur, piles, etc...

En ce qui a trait aux inconvénients,

- Maintenance : les contacts glissants (collecteurs-balais) s'usent (frottement mécanique-arcs électrique) ce qui nécessite un entretien régulier (changement des contact bagues balais)
- Cout : environ 3 fois le prix d'un moteur asynchrone
- rendement limité (90%).
- pertes thermiques situées au rotor difficiles à évacuer.

Les problèmes cités précédemment et l'évolution de l'électronique de puissance et des matériaux tels que les aimants permanents ont conduit aujourd'hui à s'orienter vers des solutions plus performantes telles que les motorisations synchrones ou asynchrones.

Les MS ont effectivement des avantages non négligeables :

- Des rapports couple/masse et puissance/masse élevés.
- Un très bon rendement.
- Moins d'usure et moins de frais d'entretien (pas de balais, ni charbons).
- la vitesse du moteur est constante quelle que soit la charge (intéressant dans le cas des ascenseurs).
- Ils peuvent supporter des chutes de tension importantes sans décrocher.

Cependant ils ont tout de même certains inconvénients :

- Coût élevé (à cause du prix des aimants).
- Problème de tenue en température des aimants (250 °C pour le samarium-cobalt)
- Impossibilité de régler l'excitation.
- Pour atteindre des vitesses élevées, il est nécessaire d'augmenter le courant statorique afin de démagnétiser la machine. Ceci entraînera inévitablement une augmentation des pertes statoriques par effet Joule.
- La non-régulation de ce flux ne permet pas une souplesse de contrôle sur une très large plage de vitesse.

Le moteur asynchrone est le candidat le plus adapté pour propulser les véhicules hybrides électriques.

I.3. Les moteurs asynchrones :

I.3.1. Introduction:

Les moteurs asynchrones comme leurs nom indique, fonctionnent contrairement aux machines synchrones, donc une différence de phase existe entre la vitesse du champ tournant créé par le stator et la vitesse de rotor.

Ces moteurs sont les plus utilisées dans les milieux industriels et même dans les centrales électriques. Ceci est due à la stabilité de sa vitesse, son bon rendement et de son excellente

fiabilité. De plus ces machines offrent une grande simplicité de construction, une simplicité du démarrage, un prix assez bas et la robustesse des pièces de la machine.

Le major inconvénient de la machine asynchrone s'agit de sa consommation relativement élevée de la puissance réactive due à la présence de l'entrefer. Si on veut minimiser la consommation de la puissance réactive par la réduction de quantité de l'entrefer. Ces machines présentent une difficulté réside à la variation de vitesse qui dépend de fréquence d'alimentation de stator, il est claire qu'on a besoin d'avoir une alimentation alternative à fréquence variable à partir d'une alimentation continue qui est assurée par un onduleur. On va présenter dans les titres suivants ces onduleurs et comment les commander.

Les moteurs asynchrones avec tous ces avantages représentent presque 80% des machines existantes dans l'industrie [5].

I.3.2. Le principe de fonctionnement des moteurs asynchrones :

Le principe de fonctionnement des moteurs asynchrone se base sur la création d'un courant induit sur un conducteur fermé lorsque ce dernier est engendré par un champ magnétique variable et tournant. La combinaison entre le courant induit et les lignes de champ donne naissance à une force motrice appelée force de Lenz.

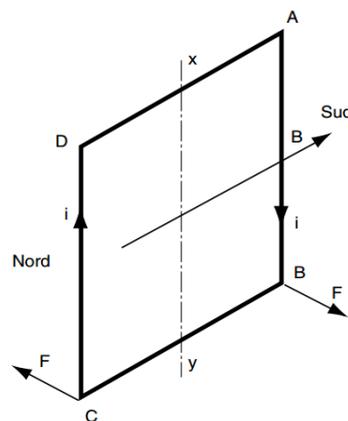


Figure 1.6: création d'un courant induit dans un conducteur fermé.

I.3.2.1. Le champ tournant :

Le champ tournant est créé à l'aide de trois enroulements radiaux (Figure 1.7) décalés de 120° . Ces trois enroulements sont parcourus par des courants alternatifs de la même amplitude et décalé de la même phase 120° . Le champ résultant est la somme vectoriel des trois champs créés par chaque bobine. Il sera donc tournant sortant de centre de cercle délimité par les trois bobines.

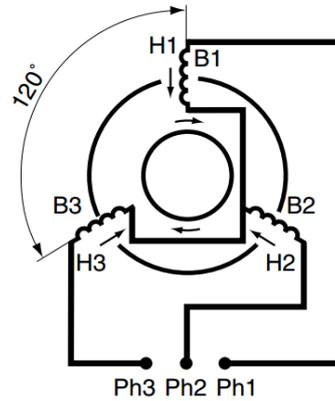


Figure 1.7 : positionnement des enroulements dans les machines asynchrones triphasées.

I.3.2.2. La vitesse de synchronisme :

La vitesse de synchronisme est définie par la vitesse de rotation de champ tournant. Elle est proportionnelle à la fréquence de courant d'alimentation et inversement proportionnelle au nombre de pair de pôle. La vitesse de synchronisme est donnée par la relation:

$$N_s = \frac{2\pi \cdot f_0}{p} \quad (1.1)$$

Où :

N_s : la vitesse de synchronisme.

f_0 : la fréquence de courant d'alimentation.

p : le nombre de pair de pôle.

I.3.2.3. Le Glissement:

On appelle un glissement la différence entre la vitesse de synchronisme (N_s) et celle de la spire (N). Il s'exprime en % et il est donné par :

$$g = 100 \cdot \frac{N_s - N}{N_s} \quad (1.2)$$

Il est notable d'après l'équation (1.1) que si on veut augmenter la vitesse de rotation de moteur on augmente la fréquence de courant d'alimentation. Mais dans la pratique ce n'est pas toujours le cas car il faut tenir compte si les conceptions mécaniques et électriques le permettent. Par exemple, la majorité des moteurs qu'on trouve dans l'industrie fonctionnent dans des fréquences nominales bien données, généralement 50~60 ou 100 Hz. Dans la commande que nous allons implémenter on va utiliser le principe de V/f constant. Ou V est la tension appliquée aux bornes de chaque enroulement.

I.3.3. Principaux composants de la machine asynchrone :

Les machines asynchrones se composent de deux principaux éléments appelés le stator et le rotor (Figure 1.8).

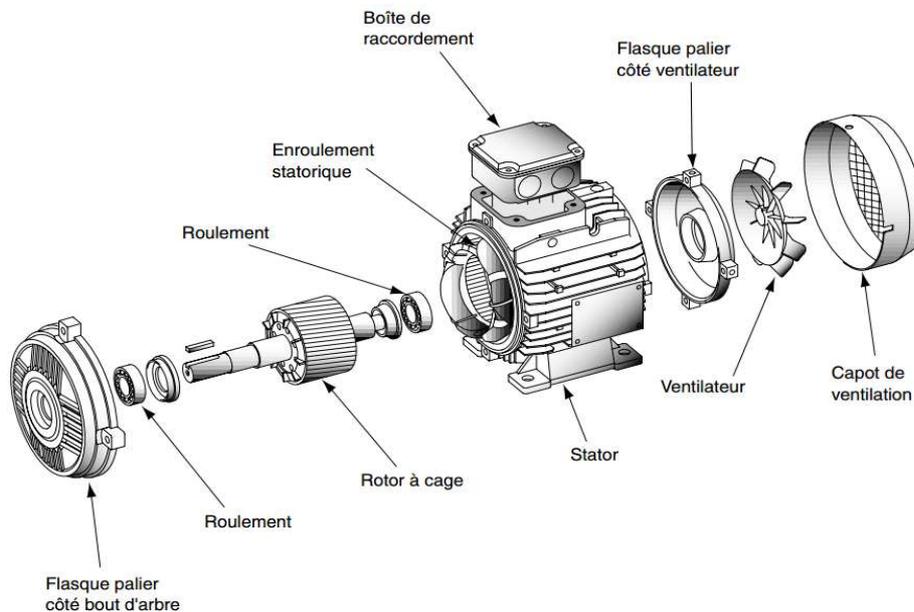


Figure 1.8 : Eclaté d'un moteur asynchrone.

Le stator c'est la seule partie alimentée dans les moteurs asynchrones. Il est responsable de générer le champ tournant. Il est de forme d'un cylindre creux contient les trois enroulements triphasés répartie régulièrement dans des encoches (Figure I.8). Il est aussi équipé d'une multitude des tôles feuilletées construisant un radiateur.

Le rotor c'est l'élément tournant dans le moteur. Il est constitué d'un empilement des tôles minces isolées entre elle répartie sur l'axe de rotation forment un cylindre claveté sur l'arbre du moteur.

Il existe deux types de rotor :

- Rotor à cage Dans ce cas, le rotor est constitué de barres conductrices régulièrement réparties entre deux couronnes métalliques formant les extrémités.
- Rotor à bague (bobiné) Dans le cas du moteur à rotor bobiné, on peut avoir un accès électrique au rotor à partir d'un ensemble de trois bagues ou cylindres métalliques disposés sur l'arbre du rotor [5].

I.3.4. Commandes des machines asynchrones :

La commande des moteur consiste à faire varier certains paramètre d'entrées tel que la tension, la fréquence et le courant à fin d'agir sur le couple moteur pour varie la vitesse, le sens de rotation et la position angulaire. On peut classer les commandes existant en deux types : commande scalaire et commande vectorielle.

- La commande scalaire : basée sur le modèle en régime permanent, elle est simple à implanter avec une dynamique lente. Elle contrôle les grandeurs en amplitude.
- La commande vectorielle : basée sur le modèle transitoire, précise et rapide elle permet le contrôle du couple. Elle est chère car nécessitant souvent des capteurs, encodeur incrémental ou estimateur de vitesse, DSP... Elle contrôle les grandeurs en amplitude et en phase [6].

I.3.4.1. La commande scalaire V/f constant :

La commande scalaire se base sur la modélisation de la machine en équations simplifiés en ne tenant compte que des amplitudes et en prenant les déphasages entre les phases constantes par rapport au temps en régime permanant sinusoïdale.

Plusieurs commandes scalaires existent selon que l'on agit sur le courant ou sur la tension. Le principe est d'avoir un bon couple électromagnétique pour un fonctionnement donné.

Ce contrôle revient à maintenir le flux constant. Cette hypothèse nous ramène à maintenir le rapport V/f constant [7].

En effet, si on suppose que les tensions aux bornes de chaque phase sont équilibrées.

Le couple moteur s'écrit [8] :

$$C = \frac{P}{\Omega} = 3 \frac{p}{g} I_r^2 \quad (1.3)$$

Avec

$$I_r = \frac{V_s - R_s I_s}{\frac{R_r}{g} + jN_r \Omega} \quad (1.4)$$

Où P c'est la puissance mécanique, Ω la vitesse de rotation. I_s , I_r , R_s , R_r , Les courants et les résistances de stator et de rotor respectivement et N_r l'inductance de fuite rotorique et ω la pulsation de stator.

Si on néglige la résistance statorique et on remplace (1.4) dans (1.3) on obtient [8]:

$$C = \frac{3p}{\Omega} V_s^2 \frac{\frac{R_r}{g}}{\left(\frac{R_r}{g}\right)^2 + (N_r \Omega)^2} \quad (1.5)$$

Pour des faibles glissements on aura :

$$C = \frac{3p}{\Omega} V_s^2 \frac{g}{R_r} \quad (1.6)$$

On voit bien d'après la relation (1.6) que le couple varie quasi linéairement en fonction de glissement (Figure 1.9).

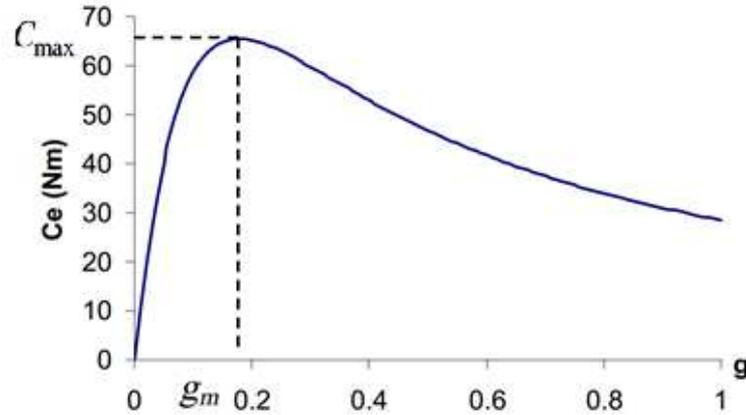


Figure 1.9: Le couple électromagnétique en fonction de glissement pour une fréquence fixe.

De plus, le couple moteur est maximum pour une valeur de glissement $g_m = \frac{R_r}{N_r \Omega}$

On a alors :

$$C_{\max} = \frac{3p}{2N_r} \frac{V_s^2}{\Omega} \quad (1.7)$$

La relation (1.7) montre bien que le couple maximum est proportionnel au carré du rapport de tension sur la pulsation de courant statorique. Pour garder le couple maximum constant il faut maintenir ce rapport constant lors de variation de la fréquence de courant d'alimentation et la tension. Cela fait varier la vitesse de rotation tant que le couple moteur et le couple résistant ne sont pas égaux (Figure 1.10).

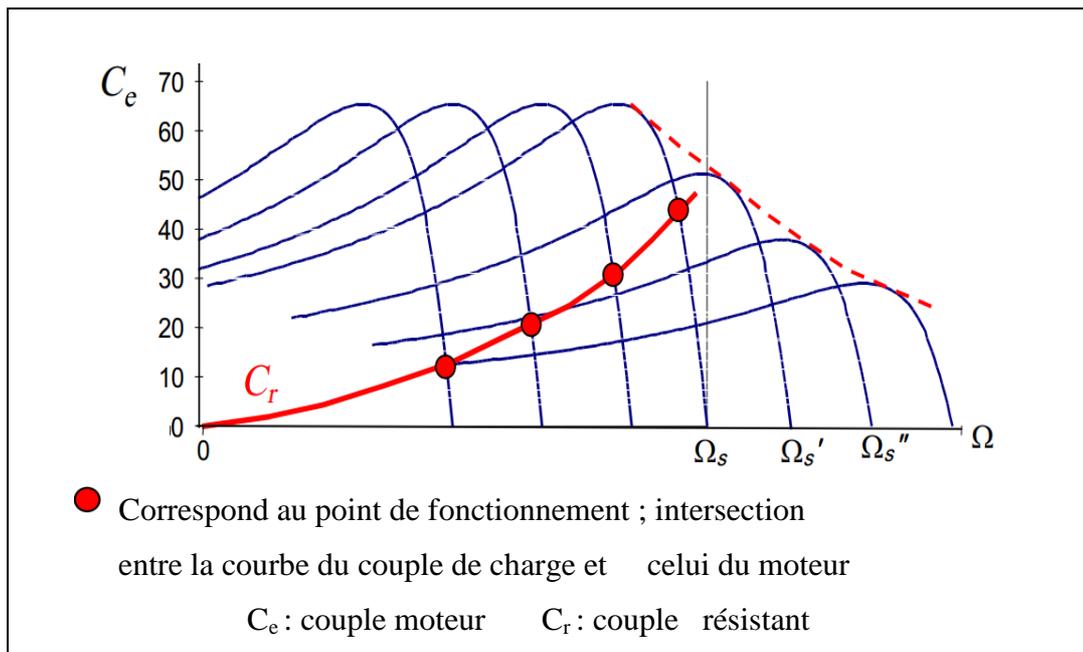


Figure 1.10 : Caractéristique couple vitesse en fonction de la fréquence d'alimentation.

En fait, garder le rapport constant revient à garder le flux constant. Quand la tension atteint sa valeur maximale, on commence alors à décroître ce rapport ce qui provoque une diminution du couple que peut produire la machine. On est en régime de "défluxage". Ce régime permet de dépasser la vitesse nominale de la machine, on l'appelle donc aussi régime de survitesse [7].

Un système de commande scalaire V/f constant d'une machines asynchrone doit contenir un module pour le contrôle de ce rapport ainsi qu'un actionneur qui génère les ondes de tension pour les trois phases. Cette topologie est représentée dans la figure 1.11.

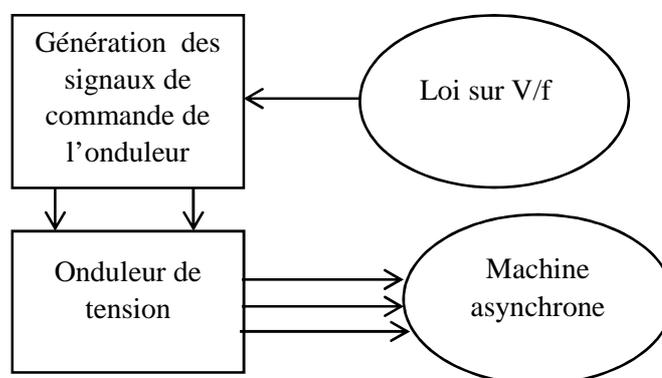


Figure 1.11 : Un système de commande V/f constant pour une machine asynchrone.

I.3.4.2. La commande vectorielle :

Comme on a vu dans la méthode de commande scalaire (Eq 1.5) nous n'avons pas montré le couplage entre le flux et le couple. Nous avons juste définie le couple en fonction de glissement. La méthode vectorielle, introduite au milieu des années 1980, se base sur le contrôle de couple par l'orientation de flux.

Le couple électromagnétique des machines à induction résulte de l'interaction de plusieurs grandeurs non découplées. Par conséquent, les stratégies de contrôle de ces machines nécessitent l'élaboration d'algorithmes de complexité d'autant plus grande que les performances globales requises doivent être élevées.

L'objectif du contrôle par orientation du flux est le découplage des grandeurs responsables de la magnétisation de la machine et de la production du couple. La loi de commande consiste à établir l'ensemble des transformations pour passer d'un système possédant une double non linéarité structurelle à un système linéaire qui assure l'indépendance entre la création du flux et la production du couple comme dans une machine à courant continu à excitation séparée [8].

L'expression de couple moteur en fonction des flux rotorique et statorique dans un repère tournant dq (d : axe direct, q : axe en quadrature) est comme suit :

$$C = \frac{3}{2} p \frac{M'}{L} (\Phi_{rd} \cdot I_{rq} + \Phi_{rq} \cdot I_{sd}) \quad (1.8)$$

Où :

M' : L'inductance mutuelle maximale entre une phase statorique et une phase rotorique.

L : L'inductance propre d'une phase statorique.

Φ_{rd}, Φ_{rq} : Flux rotorique et statorique respectivement.

I_{rq} : Le courant induit rotorique dans l'axe q.

I_{sd} : Le courant statorique dans l'axe d.

Lors de l'application de la transformation de Park on choisit l'angle de rotation de repère dq de façon à avoir une composante en quadrature nulle $\Phi_{rq} = 0$, de sorte que le flux rotorique soit porté entièrement sur l'axe direct.

I.4. Les Onduleurs

I.4.1. Notions sur les onduleurs

L'électronique de puissance a permis de développer plusieurs moyens qui assurent des entraînements alternatifs à vitesse variable. Parmi ces moyens : les onduleurs autonomes commandés en modulation de largeurs d'impulsion (MLI) [9].

I.4.2. Classification des onduleurs

Un onduleur est un convertisseur statique assurant la conversion continu- alternatif, alimenté en continu, il modifie de façon périodique les connexions entre l'entrée et la sortie et permet d'obtenir de l'alternatif à la sortie.

Une première classification peut être faite en distinguant : onduleurs non autonomes et onduleurs autonomes [9].

I.4.2.1 Onduleurs non autonome :

Un onduleur est dit non autonome si l'énergie nécessaire pour la commutation des thyristors est fournie par le réseau alternatif qui est donc un réseau actif [9].

I.4.2.2 Onduleur autonome :

Un onduleur est dit autonome si l'établissement et la connexion entre l'entrée et la sortie ne dépend que de la commande des semi-conducteurs.

On distingue deux types d'onduleurs autonomes :

- les onduleurs autonomes de tension qui sont alimentés par une source de tension continue (source d'impédance interne négligeable).
- Les onduleurs autonomes de courant qui sont alimentés par une source de courant continu [9].

I.4.3. Types d'onduleurs autonomes de tension :

I.4.3.1 Les onduleurs monophasés :

Ce type d'onduleurs est destiné à alimenter des charges alternatives monophasées, ce sont utilisés pour des applications de faible puissance. On distingue deux configurations de base: en demi-pont ou en pont complet [10].

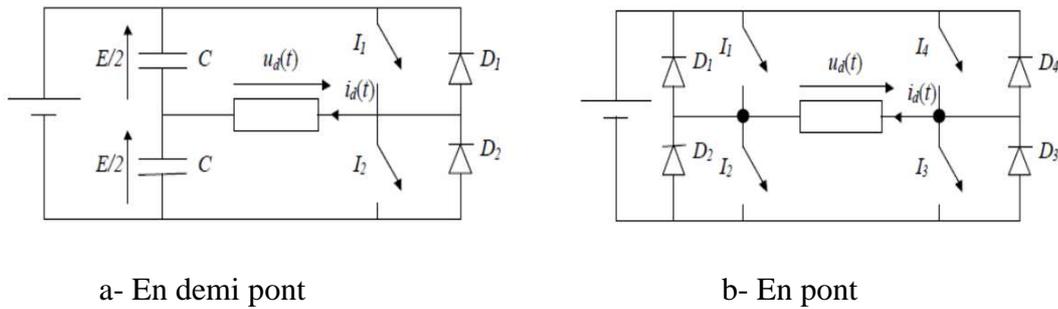


Figure 1.12 : schéma de l'onduleur monophasé.

I.4.3.2 Les onduleurs triphasés

Les onduleurs triphasés couvrent la gamme des moyennes et des fortes puissances. L'objectif de cette topologie est de fournir une source de tension triphasée, dont l'amplitude, la phase et la fréquence sont contrôlables [11].

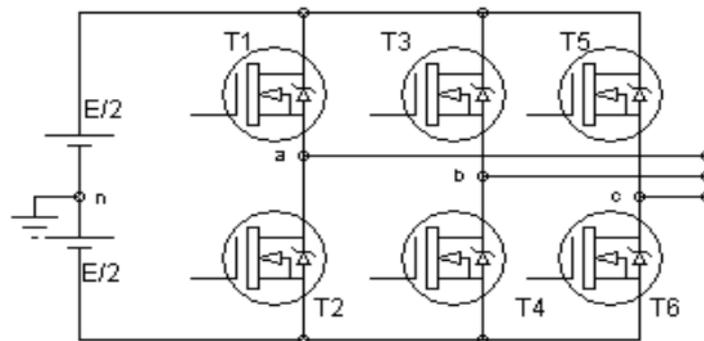


Figure 1.13 Schéma de principe d'un onduleur triphasé de tension.

I.4.4. Commandes d'onduleurs

I.4.4.1. Types de commandes :

On distingue les stratégies de commande suivante :

- a- commande pleine onde (dite 180°).
- b- Commande décalée.
- c- Commande à Modulation de largeur d'impulsion (MLI).

La tension générée par les stratégies (a, b) à une forme rectangulaire, sa décomposition en série de Fourier a montrée que cette forme d'onde est riche en harmoniques. Il est inutile d'atténuer ces harmoniques par un filtre, en effet la fréquence est variable, en plus le premier

harmonique à éliminer (L'harmonique 5) à une fréquence très voisine de celle du fondamental.

Dans un onduleur à modulation de largeur d'impulsion, au lieu de former chaque alternance d'une tension de sortie avec un seul créneau rectangulaire, on la forme de plusieurs créneaux de largeurs bien spécifiques pour éliminer les harmoniques à fréquences proches de fondamentale.

La modulation de largeur d'impulsion s'avère être la technique la plus appropriée pour la commande de l'onduleur tout en ayant une bonne neutralisation de l'onde de sortie [9].

I.4.4.2. Différentes techniques MLI

Plusieurs techniques d'obtention d'onde MLI sont utilisées pour générer le signal de contrôle nécessaire à la commande et au blocage des éléments semi-conducteurs.

1- Technique MLI engendrée

La commande MLI Triangulo-sinusoidale consiste à comparer une valeur de tension de référence de fréquence F_r , image du signal souhaité à la sortie appelée modulante, à une porteuse triangulaire ou en dent de scie de fréquence F_p .

Les points d'intersection entre la modulante et la porteuse engendrent l'enclenchement/déclenchement constituant ainsi une impulsion de durée variable et l'ensemble de ces impulsions reconstitue, de ce fait, le fondamental de la sinusoïde de référence [12].

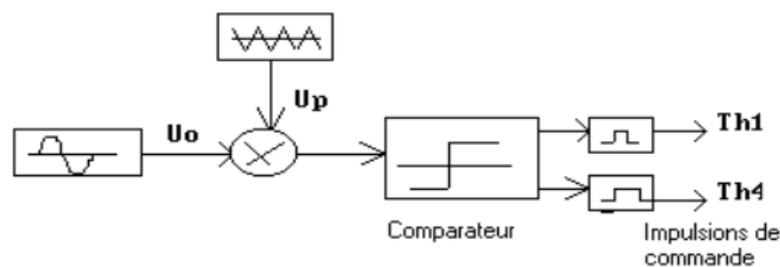


Figure 1.14 : Schéma synoptique de la loi de modulation triangulo-sinusoidale.

2-Technique MLI calculée (programmée)

Cette technique consiste à calculer les instants de commutation des interrupteurs de manière à répondre à certains critères portant sur le spectre fréquentiel de l'onde délivrée par l'onduleur. Ces séquences de fonctionnement sont alors mémorisées et restituées cycliquement pour assurer la commande des interrupteurs. Les critères usuellement retenus

sont l'élimination d'harmoniques de rangs spécifiés ou l'élimination des harmoniques dans une bande de fréquences spécifiée [13]

La technique MLI « programmée » est basée sur l'algorithme de Patel et Hoft. Dans cette technique, il est possible d'asservir le fondamental de la tension MLI et d'annuler les amplitudes des (m-1) premiers harmoniques.

La tension MLI « programmée » est définie en fonction des angles exacts de commutation $\alpha_1, \dots, \alpha_m$ qui correspondent aux instants de commutation de la tension MLI « programmée » d'une valeur positive $+E/2$ à une valeur négative $-E/2$ ou inversement.

Un circuit numérique génère alors la tension MLI « programmée » en fonction du temps. En conclusion, on peut dire que la technique MLI « programmée » présente de nombreux avantages :

- Asservissement de la tension V du fondamental
- Variation de la fréquence f du fondamental en utilisant la relation de conversion d'une valeur angulaire en valeur temporelle : $\alpha = 2\pi ft$
- Elimination des (m-1) premiers harmoniques

Ces avantages permettent de remplacer l'alimentation sinusoïdale idéale avec une alimentation pratique ayant un taux d'harmoniques que l'on peut réduire à volonté [14].

I.5. Conclusion :

Dans ce chapitre nous avons vu des généralités sur les moteurs asynchrones, les onduleurs de tension et la commande MLI. Le principe de la commande scalaire V/f est de garder une couple moteur constant pour faire varier la vitesse d'une manière contrôlée.

Il existe plusieurs types d'onduleurs de tension. Pour nous, on a utilisé un onduleur de tension autonome triphasé piloté par une commande MLI.

Dans notre cas, la commande MLI calculée est beaucoup plus performante que la MLI engendrée car les harmoniques proche au fondamentales sont non seulement atténués mais sont annulée. Dans le chapitre suivant, nous présentons la technique et l'algorithme utilisés pour générer le signal de commande MLI.

Chapitre III

ALGORITHME MLI AVEC ELIMINATION DES HARMONIQUES ET RESEAUX DE NEURONES

II.1 Introduction :

Nous avons vu au chapitre précédent en général les avantages de la MLI programmée par rapport à la MLI engendrée. La MLI programmée se base sur l'élimination des harmoniques non désirés qui peuvent produire des vibrations et des ondulations de couple et des nombreuses conséquences indésirables. Cette algorithme est appelé MLI avec élimination sélective d'harmoniques MLI ESH en anglais SHE PWM (Selective Harmonics Elimination Pulse –Width Modulation). Il se base sur la méthode de PATEL et HOFT.

Le calcul des angles de commutation avec cette méthode se fait à partir des équations non linéaires et transcendantes. Ce qui a obligé les chercheurs à faire appel à des méthodes numériques telles que Newton-Raphson. Le problème avec cette méthode c'est le choix des bonnes valeurs initiales nécessaires pour la convergence. De plus, le calcul de ces angles ne peut pas se faire en temps réel, les angles sont stockés en mémoire ce qui rend le système non optimale pour les applications dont les changements de fréquence et de tension sont rapides tel que le variateur de vitesse [15].

Récemment, un algorithme basé sur l'approche d'interpolation polynomiale de trajectoire des angles SHE PWM a été proposé. Suivant le même principe nous allons implémenter un nouvel algorithme à l'aide du réseau de neurones artificiel pour calculer les angles de commutation et générer les signaux SHE PWM de commande de l'onduleur de tension en temps réel.

Le nombre de commutations par période machine d'un signal MLI devient important dans les zones de faible vitesse. Il est donc nécessaire d'avoir recours à des méthodes de calcul des signaux MLI, adaptées à ce mode de fonctionnement afin de pouvoir couvrir toute la gamme de fonctionnement [16].

II.2. La méthode MLI ESH (Patel-Hoft):

Les harmoniques de faible rang sont les plus gênants. Les pulsations de couple qu'ils génèrent appartiennent à des domaines de fréquence dans lesquels les parties mécaniques de la machine ou de la charge sont en générales sensibles. L'annulation de ces harmoniques peut faire disparaître ces phénomènes, à condition qu'ils en soient l'unique cause. En effet, les défauts d'alignements ou de symétrie d'arbre pourraient être à l'origine des mêmes phénomènes, bien que totalement indépendants de la commande [16].

Les harmoniques de rang important n'affectent pas le bon fonctionnement de la machine car ils sont trop loins de la fréquence de pulsation de couple.

II.2.1. Principe de Patel-Hoft :

Les signaux MLI décrivant les trois tensions de sortie du convertisseur doivent posséder des propriétés qui contribuent à orienter leurs caractéristiques vers celles d'une onde sinusoïdale. Afin de s'en approcher le plus possible, on pourra, dans certains, cas leur attribuer les mêmes propriétés de symétrie qu'une onde sinusoïdale [16].

Le but de cette technique est d'éliminer un certain nombre d'harmonique de rang faible et de contrôler l'onde fondamentale (Figure 2.1). Cette dernière comporte deux niveaux et supposée périodique d'amplitude unité et de symétrie demi onde.

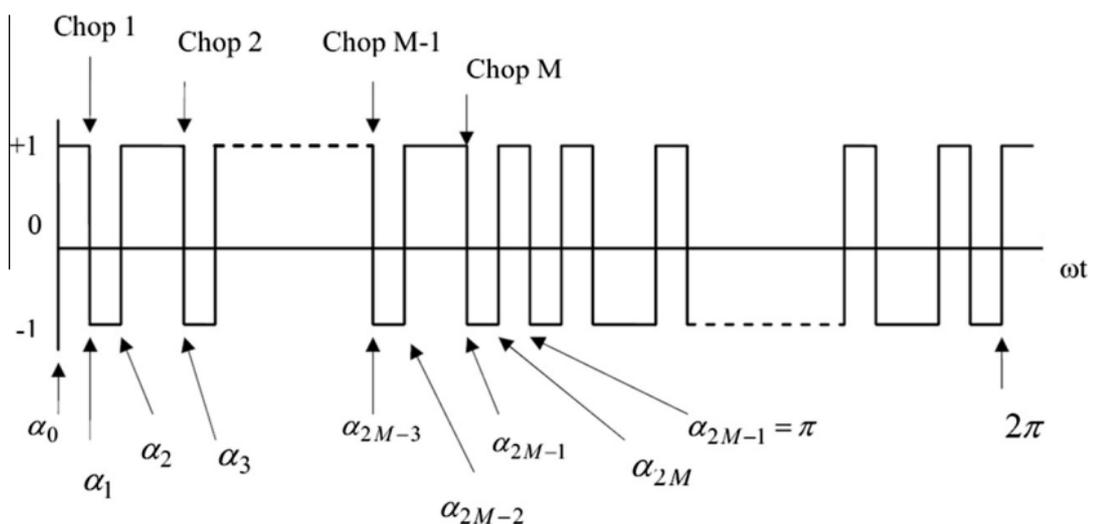


Figure 2.1: Un signal MLI à deux niveaux avec une symétrie demi onde.

Soit f la fonction représentant le signal MLI en fonction de α ($\alpha = \omega t$), On peut écrire donc :

$$f(\alpha) = -f(\alpha + \pi) \quad (2.1)$$

La fonction f est décomposable en séries de Fourier :

$$f(\alpha) = a_0 + \sum_{n=0}^{\infty} (a_n \sin(n\alpha) + b_n \cos(n\alpha)) \quad (2.2)$$

Avec $n = 1, 2, 3, 4, \dots$

Où :

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(\alpha) d\alpha \quad (2.3)$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \sin(n\alpha) d\alpha \quad (2.4)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \cos(n\alpha) d\alpha \quad (2.5)$$

Remplaçons $f(\alpha)$ par sa valeur dans l'équation (2.4) :

$$a_n = \frac{2}{\pi} \left[\int_{\alpha_0}^{\alpha_1} (-1)^0 \sin(n\alpha) d\alpha \right] + \frac{2}{\pi} \left[\int_{\alpha_2}^{\alpha_3} (-1)^2 \sin(n\alpha) d\alpha \right] + \dots + \frac{2}{\pi} \left[\int_{\alpha_{2M}}^{\alpha_{(2M+1)}} (-1)^{2M} \sin(n\alpha) d\alpha \right]$$

$$a_n = \frac{2}{\pi} \left[\sum_{k=0}^{2M} \int_{\alpha_k}^{\alpha_{(k+1)}} (-1)^k \sin(n\alpha) d\alpha \right]$$

$$a_n = \frac{2}{n\pi} \left[\sum_{k=0}^{2M} \int_{\alpha_k}^{\alpha_{(k+1)}} (-1)^k (\cos(n\alpha_k) - \cos(n\alpha_{k+1})) \right]$$

Avec : $\alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \dots \leq \alpha_{2M+1}$

$$a_n = \frac{2}{n\pi} \left[\cos(n\alpha_0) - \cos(n\alpha_{2M+1}) + 2 \sum_{k=1}^{2M} \int_{\alpha_k}^{\alpha_{k+1}} (-1)^k \cos(n\alpha_k) \right]$$

On a :

$$\begin{array}{ll} \alpha_0 = 0 & \text{alors} \quad \cos(n\alpha_0) = 1 \\ \alpha_{2M+1} = \pi & \cos(n\alpha_{2M+1}) = (-1)^n \end{array}$$

Donc :

$$a_n = \frac{2}{n\pi} \left[1 - (-1)^n + 2 \sum_{k=1}^{2M} (-1)^k \cos(n\alpha_k) \right] \quad (2.6a)$$

De même pour b_n on trouve :

$$b_n = \frac{-4}{n\pi} \sum_{k=1}^{2M} (-1)^k \sin(n\alpha_k) \quad (2.6b)$$

La valeur moyenne est nulle et seulement les harmoniques impairs existent. Par conséquence, l'indice n prend les valeurs impaires 1, 3, 5, 7, 9, ... M . (M est un nombre impair).

On a donc :

$$a_n = \frac{4}{n\pi} \left[1 + \sum_{k=1}^{2M} (-1)^k \cos(n\alpha_k) \right] \quad (2.7a)$$

$$b_n = \frac{4}{n\pi} \left[- \sum_{k=1}^{2M} (-1)^k \sin(n\alpha_k) \right] \quad \text{n impair} \quad (2.7b)$$

De fait de la symétrie quart d'onde on a:

$$\begin{array}{ll} \alpha_k + \alpha_{2M+1-k} = \pi & k=1, 2, 3, \dots, M \\ f(\pi - \alpha) = f(\alpha) & \end{array}$$

D'où :

$$\sin(n\alpha_k) = \sin(n(\pi - \alpha_{2M-k+1}))$$

$$\sin(n\alpha_k) = \sin(n\pi) \cos(n\alpha_{2M-k+1}) - \cos(n\pi) \sin(n\alpha_{2M-k+1})$$

Pour n impair on a :

$$\sin(n\pi) = 0 \quad \text{et} \quad \cos(n\pi) = 1$$

Alors :

$$\sin(n\alpha_k) = \sin((\pi - \alpha_{2M-k+1})) \quad (2.8)$$

On remplace (2.8) dans (2.7b) :

$$b_n = \frac{4}{n\pi} \sum_{k=1}^{2M} (\sin(n\alpha_k) - \sin(n\alpha_{2M-k+1})) = 0$$

De plus :

$$\cos(n\alpha_k) = \cos(n(\pi - \alpha_{2M-k+1}))$$

$$\cos(n\alpha_k) = \cos(n\pi) \cos(n\alpha_{2M-k+1}) + \sin(n\pi) \sin(n\alpha_{2M-k+1})$$

Donc :

$$\cos(n\alpha_k) = -\cos(n - \alpha_{2M-k+1})$$

En remplaçant dans (2.7a) :

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(n\alpha_k) \right] \quad (2.9)$$

Pour n pair :

$$a_n = b_n = 0 \quad (2.10a)$$

Pour n impair :

$$b_n = 0 \quad (2.10b)$$

Les harmoniques multiples de trois sont automatiquement annulés. En effet, dans un système de charges équilibrées comme dans le bobinage de moteur asynchrone la liaison du neutre au point du milieu est supprimée. La suppression de cette liaison supprime les harmoniques de rang 3 et multiple de 3.

Comme l'amplitude du fondamental doit être fixée à une valeur déterminée, ceci fixe la première valeur de m à 3 (m étant le nombre de commutations par quart d'onde ou nombre de découpages par demi-onde). Par conséquent lorsque m augmente successivement par pas égal à 2, le nombre d'harmoniques de courant qui seront éliminés augmente par pas égal à 1 [18].

L'équation (2.11) possède $M-1$ variables, il s'agit d'un ensemble d'angles $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{M-1}, \alpha_M$. Où $\alpha_1, \alpha_3, \dots, \alpha_{2i+1}$ définissent les transitions négatifs et $\alpha_2, \alpha_4, \dots, \alpha_{2i}$ définissent les transitions positifs.

Ces angles sont obtenus en assimilant tous les $M-1$ harmoniques à zéro et en attribuant une valeur i_m appelé indice de modulation de la composante fondamentale. Cela est formulé par le système d'équation (2.11).

$$\begin{aligned}
 f_1(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M) &= \frac{4}{\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(\alpha_k) \right] + i_m = 0 \\
 f_2(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M) &= \frac{4}{5\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(5\alpha_k) \right] = 0 \\
 f_3(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M) &= \frac{4}{7\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(7\alpha_k) \right] = 0 \\
 &\dots\dots\dots \\
 f_M(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M) &= \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(n\alpha_k) \right] = 0
 \end{aligned} \tag{2.11}$$

II.2.2. Calcul de valeurs exactes des angles de commutation par la Méthode de Newton-Raphson :

Les équations de système (2.11) sont non-linéaires et transcendantes, on a démontré qu'on peut les résoudre en utilisant des méthodes numériques [17].

Le système des équations (2.11) satisfait les conditions de la méthode de Newton-Raphson, i.e. : tous les équations f_i sont continues et continument différentiables.

Pour assurer la convergence de cette méthode, on doit obtenir un bon estimé initial de la solution exacte recherchée. Un algorithme a été mis au point pour estimer les valeurs initiales, c'est l'algorithme de Taufik, Mellitt et Goodmanla (TMG) [18].

II.2.2.1. Algorithme TMG [18]:

Pour k impair :

$$\alpha_k = \frac{60(k+1)}{(M+1)} - \left[\frac{2 \times 60}{(M+1)} \times \frac{\Delta_k}{0.8} \times NP1 \right]$$

Avec :

$$\Delta_k = 0.4025 \sin \left[\frac{59.184^\circ (k-0.5)}{M} + 60.408^\circ \right] \quad k = 1, 3, 5, \dots, M$$

Pour k pair

$$\alpha_k = \frac{60k}{(M+1)} - \left[\frac{2 \times 60}{(M+1)} \times \frac{\Delta_k}{0.8} \times NP1 \right]$$

Avec :

$$\Delta_k = 0.381 \sin \left[\frac{k}{M-1} \left(58.558 - \frac{1.135 \times 11}{M-2} \right) \right] \quad k = 2, 4, 6, \dots, M-1$$

NP1 est le taux de modulation identique à im , il prend des valeurs entre 0 et 0.8.

Une fois ces valeurs sont calculées on procède les étapes itératives de Newton-Raphson.

II.2.2.2. Algorithme de calcul des angles [19]:

Soit le vecteur solution du système d'équation (2.11) :

$$\alpha^* = (\alpha^*_1, \alpha^*_2, \alpha^*_3, \dots, \alpha^*_M)$$

$$f_i(\alpha) = 0 \quad i=1, 2, \dots, M$$

En développant les fonctions f_i en série de Taylor au voisinage d'un estimé $\alpha^{(k)}$ ($k^{\text{ème}}$ itération) proche de α^* , on obtient :

$$f_i(\alpha) = f_i(\alpha^{(k)} + (\alpha^* - \alpha^{(k)})) \quad i=1, 2, \dots, M$$

$$\begin{aligned}
f_i(\alpha) &= f_i(\alpha^{(k)}) + \sum_{j=1}^M \left[\frac{\partial f_i(\alpha)}{\partial \alpha_j} \right]_{\alpha=\alpha^{(k)}} (\alpha_j^* - \alpha_j^{(k)}) + \dots \\
\dots &+ \frac{1}{2} \sum_{j=1}^M \sum_{r=1}^M (\alpha_j^* - \alpha_j^{(k)}) (\alpha_r^* - \alpha_r^{(k)}) \left[\frac{\partial^2 f_i(\alpha)}{\partial \alpha_j \partial \alpha_r} \right]_{\alpha=\alpha^{(k)}} + \dots + \dots = 0
\end{aligned} \tag{2.12}$$

Du fait que $\alpha^{(k)}$ est proche de α^* on néglige $(\alpha_j^* - \alpha_j^{(k)})^2$ et les termes d'ordre supérieur.

L'équation (2.12) devient :

$$\sum_{j=1}^M \left[\frac{\partial f_i(\alpha)}{\partial \alpha_j} \right]_{\alpha=\alpha^{(k)}} (\alpha_j^* - \alpha_j^{(k)}) = -f_i(\alpha^{(k)}) \tag{2.13}$$

En introduisant la matrice des dérivées premières :

$$E^{(k)} = \left(E_{ij}^{(k)} \right)$$

$$\text{Avec : } E_{ij}^{(k)} = \left[\frac{\partial f_i(\alpha)}{\partial \alpha_j} \right]_{\alpha=\alpha^{(k)}} \quad i=1, 2, \dots, M \quad j=1, 2, \dots, M$$

D'où :

$$E^{(k)} = \frac{8}{\pi} \begin{bmatrix} \sin(\alpha_1) & -\sin(\alpha_2) & \dots & \sin(\alpha_M) \\ \sin(5\alpha_1) & -\sin(5\alpha_2) & \dots & \sin(5\alpha_M) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(n\alpha_1) & -\sin(n\alpha_2) & \dots & \sin(n\alpha_M) \end{bmatrix}$$

Définissons le vecteur erreur :

$$\Delta \alpha^{(k)} = \left(\Delta \alpha_1^{(k)}, \Delta \alpha_2^{(k)}, \Delta \alpha_3^{(k)}, \dots, \Delta \alpha_M^{(k)} \right)$$

$$\text{Où : } \Delta \alpha_j^{(k)} = \alpha_j^* - \alpha_j^{(k)}$$

Soit :

$$F^{(k)} = \left(F_1^{(k)}, F_2^{(k)}, \dots, F_M^{(k)} \right)$$

$$\text{Avec : } F_i^{(k)} = -f_i(\alpha^{(k)})$$

Le système (2.13) devient :

$$E^{(k)} \Delta \alpha^{(k)} = F^{(k)} \quad (2.14)$$

Le vecteur $\Delta \alpha^{(k)}$ est l'inconnu. Le système (2.14) est linéaire donc on peut déterminer $\Delta \alpha^{(k)}$.

Une fois $\Delta \alpha^{(k)}$ est calculé, on détermine $\alpha^{(k+1)}$ par :

$$\alpha^{(k+1)} = \alpha^{(k)} + \Delta \alpha^{(k)}$$

On continue jusqu'à ce que :

$$|\alpha^* - \alpha^{(k)}| \rightarrow 0$$

En pratique, α^* étant l'inconnue, on arrête les opérations par l'un des tests suivants :

- 1) $k \geq k_{\max}$
- 2) $|f_i(\alpha^{(k)})| \leq E_0$

E_0 est une borne supérieure de l'erreur fixée a priori et k_{\max} le nombre maximum des itérations.

II.2.3. calcul des angles de commutation :

Nous avons mis au point un programme Matlab à base de cet algorithme pour calculer les angles de commutation. Pour des valeurs de im entre 0.01 et 0.06. Les résultats sont mis dans le tableau 2.1 :

Les angles en degré °	les valeurs de im (x100)					
	1	2	3	4	5	6
α_1	5	4,97865347	4,95730219	4,93594599	4,9145847	4,89321812
α_2	5	5,00349275	5,00698604	5,01047959	5,01397307	5,01746617
α_3	10	9,97862815	9,9572428	9,93584362	9,91443026	9,89300237
α_4	10	10,0064647	10,0129217	10,0193706	10,0258109	10,0322418
α_5	15	14,9784294	14,956836	14,9352193	14,9135789	14,8919143
α_6	15	15,0090513	15,0180863	15,0271043	15,0361047	15,0450868
α_7	20	19,9781553	19,9562802	19,9343744	19,9124372	19,8904682
α_8	20	20,0113296	20,0226359	20,0339183	20,0451761	20,0564087
α_9	25	24,9778711	24,9557071	24,9335074	24,9112715	24,888999
α_{10}	25	25,0133453	25,0266631	25,0399527	25,0532135	25,066445
α_{11}	30	29,9776231	29,9552091	29,9327574	29,9102676	29,8877391
α_{12}	30	30,0151272	30,0302254	30,045294	30,0603326	30,0753405
α_{13}	35	34,9774454	34,9548543	34,9322262	34,9095609	34,8868576
α_{14}	35	35,0166939	35,03336	35,0499977	35,0666068	35,0831866
α_{15}	40	39,9773637	39,9546941	39,9319908	39,9092534	39,8864813
α_{16}	40	40,0180581	40,0360918	40,0541009	40,072085	40,0900437
α_{17}	45	44,9773985	44,9547688	44,9321107	44,9094238	44,8867077
α_{18}	45	45,0192289	45,0384389	45,05763	45,0768018	45,0959541
α_{19}	50	49,9775657	49,9551101	49,9326329	49,9101339	49,8876126
α_{20}	50	50,0202138	50,0404158	50,060606	50,0807842	50,1009503
α_{21}	55	54,9778783	54,9557433	54,9335947	54,9114323	54,8892559
α_{22}	55	55,0210196	55,0420354	55,0630476	55,0840561	55,1050609
α_{23}	60	59,978347	59,9566891	59,9350263	59,9133584	59,8916852

Tableau 2.1 : les angles de commutation en fonction de im.

II.2.4. Interprétation des résultats :

L'inconvénient de la méthode de Patel-Hoft c'est la difficulté d'implémentation ou de réalisation en temps réel, vu de la complexité de l'algorithme proposé, en débutant de l'estimation des angles initiales et en finissant par le calcul des angles par la méthode de Newton-Raphson. La technique de Patel-Hoft est donc une technique 'off-line'. Autrement dit, les valeurs exactes des angles de commutation sont d'abord calculées sur ordinateur ensuite ces valeurs sont stockées en mémoire pour être utilisées dans une commande de vitesse d'un moteur.

Pour pouvoir implémenter la commande de vitesse de moteur en temps réel, nous allons utiliser un algorithme basé sur les réseaux de neurones artificiels (RNA).

Dans la suite de ce chapitre nous présentons quelque généralité sur les RNA et l'algorithme que nous avons implémenté. Les résultats de la méthode de Patel-Hoft sont précis, ce qui nous a aidé à les utiliser comme base de données nécessaire pour la séquence d'apprentissage de réseaux de neurones.

II.3. Réseau de neurones artificiels :

II.3.1. Généralité sur les réseaux de neurones artificiels :

II.3.1.1. Définitions :

- **Le neurone biologique :**

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angströms (10^{-9} m) entre l'axone du neurone afférent et les dendrites (on dit une dendrite) du neurone efférent. La jonction entre deux neurones est appelée la synapse (Figure 2.2) [20].

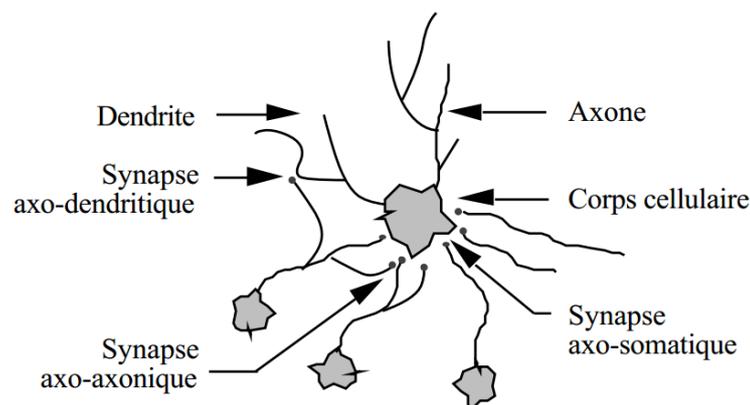


Figure 2.2: Un neurone avec son arborisation dendritique

- **Réseaux de neurones artificiels :**

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [20].

II.3.1.2. Modélisation mathématique :

Comme le réseau neuronal biologique, Le RNA est une interconnexion de nœuds, analogue aux neurones. Chaque réseau neuronal a trois composantes essentielles: le caractère de nœud, la topologie de réseau, et l'apprentissage des règles.

- Le caractère nœud détermine la manière dont les signaux sont traités par le nœud, tel que le nombre d'entrées et de sorties associées au nœud, le poids associé à chaque entrée et sortie, et la fonction d'activation.
- La topologie de réseau détermine les moyens parmi lesquels les nœuds sont organisés et connectés.
- Les règles d'apprentissage déterminent la façon dont les poids sont initialisés et ajustés.

II.3.1.2 .1. Structure de nœud

Le modèle de base pour un nœud dans l'RNA est représenté sur la Figure 2.3.

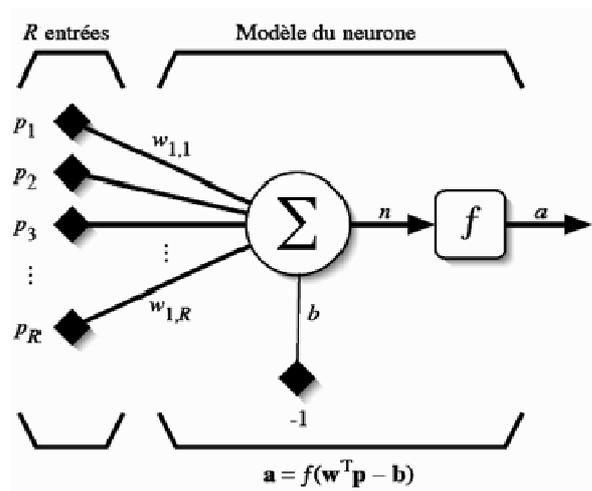


Figure 2.3: structure d'un nœud de RNA.

Chaque nœud reçoit plusieurs entrées des autres par l'intermédiaire de connexions associées qui ont des poids (chaque entrée est multipliée par un poids w_i). Quand la somme pondérée des entrées dépasse la valeur de seuil du nœud, il active le signal par une fonction d'activation

f et il l'envoie aux nœuds voisins. Ce processus peut être exprimé comme un modèle mathématique :

$$a = f\left(\sum_{i=1}^R w_i p_i - b\right) \quad (2.15)$$

Où a est la sortie du nœud, w_i sont les poids associés à chaque entrée, p_i représentent les entrées et b la valeur de seuil.

La fonction de transfert a de nombreuses formes. Une fonction de transfert non linéaire est plus utile que les linéaires [21], puisque seuls quelques problèmes sont linéairement séparables. On va utiliser la fonction sigmoïde (Equation 2.16 Fig. 2.6).

$$f(x) = \frac{1}{1 + e^{-\beta x}} \quad (2.16)$$

Où β est un coefficient.

II.3.1.2 .2. La topologie du RNA :

En général Dans un RNA, les nœuds sont organisés en réseaux linéaires, appelée couches. Habituellement, il existe plusieurs types de couches, des couches d'entrée, des couches de sortie et des couches cachées. Il peut n'y avoir aucune couche cachée. La conception de la topologie du réseau implique la détermination du nombre de couche et de nœuds à chaque couche. Il existe plusieurs topologies de réseaux, la plus utilisée est la topologie multicouche (Figure 2.4)

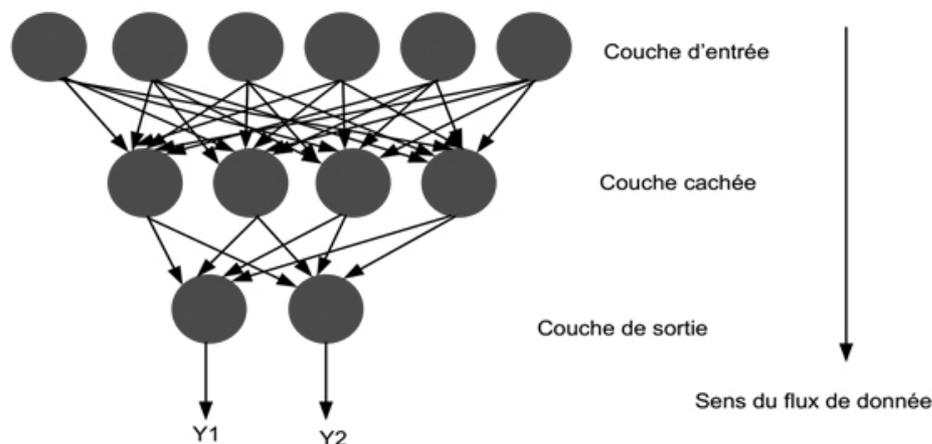


Figure 2.4 Topologie d'un réseau multicouche (MLP).

II.3.1.2 .2. Apprentissage du RNA :

Le RNA utilise un processus d'apprentissage pour former le réseau. Pendant l'apprentissage, les poids sont ajustés à des valeurs souhaitées. L'apprentissage peut être classé en deux grandes catégories: l'apprentissage supervisé et l'apprentissage non supervisé. Dans l'apprentissage supervisé, un ensemble des entrées et des sorties est fourni. Les coefficients de pondération (poids) sont ajustés pour minimiser l'erreur entre la sortie du réseau et la sortie correcte. Pour les réseaux utilisant un apprentissage supervisé, le réseau doit être formé en premier. Lorsque le réseau donne les résultats souhaités pour une série d'entrées, les poids sont fixes et le réseau peut être mis en fonctionnement. En revanche, l'apprentissage non supervisé n'utilise pas les valeurs de sortie cibles. Le réseau tente de découvrir le bon motif dans les données d'entrée seulement. Dans notre cas la base de données existe alors on utilise un apprentissage supervisé. Il existe plusieurs méthodes d'apprentissage, parmi lesquelles, la méthode de gradient.

II.3.2. Méthode de gradient [14] :

Soit \mathbf{D} , un ensemble représentant une base de données. Chaque élément de cet ensemble représente un couple $(\mathbf{x}, \mathbf{y}^*)$ appelé exemple d'apprentissage, où x est une valeur d'entrée et y^* est la valeur désirée de sortie de chaque neurones de réseau. Le but de l'apprentissage est donc de déterminer les valeurs \mathbf{W}^* de la matrice \mathbf{W} des poids des connexions du réseau de telle sorte que les sorties \mathbf{y} soient proches des valeurs désirées \mathbf{y}^* . \mathbf{W}^* est donc la solution d'un problème d'optimisation, consistant à minimiser une fonction de coût : $\mathbf{E}(\mathbf{W}, \mathbf{D})$.

Au début, on choisit une valeur initiale $\mathbf{W}(0)$ de la matrice \mathbf{W} , puis on utilise un processus itératif dans lequel on tente d'optimiser la fonction \mathbf{E} . À chaque itération, cette optimisation implique deux étapes

- Le choix de la direction dans laquelle on va chercher la valeur suivante $\mathbf{W}(t+1)$.
- Et le déplacement η le long de cette direction.

À chaque itération, on retire un exemple d'apprentissage $(\mathbf{x}, \mathbf{y}^*)$ et on calcule une nouvelle estimation de $\mathbf{W}(t)$, les itérations sont réalisées en deux phases :

- a- Propagation :** À chaque itération, un élément de l'ensemble d'apprentissage \mathbf{D} est introduit à travers la couche d'entrée. L'évaluation des sorties du réseau se fait couche par couche, de l'entrée du réseau vers sa sortie.

b- Rétro propagation : À la sortie du réseau, on forme le critère de performance \mathbf{J} en fonction de la sortie réelle du système et sa valeur désirée. Puis, on évalue le gradient de \mathbf{J} par rapport aux différents poids en commençant par la couche de sortie et en remontant vers la couche d'entrée. Le gradient \mathbf{J} est calculé par :

$$J = \frac{1}{2} \sum_{k=1}^{N_2} (y_i^{des} - y_i)^2 \quad (2.17)$$

Où :

y_i^{des} : La composante i de la sortie désirée du système.

y_i : La composante i de la sortie calculée du système.

N_2 : Le nombre d'exemples (de valeurs) dans la base d'apprentissage.

Le calcul des poids se fait par :

$$\Delta W = -\eta \frac{\partial J}{\partial W} \quad (2.18)$$

$$W_{ij}^{(k)}(t+1) = W_{ij}^{(k)}(t) + \Delta W \quad (2.19)$$

On doit calculer donc $\frac{\partial J}{\partial W}$.

On se base sur la topologie représentée sur la figure 2.5.

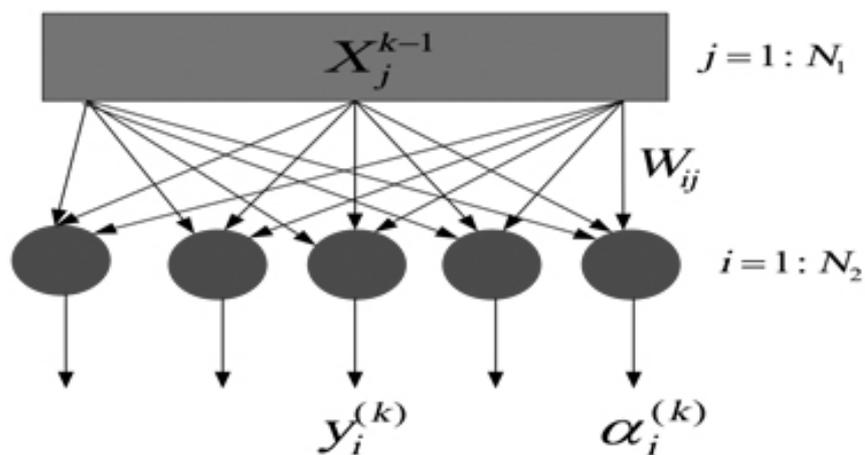


Figure 2.5 Représentation de la couche de sortie d'un réseau de neurones.

On peut écrire :

$$\frac{\partial J}{\partial W_{ij}^{(k)}} = \frac{\partial J}{\partial y_i^{(k)}} \frac{\partial y_i^{(k)}}{\partial a_i^{(k)}} \frac{\partial a_i^{(k)}}{\partial W_{ij}^{(k)}} \quad (2.20)$$

D'après l'équation 2.17 :

$$\frac{\partial J}{\partial y_i^{(k)}} = -(y_i^{des} - y_i)^2 \quad (2.21)$$

D'autre part :

$$y_i^{(k)} = g(a_i^{(k)}) \Rightarrow \frac{\partial y_i^{(k)}}{\partial a_i^{(k)}} = g'(a_i^{(k)}) \quad (2.22)$$

Et :

$$a_j^{(k)} = \sum_{j=1}^{N_2} W_{ij}^{(k)} X_i^{(k-1)} \Rightarrow \frac{\partial a_i^{(k)}}{\partial W_{ij}^{(k)}} = X_i^{(k-1)} \quad (2.23)$$

On substitue (2.21), (2.22) et (2.23) dans (2.20) :

$$\frac{\partial J}{\partial W_{ij}^{(k)}} = Err_i^{(k)} X_i^{(k-1)} \quad (2.24)$$

Où :

$$Err_i^{(k)} = \frac{\partial J}{\partial a_i^{(k)}} = -(y_i^{des} - y_i)^2 g'(a_i^{(k)})$$

On procède de la même manière pour les couche cachées on trouve :

$$\frac{\partial J}{\partial W_{ij}^{(k-1)}} = Err_i^{(k-1)} X_i^{(k-2)} \quad (2.25)$$

$$Err_i^{(k-1)} = \frac{\partial J}{\partial a_i^{(k-1)}} = \sum_{k=1}^{N_2} (Err_i^{(k)} W_{ij}) g'(a_i^{(k)})$$

Nous sommes arrivés donc à évaluer le gradient simple de la fonction de coût **J** par rapport aux paramètres **W** du réseau de neurones. Le but est de calculer les poids du réseau par l'équation itératifs (2.19), ces poids doivent correspondre à un minimum de la fonction de coût **J** dans l'espace des poids.

Pour cela il faut que la condition suivante soit vérifiée pour arrêter les itérations :

$$\frac{\partial J}{\partial W_{ij}} = 0 \quad \forall i, j$$

II.3.3. Centrage de données :

Le centrage des données est une étape indispensable, car la fonction d'activation qu'on va utiliser n'est pas linéaire que dans la zone au tour de son centre. De plus, si la base de données contient des valeurs qui diffèrent par des grandes valeurs entre eux on aura une partie qui ne donne aucun effet pour l'étape d'apprentissage (Figure 2.6).

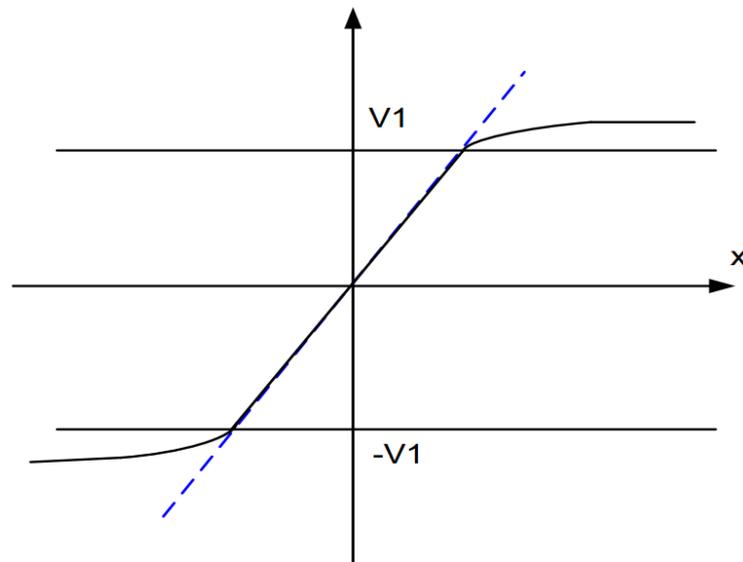


Figure 2.6 : Centrage de la base de données.

Le centrage de toute valeur v s'effectue par l'équation suivante :

$$v_{centrée} = \frac{(v - v_{moy})}{(2V_1)} \quad (2.26)$$

II.4. Une commande MLI à base des réseaux de neurones :

Le but principal de l'algorithme proposé consiste à construire un réseau multicouche qui calcule et génère les angles de commutation d'un signal MLI en sortie. Ces angles de commutation doivent être suffisamment proches de celles calculées par la méthode de Patel et

Hoft utilisant l'algorithme itératif de Newton Raphson afin de commander un onduleur triphasé conçu pour véhicules électriques.

II.4.1. Architecture de l'RNA :

Afin de simplifier l'architecture de RNA, qui sera mise en œuvre dans le circuit FPGA, la topologie de réseau multicouche choisie se compose d'une couche d'entrée, une couche cachée et une couche de sortie. En outre, pour les fonctions d'activation, la fonction de transfert tangente hyperbolique sigmoïde (Figure 2.7) est utilisée entre la couche d'entrée et la couche cachée, et une fonction linéaire est utilisée entre la couche cachée et la couche de sortie.

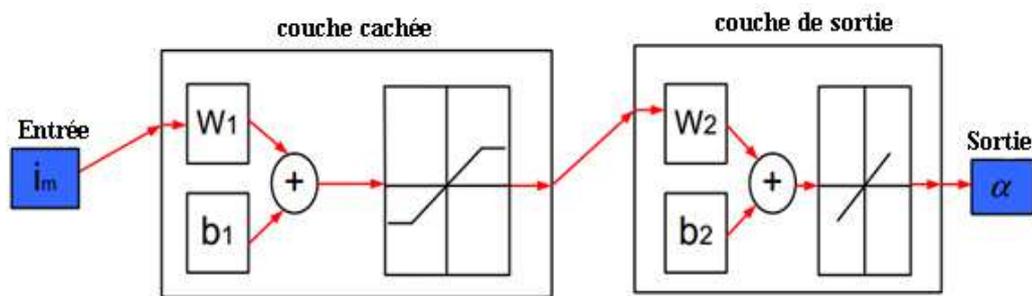


Figure 2.7 Architecture de réseau de neurones pour l'MLI.

Où : W_1 : la matrice des poids entre la couche d'entrée et la couche cachée.

b_1 : matrice des poids de la couche cachée.

W_2 : la matrice des poids entre la couche cachée et la couche de sortie.

b_2 : matrice des poids de la couche de sortie.

L'indice de modulation i_m , qui est l'entrée de RNA, varie de zéro à un. Il est modifié par pas de 0,01 pour générer 100 valeurs de fréquence et construire la base de données du dispositif de commutation angles. En revanche, les valeurs des angles de commutation, qui forment la sortie de RNA, dépendront de l'indice de modulation i_m selon le système d'équation (2.11). Pour rendre efficace la solution, pour permettre la convergence de l'étape de formation et d'avoir une bonne estimation des paramètres de RNA (poids et biais), l'intervalle de variation de i_m est divisée en six intervalles. Par conséquent, pour chaque intervalle, un réseau

spécifique est construit (poids et seuils). Le choix du nombre d'angles de commutation c'est-à-dire le nombre d'harmoniques à éliminer dépend de l'effet de l'harmonique.

On effectue le choix de nombre d'angles selon le nombre d'harmonique à éliminer. Donc le nombre des angles doit être important aux faibles fréquences et diminue lorsque la fréquence augmente. L'indice de modulation im prendra alors une variation inverse i.e. si im augmente alors le nombre des angles diminue.

Nous avons précisé notre choix de nombre des angles en divisant l'intervalle de variation de im normalisé en six sous-intervalles et en prenant un certain nombre d'angles pour chaque sous-intervalle.

Indice de modulation im	Le nombre des angles	Nom de réseaux de neurones	Nombre des neurones dans la couche cachée
$0 < im < 0.1$	23	ANN-1	1
$0.1 < im < 0.2$	19	ANN-2	1
$0.2 < im < 0.4$	15	ANN-3	1
$0.4 < im < 0.6$	7	ANN-4	1
$0.6 < im < 0.8$	5	ANN-5	1
$0.8 < im < 1$	3	ANN-6	1

Tableau 2.2 : Les nombres des angles en fonction d'indice de modulation im .

II.4.2. Calcul des paramètres des réseaux :

Les paramètres des six réseaux RNA, qui sont les poids et les seuils vont être calculés. Pour chaque ANN, la base de données correspondante, calculée dans la section précédente, est utilisée comme entrée dans un programme d'apprentissage basé sur la méthode du gradient (paragraphe II.3.2). Ce programme a deux conditions de rupture: la performance et le nombre d'itérations.

Le tableau 2.3 donne les poids et les seuils pour ANN-1, obtenus par le programme (Figure 2.8) pour $0 < im < 0,1$.

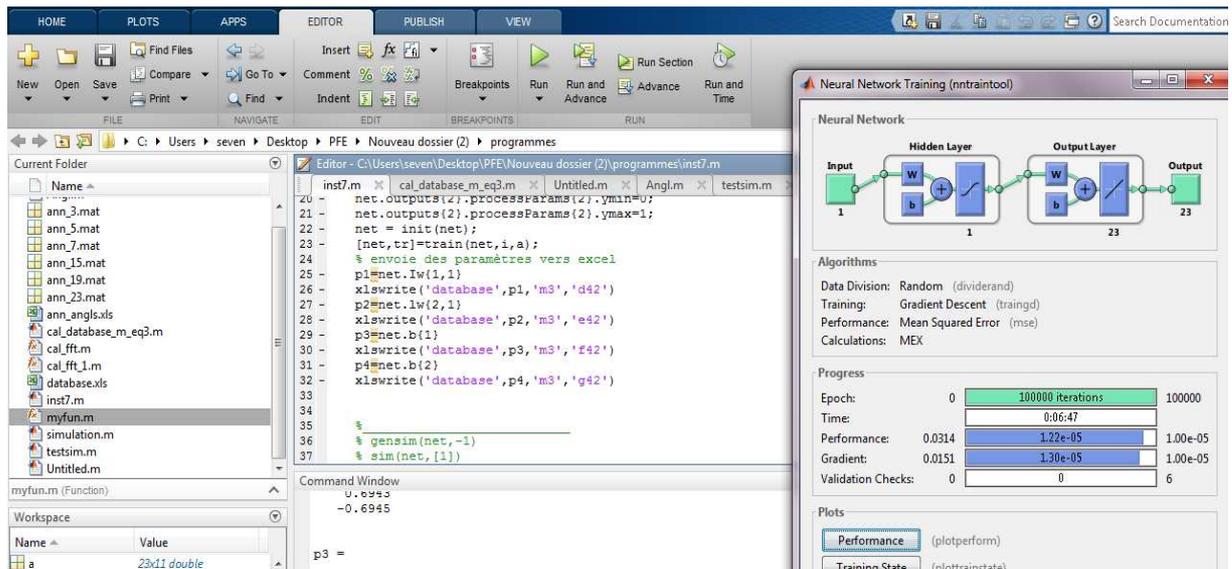


Figure 2.8 : Utilisation de la fonction "nn_train_tool" pour le calcul des paramètres de réseau ANN-1.

W_1	W_2	b_1	b_2
1,64820136	-0,69024787	-0,831685308	0,49609283
	0,6054777		0,506863681
	-0,69025598		0,496595326
	0,66344475		0,506174495
	-0,69031075		0,497088784
	0,67751912		0,506767646
	-0,69038081		0,497456221
	0,68336945		0,506945629
	-0,69044806		0,497673312
	0,68605924		0,50686505
	-0,69050257		0,497745743
	0,68759081		0,506620491
	-0,69053881		0,497688915
	0,68854898		0,506267687
	-0,6905533		0,497520664
	0,68918642		0,505844198
	-0,69054313		0,497258689
	0,68962782		0,505377705
	-0,69050497		0,49691983
	0,68994085		0,504890228
	-0,69043425		0,496520083
	0,69016493		0,504400452
	-0,69032426		0,496074885

Tableau 2.3 : Paramètre de réseau ANN-1 obtenus.

II.4.3. Calcul des angles de commutation :

Après avoir calculé les paramètres, les angles de commutation sont ensuite calculés par un programme qui traduit le réseau défini dans la figure 2.7. Le tableau 2.4 donne les résultats de ANN-1 pour $0,01 < im < 0,06$

Les angles en degré °	les valeurs de im (x100)					
	1	2	3	4	5	6
α_1	4,9933518	4,97872984	4,96103953	4,94045954	4,91758155	4,89339892
α_2	5,00329169	5,0053768	5,00789947	5,0108342	5,01409664	5,01754512
α_3	9,99340322	9,97873538	9,96098957	9,940345	9,91739523	9,89313672
α_4	10,0033184	10,0075352	10,0126369	10,018572	10,0251698	10,0321438
α_5	14,9934091	14,9785744	14,9606268	14,9397475	14,9165367	14,8920023
α_6	15,0037662	15,0097855	15,0170678	15,0255398	15,0349577	15,0449127
α_7	19,9933812	19,9783339	19,960129	19,9389504	19,9154069	19,8905209
α_8	20,0042559	20,0118511	20,0210402	20,0317302	20,043614	20,0561754
α_9	24,9933331	24,9780748	24,9596146	24,9381389	24,9142653	24,8890302
α_{10}	25,0047566	25,0137407	25,02461	25,0372549	25,0513117	25,0661701
α_{11}	29,9932766	29,9778412	29,9591667	29,9374418	29,9132911	29,8877631
α_{12}	30,0052111	30,0154265	30,0277854	30,0421632	30,0581465	30,0750412
α_{13}	34,9932222	34,9776666	34,9588468	34,9369528	34,9126141	34,8868874
α_{14}	35,0056087	35,0169124	35,0305882	35,0464979	35,0641841	35,0828789
α_{15}	39,9931788	39,9775771	39,9587014	39,9367424	39,9123314	39,8865284
α_{16}	40,0059464	40,0182047	40,0330354	40,0502886	40,0694684	40,0897419
α_{17}	44,9931547	44,9775935	44,9587669	44,9368649	44,9125174	44,8867814
α_{18}	45,0062249	45,0193097	45,0351403	45,0535568	45,0740296	45,09567
α_{19}	49,993157	49,9777327	49,9590718	49,9373626	49,9132293	49,8877198
α_{20}	50,0064465	50,0202338	50,0369142	50,0563194	50,0778913	50,1006935
α_{21}	54,9931923	54,9780088	54,9596392	54,9382688	54,9145123	54,889401
α_{22}	55,0066146	55,0209836	55,0383678	55,0585918	55,081074	55,1048382
α_{23}	59,9932666	59,9784334	59,9604876	59,9396102	59,9164017	59,8918697

Tableau 2.4 : Les angles de commutations calculées par le réseau ANN-1.

II.4.4. Interprétation des résultats:

Pour avoir un bon repère de décision de l'efficacité de la méthode de réseau de neurones, nous avons établi une comparaison entre les résultats obtenus par cette dernière et celle de Patel-Hoft. Sachant que la méthode de Patel-Hoft a donné des résultats.

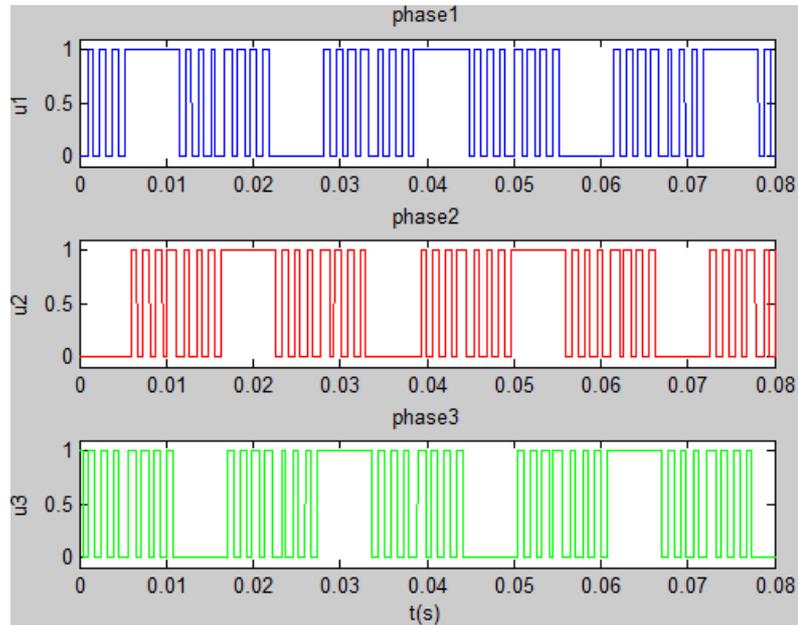
Les angles	im=0,01			im=0,04		
	Calculées par ANN-1	Exactes	L'erreur en °	Calculées Par ANN-1	Exactes	L'erreur en °
α_1	4,9933518	5	0,0066482	4,94045954	4,93594599	0,00451354
α_2	5,00329169	5	0,00329169	5,0108342	5,01047959	0,00035462
α_3	9,99340322	10	0,00659678	9,940345	9,93584362	0,00450138
α_4	10,0033184	10	0,00331837	10,018572	10,0193706	0,00079867
α_5	14,9934091	15	0,00659095	14,9397475	14,9352193	0,00452816
α_6	15,0037662	15	0,00376622	15,0255398	15,0271043	0,00156456
α_7	19,9933812	20	0,00661879	19,9389504	19,9343744	0,00457605
α_8	20,0042559	20	0,0042559	20,0317302	20,0339183	0,00218803
α_9	24,9933331	25	0,00666694	24,9381389	24,9335074	0,00463155
α_{10}	25,0047566	25	0,00475661	25,0372549	25,0399527	0,00269778
α_{11}	29,9932766	30	0,00672341	29,9374418	29,9327574	0,00468439
α_{12}	30,0052111	30	0,00521113	30,0421632	30,045294	0,00313081
α_{13}	34,9932222	35	0,00677784	34,9369528	34,9322262	0,00472655
α_{14}	35,0056087	35	0,00560869	35,0464979	35,0499977	0,0034998
α_{15}	39,9931788	40	0,00682115	39,9367424	39,9319908	0,0047516
α_{16}	40,0059464	40	0,00594643	40,0502886	40,0541009	0,00381223
α_{17}	44,9931547	45	0,00684529	44,9368649	44,9321107	0,00475416
α_{18}	45,0062249	45	0,00622493	45,0535568	45,05763	0,0040732
α_{19}	49,993157	50	0,006843	49,9373626	49,9326329	0,00472967
α_{20}	50,0064465	50	0,00644651	50,0563194	50,060606	0,00428661
α_{21}	54,9931923	55	0,00680769	54,9382688	54,9335947	0,00467413
α_{22}	55,0066146	55	0,00661459	55,0585918	55,0630476	0,00445581
α_{23}	59,9932666	60	0,00673336	59,9396102	59,9350263	0,00458394

Tableau 2.5 : L'erreur entre les valeurs calculées par ANN-1 et les valeurs exactes.

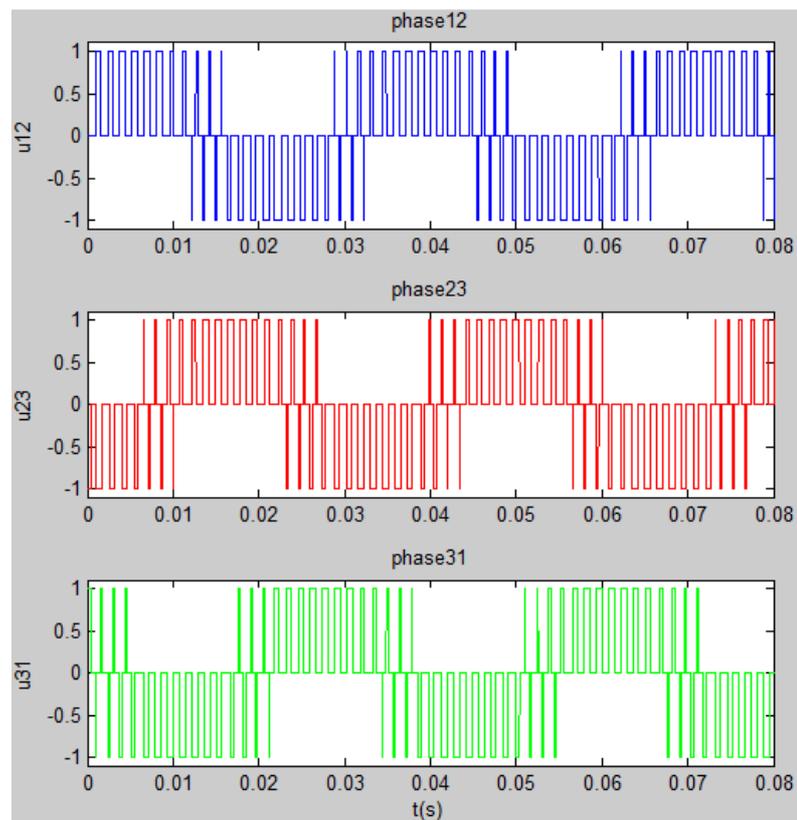
D'après le tableau 2.5, nous voyons clairement que les angles de commutation calculés par le réseau de neurones sont très proches des valeurs exactes vu que l'erreur ne dépasse pas l'ordre de 10^{-3} en degré, disant 10^{-6} dans l'échelle des secondes.

II.4.5. Simulation :

La simulation a été faite avec Matlab:



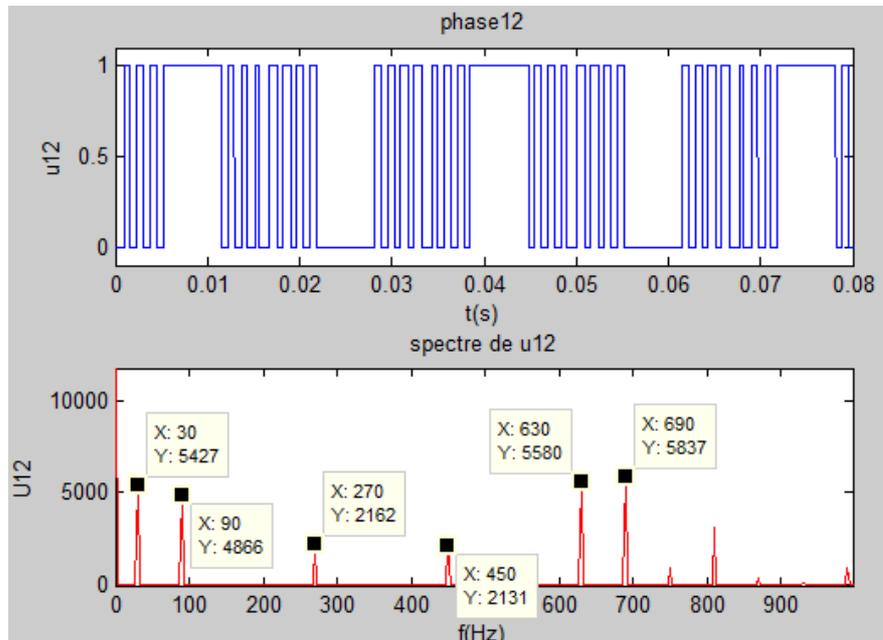
(a) Tensions des phase



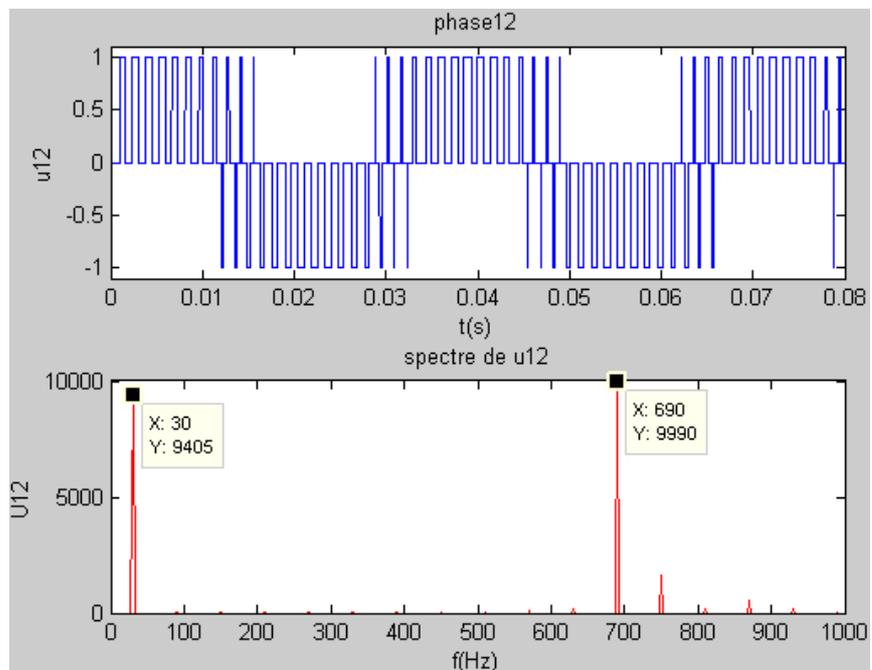
(b) Tension entre phases

Figure 2.9 : Signal MLI pour $im=0.6$, $m=7$.

La figure 2.9 montre que les tensions des phases obtenues sont périodiques avec une fréquence de 30 Hz ($m = 0,6$). En outre, on voit clairement l'existence de sept angles de commutation par un quart de période pour ($m = 7$). En plus les trois signaux de commande ANN SHE PWM sont déphasé par 120° .



(a) Tension simple



(b) Tension composée

Figure 2.10 : Le spectre de signal MLI pour $m=0.6$ et $m=7$.

La figure 2.10.a montre bien l'élimination des harmoniques d'ordre 5,7,11,13 et 17 par la commande MLI, et les harmoniques d'ordre paire bien que la figure 2.10.b montre l'élimination en plus des harmoniques d'ordre multiple de trois le premier harmonique apparu est d'ordre 23, d'où l'efficacité et la fiabilité de la commande.

II.5. Conclusion :

Dans ce chapitre nous avons vu deux méthodes différentes de calcul des angles de commutation pour un signal MLI destinés pour la commande des onduleurs triphasés pour un système de variation de vitesse d'un moteur asynchrone.

La première méthode c'est la MLI programmée ou de Patel-Hoft. Cette méthode a deux inconvénients inévitables. Le premier s'agit des équations non linéaires et transcendantes utilisées pour calculer les angles de commutation, ce qui augmente la complexité de son algorithme et le rend impossible de l'implémenter en temps réel sur un système embarqué. Le deuxième est la capacité de mémoire important nécessaire pour sauvegarder les valeurs des angles trouvées pour chaque valeur d'indice de modulation. Une mémoire en système embarqué demande la réalisation des différents blocs d'adressage et de traitement des bus.

La deuxième méthode est basée sur les réseaux de neurones. Elle ne présente aucun souci pour l'implémentation en temps réel ni pour les performances du système embarqué. Le seul inconvénient de cette méthode c'est le besoin d'une base de données pour pouvoir configurer les paramètres du réseau. Mais vu de la présence de cette dernière, nous avons pu la réaliser sur un circuit FPGA. La partie d'implémentation sera présentée dans le chapitre suivant.

Chapitre III

IMPLEMENTATION DE LA COMMANDE ANN SHE PWM SUR UNE CARTE FPGA.

III.1. Introduction

L'électronique moderne se tourne de plus en plus vers le numérique qui présente de nombreux avantages sur l'analogique : grande insensibilité aux parasites, reconfigurabilité, facilité de stockage de l'information etc...

Les circuits numériques nécessitent par contre une architecture plus lourde, et leur mode de traitement de l'information met en œuvre plus de fonctions élémentaires que l'analogique. Il en découle des temps de traitement plus long. Aussi les fabricants de circuits intégrés numériques s'attachent à fournir des circuits présentant des densités d'intégration toujours plus élevée, pour des vitesses de fonctionnement de plus en plus grandes [22].

D'abord réalisées avec des circuits SSI (**S**mall **S**cale **I**ntegration), les fonctions logiques intégrées se sont développées avec la mise au point du transistor MOS (**M**etal **O**xide **S**emiconductor) dont la facilité d'intégration a permis la réalisation de circuits MSI (**M**edium **S**cale **I**ntegration), puis LSI (**L**arge **S**cale **I**ntegration), puis VLSI (**V**ery **L**arge **S**cale **I**ntegration). Ces deux dernières générations ont vu l'avènement des microprocesseurs et microcontrôleurs.

Bien que ces derniers aient révolutionné l'électronique numérique par la possibilité de réaliser n'importe quelle fonction par programmation d'un composant générique, ils traitent l'information de manière séquentielle (du moins dans les versions classiques), ne répondant pas toujours aux exigences de rapidité.

Au début des années 70, les premiers composants entièrement configurables par programmation sont apparus en technologie bipolaire. La nouveauté résidait dans le fait qu'il était possible d'implanter physiquement par simple programmation, au sein du circuit, n'importe quelle fonction logique [22].

Le circuit n'est plus limité à un mode de traitement séquentielle de l'information comme avec les microprocesseurs. L'intégration des principales fonctions numériques d'une carte au sein d'un même boîtier permet de répondre à la fois aux critères de densité et de probabilité.

En cas d'erreur ils sont reprogrammables électriquement sans avoir à extraire le composant de son environnement. Parallèlement à ces circuits on trouve les ASIC qui sont des composantes où le concepteur intervient au niveau de dessin de la pastille de silicium en fournissant des masques à un fondeur. On ne peut plus franchement parler de circuits programmables. Les temps de développement long ne justifient l'utilisation que pour des grandes séries.

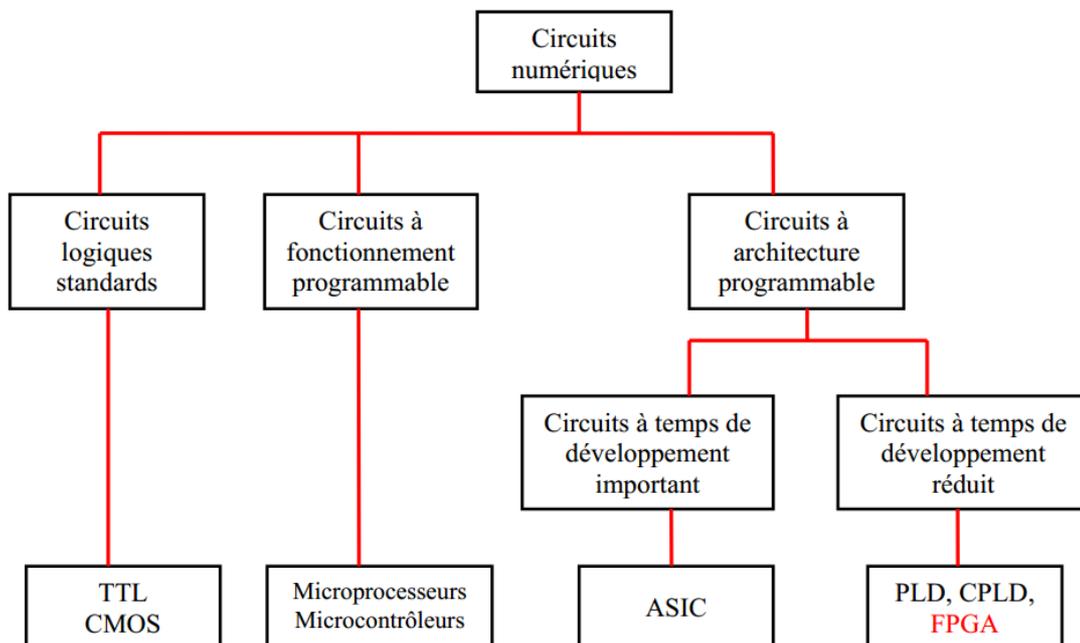


Figure 3.1 : Classification des circuits numériques.

III.2. Les circuits logiques programmables PLD [23]

Ce sont des circuits dont les fonctions peuvent être programmées pour une application particulière, après sa fabrication. Ils sont capables de réaliser plusieurs fonctions dans un seul circuit. Ce sont rapides flexibles à très haute densité d'intégration, à faibles cout de fabrication, de développement et de maintenance. On trouve des circuits programmables une seule fois et d'autres sont reprogrammables. Il y a plusieurs types de programmation telle que : EPROM, EEPROM, anti-fuse, SRAM. Ils implémentent n'importe quelle table de vérité d'une fonction logique.

Ces composants ne nécessitant aucune étape technologique supplémentaire pour être personnalisés. Nous y trouvons les circuits logiques programmables incluant un grand nombre

de solutions, toutes basées sur des variantes de l'architecture des portes ET, OU, nous distinguons parmi ces circuits : Les PLD, les EPLD, les EEPLD, les PAL, les PLA.

III.3. Les circuits FPGA

III.3.1. Définition :

Les FPGA (Field Programmable Gate Arrays ou "réseaux logiques programmables") sont des composants entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs.

L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court [22].

III.3.2. Application des FPGA:

Les FPGA sont utilisés dans de nombreuses applications, on en cite dans ce qui suit quelques-unes[14]:

- Prototypage de nouveaux circuits ;
- Fabrication de composants spéciaux en petite série ;
- Adaptation aux besoins rencontrés lors de l'utilisation ;
- Systèmes de commande en temps réel ;
- DSP (Digital Signal Processor) ;
- Imagerie médicale.

III.3.3. Architecture des FPGA [14] [24] :

Les circuits FPGA sont constitués d'une matrice de blocs logiques programmables entourés de blocs d'entrée sortie programmables. L'ensemble est relié par un réseau d'interconnexions programmables (figure3.2).

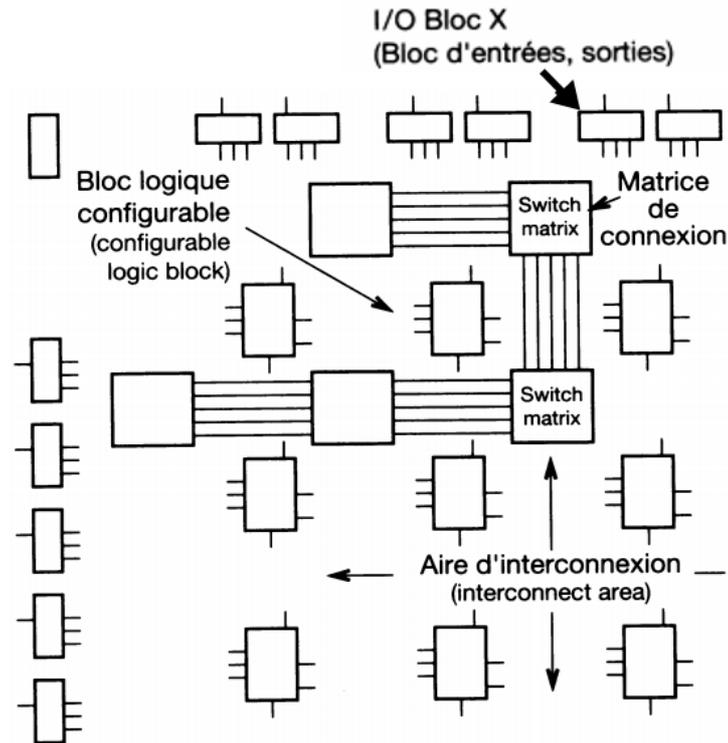


Figure 3.2 : Architecture interne du FPGA.

Il existe actuellement plusieurs fabricants de circuits FPGA, Xilinx et Altera sont les plus connus, et plusieurs technologies et principes organisationnels.

L'architecture, retenue par Xilinx, se présente sous forme de deux couches:

- une couche appelée circuit configurable.
- une couche réseau mémoire SRAM (Static Random Access Memory).

La couche dite « circuit configurable » est constituée d'une matrice de blocs logiques configurables (CLB) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs d'entrées/sorties (IOB) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs.

La programmation du circuit FPGA, appelé aussi LCA (Logic Cells Arrays), consistera en l'application d'un potentiel adéquat sur la grille de certains transistors à effet de champ servant à interconnecter les éléments des CLB et des IOB, afin de réaliser les fonctions souhaitées et d'assurer la propagation des signaux. Ces potentiels sont tout simplement mémorisés dans le réseau mémoire SRAM.

III.3.4. Réseau mémoire SRAM [14]:

La programmation d'un circuit FPGA est volatile, la configuration du circuit est donc mémorisée sur la couche réseau SRAM et stockée dans une ROM externe. Un dispositif interne permet à chaque mise sous tension de charger la SRAM interne à partir de la ROM. Ainsi on conçoit aisément qu'un même circuit puisse être exploité successivement avec des ROM différentes puisque sa programmation interne n'est jamais définitive.

On voit tout le parti que l'on peut tirer de cette souplesse en particulier lors d'une phase de mise au point. Une erreur n'est pas rédhibitoire, mais peut aisément être réparée.

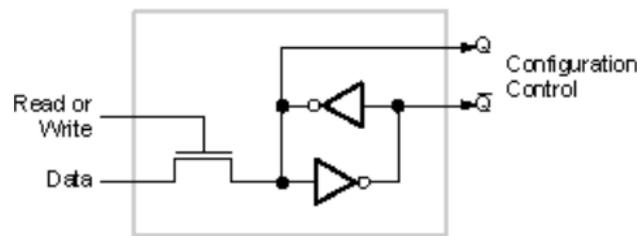


Figure 3.3 : Une cellule SRAM.

Les SRAM sont utilisé pour :

- Implémenter les tables de vérité LUT «les tables de vérité LUT «Look Up Table» des fonctions logiques
- Contrôler les interconnexions programmables avec -Mux - Portes ou transistors de transmission

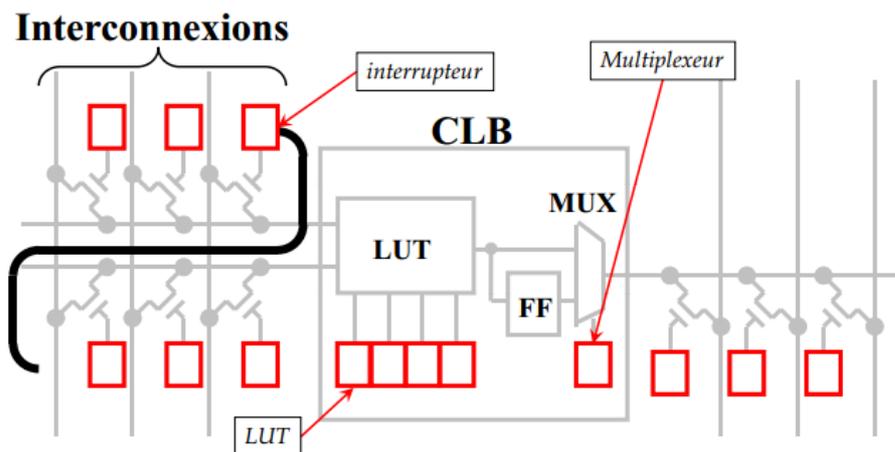


Figure 3.4 : Différentes utilisation de cellules SRAM.

La mise au point d'une configuration s'effectue en deux temps :

Une première étape purement logicielle va consister à dessiner puis simuler logiquement le circuit fini.

Dans la seconde étape, on effectuera une simulation matérielle en configurant un circuit réel.

III.3.5. Les CLB (Configurable Logic Bloc) :

Les blocs logiques configurables sont les éléments déterminants des performances du FPGA.

Chaque bloc est composé d'un bloc de logique combinatoire composé d'un ou plusieurs générateurs de fonctions réalisés avec des tables de correspondance (LUT look-up tables) qui peuvent mettre en œuvre une logique arbitraire en fonction de leur configuration. La taille des tables LUT est généralement de 16x1 (c'est à dire qu'elles disposent de 4 entrées) Autour d'une LUT, il y a une logique d'interconnexion qui permet les liaisons à destination et vers la LUT. Elle est mise en œuvre à l'aide de portes logiques et de multiplexeurs. Pendant le processus de configuration d'un FPGA, les mémoires des LUT sont écrites pour y implémenter une fonction, et la logique qui l'entoure est configurée pour router correctement les signaux afin de construire un système complexe.

La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter dans la cellule comme il est montré sur la figure suivante [25] :

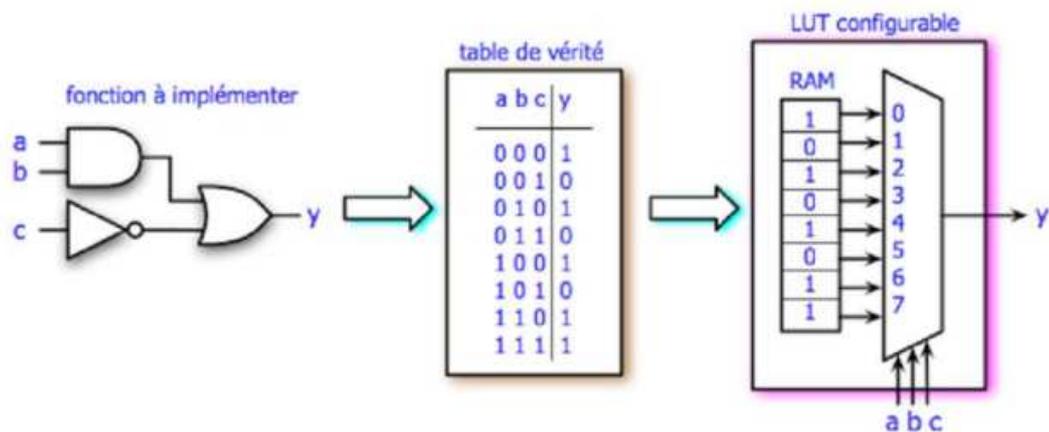


Figure 3.5 : Exemple de LUT.

Il existe différents types de CLB en fonction du FPGA utilisé. Pour Xilinx ils sont appelés slices. L'architecture du slice est utilisée dans toutes les familles **FPGA Virtex** et toutes les familles de **FPGA Spartan** [24].

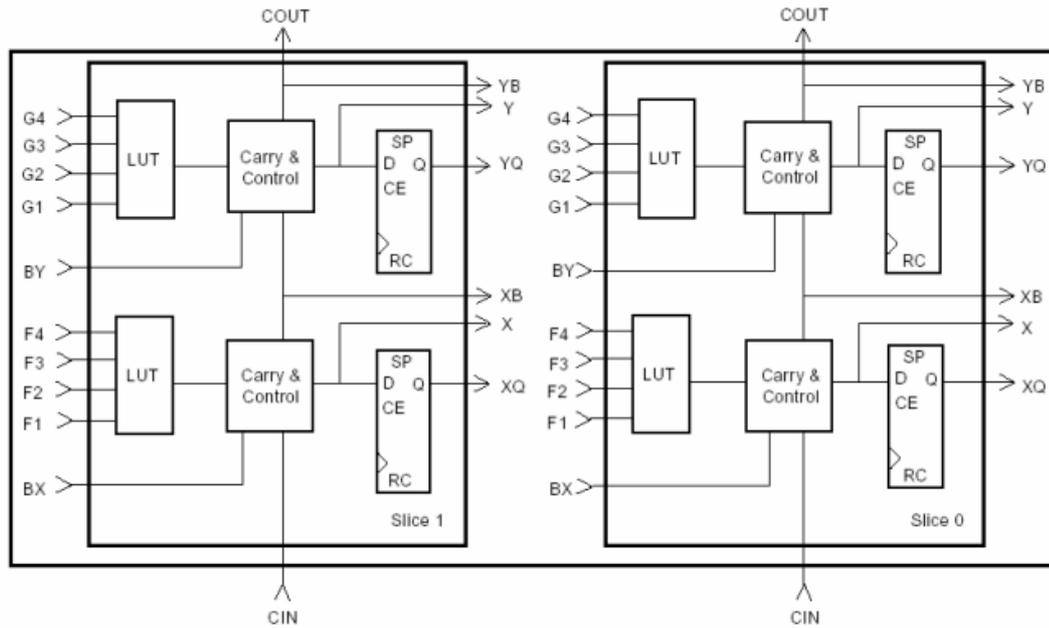


Figure 3.6 : Structure de base d'un composant programmable.

III.3.6. Les IOB (input output bloc) :

Les blocs entrée/sortie ou IOB permettent l'interface entre les broches du composant FPGA et la logique interne développée à l'intérieur du composant.

Ils sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé (haute impédance) [14].

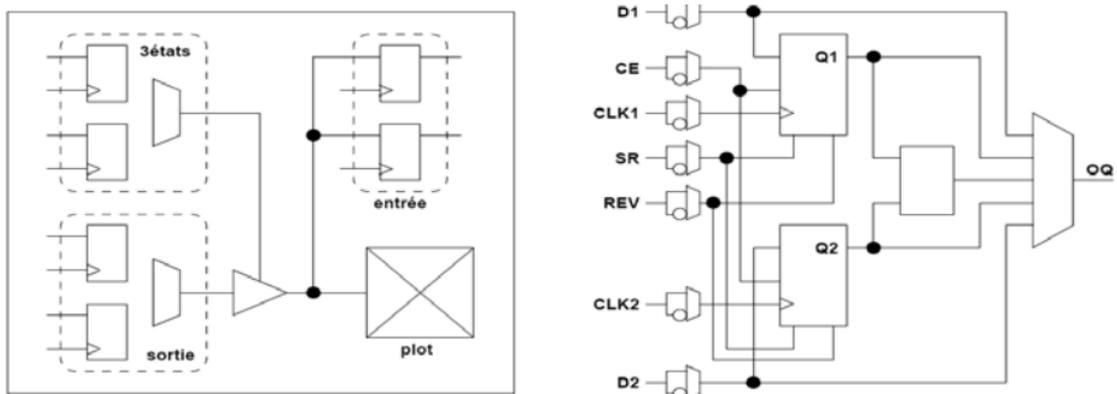


Figure 3.7 : Structure d'un bloc d'entrée/sortie.

III.3.7. Les interconnexions [26] :

Les connexions internes dans les circuits FPGA sont composées de segments métallisés.

Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement entre les divers CLB.

Elles permettent les connexions entre les diverses lignes, celles-ci sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM.

Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible.

Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons.

a- Les interconnexions à usage général

Elles servent à relier un CLB à n'importe quel autre CLB à l'aide des matrices de routage, évitant ainsi que les signaux ne traversent les grandes lignes et ne soient affaiblis.

b- Les interconnexions directes

Elles relient les CLB voisins directement avec un maximum d'efficacité en termes de vitesse et d'occupation du circuit, sans passer par des matrices d'interconnexion.

c- Les longues lignes

Elles traversent le FPGA dans toute sa longueur et sa largeur, et permettent de transmettre les signaux avec un minimum de retard entre les différents éléments dans le but d'assurer un synchronisme aussi parfait que possible. De plus, ces longues lignes permettent d'éviter la multiplicité des points d'interconnexion.

III.3.8. Les outils de développement des FPGA

Les outils de développement permettent au concepteur de programmer le circuit à partir de la description de la fonction à réaliser. Cette description peut être textuelle (VHDL, Verilog) ou graphique (chronogrammes, graphes d'état, symboles de fonction) [14].

III.4. Langage de description VHDL

III.4.1. Définition :

Le VHDL est un langage de spécification, de simulation et également de conception. Contrairement à d'autres langages (CUPL, ABEL) qui se trouvaient être en premier lieu des langages de conception, VHDL est d'abord un langage de spécification. La normalisation a d'abord eu lieu pour la spécification et la simulation (1987) et ensuite pour la synthèse (1993). Cette notion est relativement importante pour comprendre le fonctionnement du langage et son évolution. Grâce à la normalisation, on peut être certain qu'un système décrit en VHDL standard est lisible quel que soit le fabricant de circuits. Par contre, cela demande un effort important aux fabricants de circuits pour créer des compilateurs VHDL adaptés à et autant que possible optimisés pour leurs propres circuits [14].

III.4.2. Pour quoi un langage de description [27] :

L'électronicien a toujours utilisé des outils de description pour représenter des structures logiques ou analogiques. Le schéma structurel que l'on utilise depuis si longtemps et si souvent n'est en fait qu'un outil de description graphique. Aujourd'hui, l'électronique numérique est de plus en plus présente et tend bien souvent à remplacer les structures analogiques utilisées jusqu'à présent. Ainsi, l'ampleur des fonctions numériques à réaliser nous impose l'utilisation d'un autre outil de description. Il est en effet plus aisé de décrire un compteur ou un additionneur 64 bits en utilisant l'outil de description VHDL plutôt qu'un schéma. Le deuxième point fort du VHDL est d'être "un langage de description de haut niveau". En fait, un langage est dit de haut niveau lorsqu'il fait la plus possible abstraction de l'objet auquel ou pour lequel il est écrit. Dans le cas du langage VHDL, il n'est jamais fait référence au composant ou à la structure pour lesquels on l'utilise.

III.4.3. Spécification :

Etabli en premier lieu pour de la spécification. Il est tout-à-fait possible de décrire un circuit en un VHDL standard pour qu'il soit lisible de tous. Cette possibilité de décrire des circuits dans un langage universel est aussi très pratique pour éviter les problèmes de langue. De longues explications dans une langue peuvent ainsi être complétées par du code VHDL pour en faciliter la compréhension.

III.4.4. Simulation :

Le VHDL est également un langage de simulation. Pour ce faire, la notion de temps, sous différentes formes, y a été introduite. Des modules, destinés uniquement à la simulation,

peuvent ainsi être créés et utilisés pour valider un fonctionnement logique ou temporel du code VHDL.

La possibilité de simuler avec des programmes VHDL devrait considérablement faciliter l'écriture de tests avant la programmation du circuit et éviter ainsi de nombreux essais sur un prototype qui sont beaucoup plus coûteux et dont les erreurs sont plus difficiles à trouver. Bien que la simulation offre de grandes facilités de test, il est toujours nécessaire de concevoir les circuits en vue des tests de fabrication, c'est-à-dire en permettant l'accès à certains signaux internes.

III.4.5. Conception :

Le VHDL permet la conception de circuits avec une grande quantité de portes.

L'avantage d'un langage tel que celui-ci par rapport aux langages précédents de conception matérielle est comparable à l'avantage d'un langage informatique de haut niveau (Pascal, ADA, C) vis-à-vis de l'assembleur. Ce qui veut aussi dire que malgré l'évolution fulgurante de la taille des circuits, la longueur du code VHDL n'a pas suivi la même courbe.

Le VHDL, bien que facilement accessible dans ses bases, peut devenir extrêmement compliqué s'il s'agit d'optimiser le code pour une architecture de circuit. C'est pour cette raison que de plus en plus de fabricants offrent des macros, gratuites pour les fonctions sans grandes difficultés et payantes pour les autres.

III.4.6. Structure d'une description VHDL simple :

La structure typique d'une description VHDL est composée de 2 parties indissociables qui sont : l'entité et l'architecture.

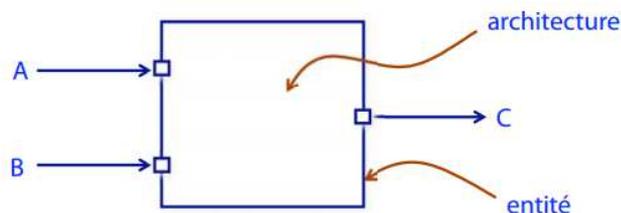


Figure 3.8 : Schéma montrant structure d'un code VHDL.

- **L'entité** est vue comme une boîte noire avec des entrées et des sorties caractérisée par des paramètres. Elle représente une vue externe de la description.
- **L'architecture** contient les instructions VHDL permettant de décrire et de réaliser le fonctionnement attendu. Elle représente la structure interne de la description. Les entités et

architectures sont des unités de conception dites primaire et secondaire. Un couple entité-architecture donne une description complète d'un élément ; ce couple est appelé modèle.

III.4.6.1 Déclaration des bibliothèques

Toute description VHDL utilisée pour la synthèse a besoin de bibliothèques.

L'IEEE les a normalisées et plus particulièrement la bibliothèque IEEE1164.

Elles contiennent les définitions des types de signaux électroniques, des fonctions et sous programmes permettant de réaliser des opérations arithmétiques et logiques.

La directive **use** permet de sélectionner les bibliothèques à utiliser.

On peut mettre dans un package des composants et des fonctions qui pourront être utilisés par la suite, par d'autres programmes [14]

III.4.6.2 Déclaration de l'entité et des entrées/sorties :

Cette opération permet de définir le nom de la description VHDL ainsi que les entrées et sorties utilisées, l'instruction qui les définit est **port**

Le nom du signal est composé d'une chaîne de caractères dont le premier est une lettre. Le langage VHDL n'est pas sensible à la « casse », c'est à dire qu'il ne fait pas la distinction entre les majuscules et les minuscules. Le signal peut être défini en entrée (**in**), en sortie (**out**), en entrée/sortie (**inout**) ou en **buffer**, c'est-à-dire qu'il est en sortie mais il peut être utilisé comme entrée à l'intérieur de la description. L'affectation des broches d'entrées/sorties se fait par la définition d'attributs supplémentaires qui dépendent du logiciel de développement utilisé.

III.4.7. Les deux modes de travail en VHDL :

Le VHDL utilise deux modes de fonctionnement : le mode combinatoire (ou concurrent) et le mode séquentiel. Chacun de ces modes est utilisé dans des cas bien précis.

III.4.7.1. Le mode combinatoire :

En mode combinatoire (ou concurrent), toutes les instructions d'une description VHDL sont évaluées et affectent les signaux de sortie en même temps, l'ordre dans lequel les instructions sont écrites n'a donc aucune importance. En effet la description génère des structures électroniques, c'est la grande différence entre une description VHDL et un langage informatique classique.

Dans un système à microprocesseur, les instructions sont exécutées les unes à la suite des autres. Avec VHDL il faut essayer de penser à la structure qui va être générée par le synthétiseur pour écrire une bonne description, cela n'est pas toujours évident.

III.4.7.2. Le mode séquentiel :

Le mode séquentiel utilise les process dans lesquels le temps est une variable essentielle.

Un process est une partie de la description d'un circuit dans laquelle les instructions sont exécutées séquentiellement, c'est-à-dire les unes à la suite des autres. Il permet d'effectuer des opérations sur les signaux en utilisant les instructions standards de la programmation structurée comme dans les systèmes à microprocesseurs. Dans le mode séquentiel, on distingue :

- **Le mode séquentiel asynchrone**, dans lequel les changements d'état des sorties peuvent se produire à des instants difficilement prédictibles (retards formés par le basculement d'un nombre souvent inconnu de fonctions),
- **Le mode séquentiel synchrone**, dans lequel les changements d'état des sorties se produisent tous à des instants quasiment connus (un temps de retard après un front actif de l'horloge).

III.5. Etapes nécessaires au développement d'un projet sur FPGA :

Le développement en VHDL nécessite l'utilisation de deux outils : le simulateur et le synthétiseur. Le premier va nous permettre de simuler notre description VHDL avec un fichier de simulation appelé « test-bench »; Cet outil interprète directement le langage VHDL et il comprend l'ensemble du langage. L'objectif du synthétiseur est très différent : il doit traduire le comportement décrit en VHDL en fonctions logiques de bases, celles-ci dépendent de la technologie choisie ; Cette étape est nommée « synthèse ». L'intégration finale dans le circuit cible est réalisée par l'outil de placement et routage. Celui-ci est fourni par le fabricant de la technologie choisie [14].

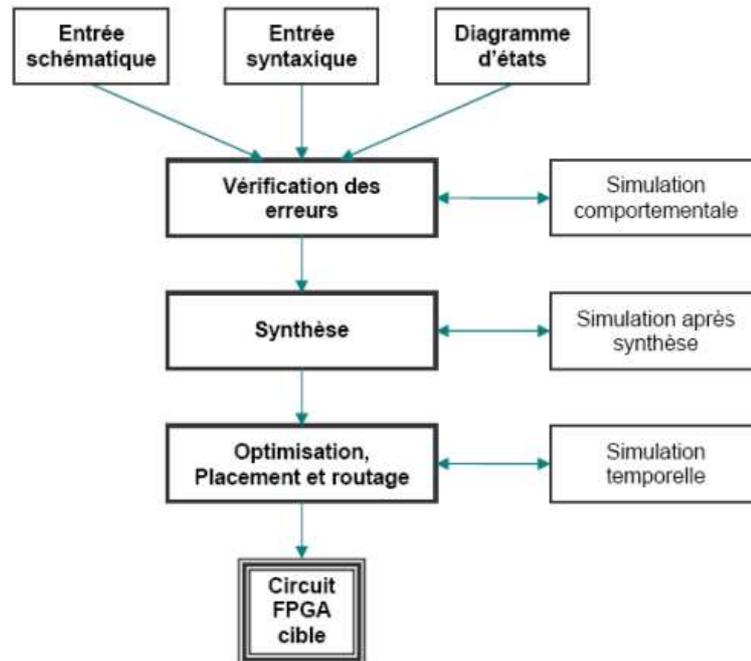


Figure 3.9 : Organisation fonctionnelle de développement d'un projet sur circuit FPGA.

III.5.1. Saisie du texte VHDL :

La saisie du texte VHDL se fait sur le logiciel « ISE Xilinx Project Navigator ». Ce logiciel propose une palette d'outils permettant d'effectuer toutes les étapes nécessaires au développement d'un projet sur circuit FPGA. Il faut commencer par créer un projet, ensuite inclure des fichiers sources dans lesquels il faut saisir le texte VHDL désiré. On peut inclure autant de sources qu'on veut dans un projet.

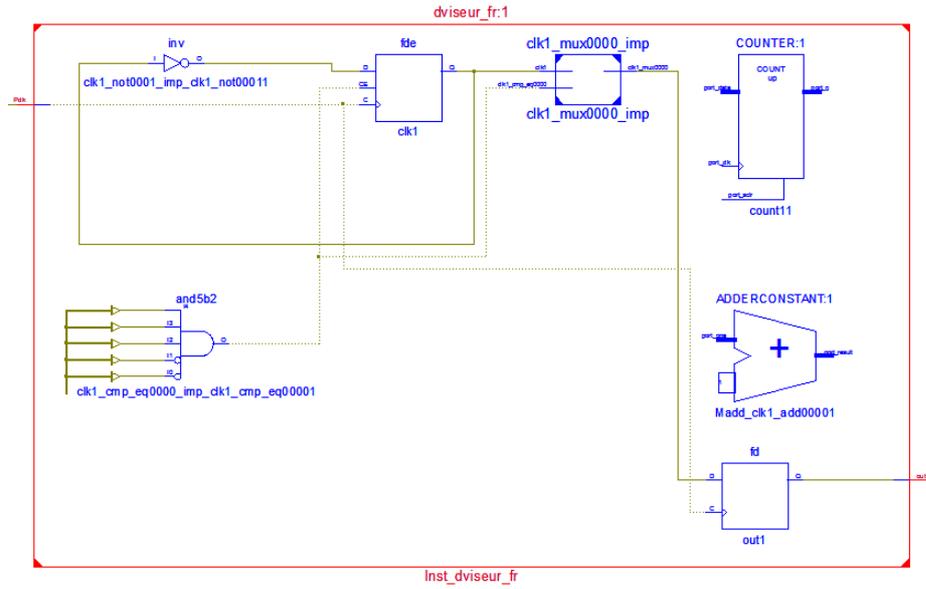
III.5.2. Vérification des erreurs :

Cette étape est effectuée en appuyant sur le bouton « check syntax ».

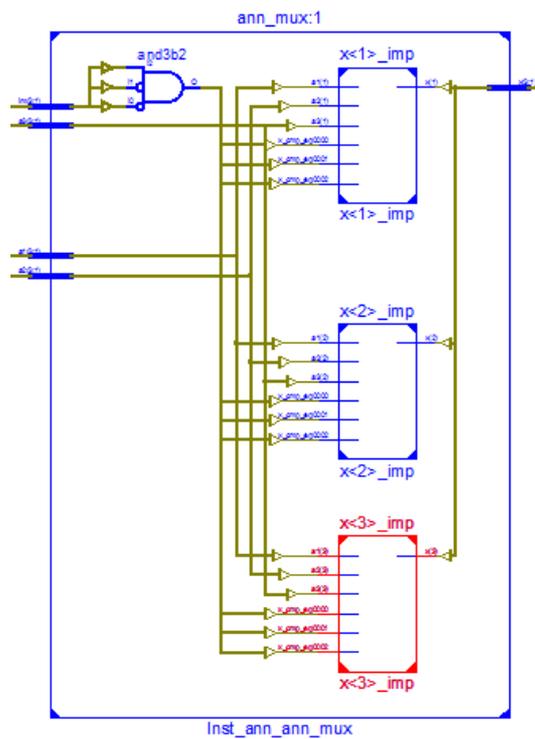
Elle permet de vérifier les erreurs de syntaxe du texte VHDL et d'afficher les différentes alarmes liées au programme. Cette étape permet donc de valider la syntaxe du programme et de générer la « netlist », qui est un fichier contenant la description de l'application sous forme d'équations logiques.

III.5.3. Synthèse

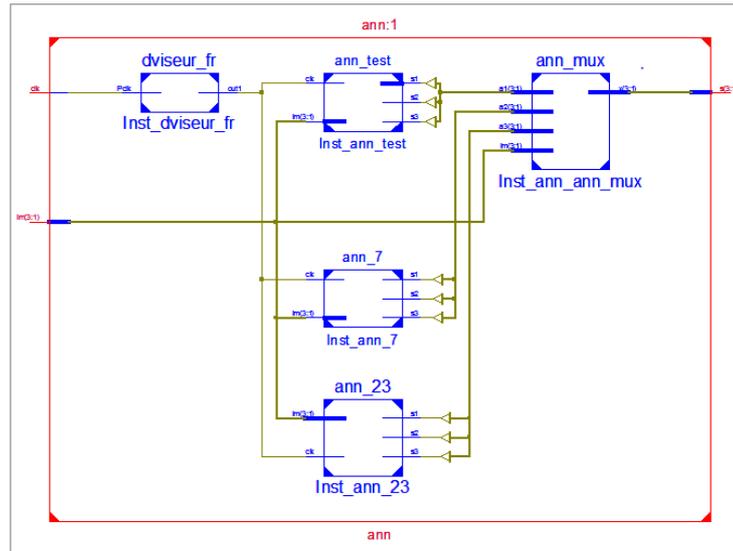
Le synthétiseur a pour rôle de convertir le projet, en fonction du type du circuit FPGA cible utilisé, en portes logiques et bascules de base. L'outil «View RTL Schematic» permet de visualiser les schémas électroniques générés par le synthétiseur (figure 3.11).



(a) Schéma de diviseur de fréquence



(b) Schéma de bloc de multiplexage



(c) Schéma électronique équivalent global

Figure 3.10 : Quelques schémas électroniques générés par le synthétiseur, visualisés par l'outil «RTL Schematic Viewer».

De plus, le synthétiseur permet à l'utilisateur d'imposer des contraintes de technologie: par exemple fixer la vitesse de fonctionnement, délimiter la zone du circuit FPGA dans laquelle le routage doit se faire ou affecter les broches d'entrées/sorties.

III.5.4.Simulation :

La simulation permet de vérifier le comportement d'un design avant ou après implémentation dans le composant cible. Elle représente une étape essentielle qui nous fera gagner du temps lors de la mise au point sur la carte. Il faut juste noter qu'un projet peut être simulé même s'il n'est pas synthétisable. Lors de l'étape de simulation comportementale, on valide l'application indépendamment de l'architecture et des temps de propagation du futur circuit cible. La phase de simulation après synthèse valide l'application sur l'architecture du circuit cible, et enfin la simulation temporelle prend en compte les temps de propagation des signaux à l'intérieur du circuit FPGA cible.

III.5.5. Optimisation, placement et routage :

Pendant l'étape d'optimisation, l'outil cherche à minimiser les temps de propagation et à occuper le moins d'espace possible sur le circuit FPGA cible. Le placement et routage permet de tracer les routes à suivre sur le circuit afin de réaliser le fonctionnement attendu.

III.5.6. Programmation du composant et test :

Dans cette dernière étape (Generate Programming Files), on génère le fichier à charger sur le circuit FPGA à travers l'interface JTAG. Une fois le programme chargé sur le circuit, on peut tester et visualiser les résultats directement sur la carte de développement.

III.6. Implémentation de l'algorithme de commande sur FPGA:

III.6.1. Description de l'algorithme :

Le programme comporte cinq modules :

- 1- Sélection du réseau de neurone adapté au im tab 2.1
- 2- Attribution de poids et de biais: cette partie du programme attribue pour chaque réseau neuronal, précédemment sélectionné, les paramètres appropriés (des poids et biais)
- 3- Calcul des instants de commutation: à cette étape, le programme calcule les instants de commutation à l'aide de l'im d'entrée, l'horloge clk et les paramètres ANN calculées dans la partie précédente.
- 4- Diviseur de fréquence: ce module est un diviseur de fréquence qui divise la fréquence d'entrée par 50; par conséquent, la fréquence de sortie est de 1 MHz.
- 5- Génération de signaux de commande MLI: ce module génère les trois signaux de commande MLI déphasé par 120° en utilisant l'horloge clk et les angles de commutations calculés précédemment.

La figure 3.12 représente le diagramme de notre algorithme. im est l'indice de modulation et clk est l'horloge d'entrée, dans le programme $clk = 50$ MHz.

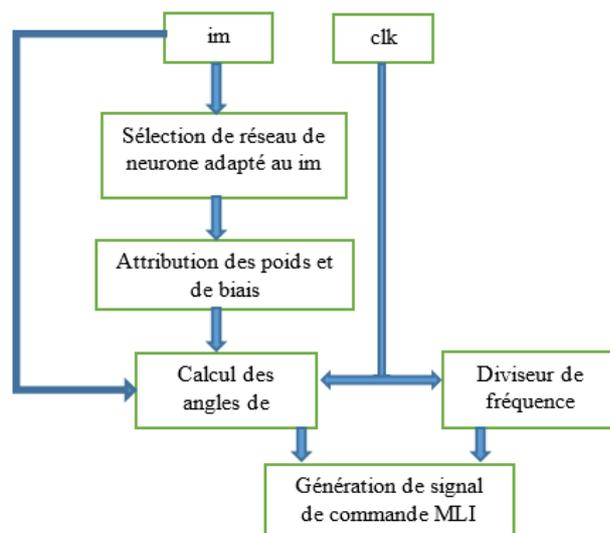
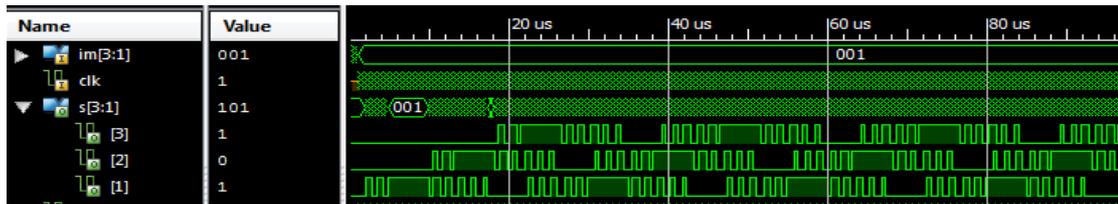


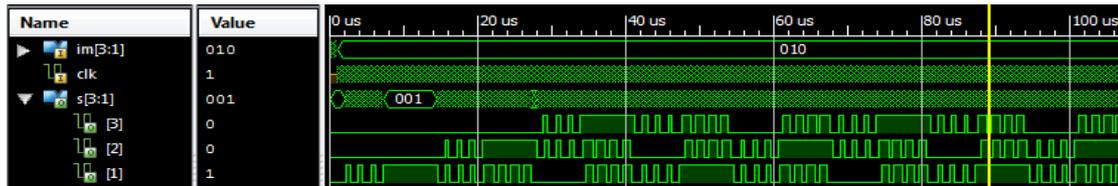
Figure 3.11 : Organigramme de l'algorithme de commande.

III.6.2. Simulation

On a utilisé le simulateur Isim de xilinx. La fréquence d'horloge est 50 MHz



(a) $im=0.8$: $f = 40$ Hz .



(b) $im=0.5$: $f = 25$ Hz .

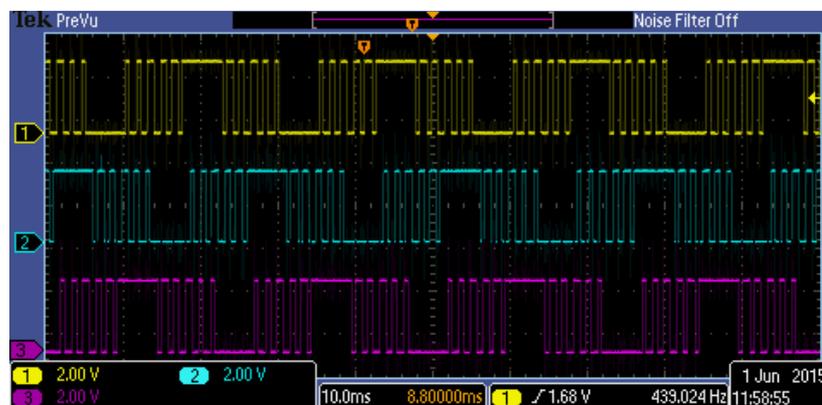


(c) $im=0.1$: $f = 5$ Hz .

Figure 3.12 : Résultat de simulation des signaux des sorties pour les trois phases.

III.6.3. Test de signaux de sortie :

Pour vérifier le fonctionnement de la commande ANN SHE PWM, on a fait le test des signaux sur l'oscilloscope. La figure 3.14 montre les signaux triphasés pour les trois valeurs de l'indice de modulation im .



(a) $im=0.8$: $f = 40$ Hz.

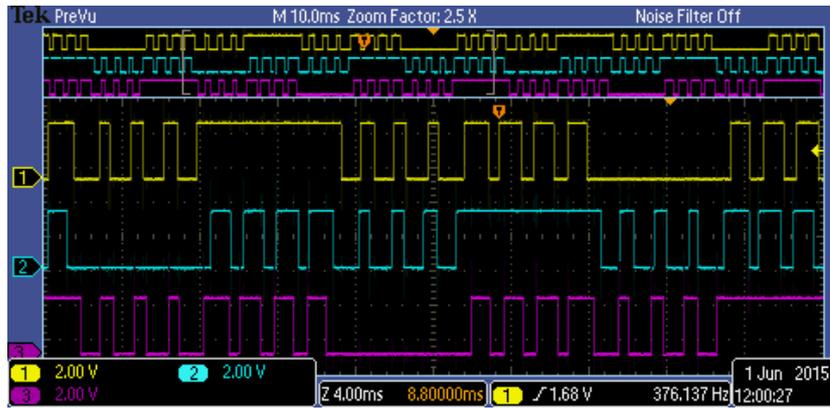
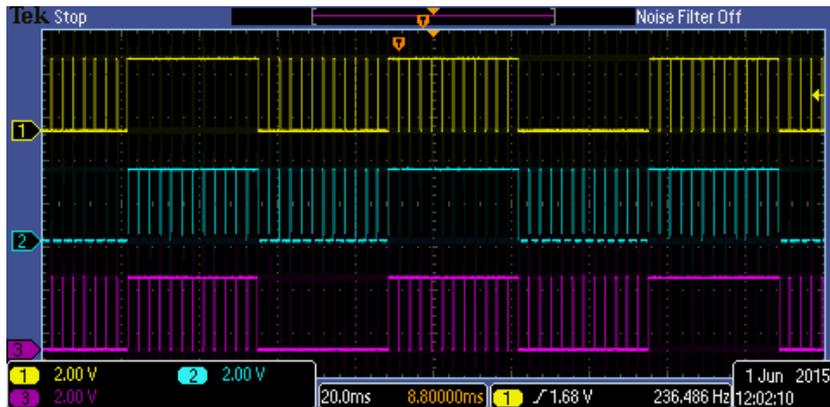
(b) $im=0.5 : f = 25 \text{ Hz}$.(c) $im=0.1 : f = 5 \text{ Hz}$

Figure 3.13: Visualisation des signaux ANN SHE PWM.

III.6.4. Interprétation :

La figure 3.14 montre que les tensions de phase obtenues sont périodiques avec une fréquence de 40 Hz ($im = 0,8$), 25 Hz ($im = 0,5$) et 5 Hz ($im = 0,1$). En outre, on voit clairement l'existence de cinq, sept et vingt-trois angles de commutation par un quart de période pour $m = 5$, 7 et $m = 23$ respectivement. En plus les trois signaux de commande ANN PWM SHE sont déphasé par 120° .

Un autre avantage de l'algorithme proposé est qu'il permet l'économie d'une quantité considérable l'espace mémoire. En effet, pour les six réseaux de neurones, l'espace mémoire nécessaire pour enregistrer tous les paramètres (les poids et les biais) est 156 emplacement mémoire. Toutefois, l'espace mémoire nécessaire pour sauvegarder tous les angles de commutation de la base de données est 1020 (cinq fois plus).

En outre, l'utilisation des six réseaux de neurones implique que les angles de commutation sont calculés et générés en temps réel pour une valeur de l'indice de modulation (im), tandis

que l'utilisation d'une base de données comme des tables de consultation où seules certaines valeurs discrètes sont mémorisées, rend impossible l'obtention des angles de commutation correspondant à des valeurs intermédiaires non mémorisées de l'indice de modulation.

III.7. Conclusion :

Dans ce chapitre, la commande de tension et de fréquence d'un onduleur triphasé conçu pour les véhicules électriques à base de moteur à induction est obtenue en utilisant un nouvel algorithme de MLI on-ligne basé sur le principe de Patel-Hoft et en faisant appel aux réseaux de neurones artificiels. Les résultats de simulation obtenus montrent que les angles de commutation calculés par l'algorithme ANN SHE PWM proposé sont très proches de celles calculées par la méthode itérative de Newton-Raphson. Les résultats expérimentaux obtenus (signaux de sortie de FPGA) montrent ainsi que le calcul de l'angle de commutation et la génération des signaux PWM sont effectuées en temps réel et avec une très bonne précision.

Chapitre IV

REALISATION D'UN VARIATEUR DE VITESSE PILOTE PAR UNE COMMANDE MLI CALCULEE PAR UN RESEAU DE NEURONES

IV.1. Introduction :

Après avoir acquis toutes les connaissances et les éléments nécessaires pour réaliser un système de variation de vitesse pour un moteur asynchrone triphasé, commandé par un onduleur de tension et piloté par une carte FPGA « Spartan 3E » en utilisant une commande MLI programmée, nous passons à l'étape de réalisation et mettre en œuvre ce système. Cette étape doit comporter les études établies pour la conception des blocs du système ainsi que le dimensionnement des grandeurs et le choix de différentes composantes utilisées.

Dans ce chapitre, on va voir une description globale et détaillée des deux étages réalisés, l'étage de commande et l'étage de puissance. A la fin, nous établirons une interprétation des résultats expérimentaux au repère des résultats théoriques.

IV.2. Principaux blocs du système :

Le système comporte deux principaux étages : l'étage de commande et l'étage de puissance.

- L'étage de commande :

Il s'agit de la partie responsable de calcul et génération des signaux MLI. Il se compose de trois éléments : LA carte de développement FPGA, le circuit d'isolation galvanique, le circuit de commande approché des interrupteurs. Cet étage doit également assurer la protection et l'isolation de la carte FPGA de reste de la partie de puissance.

- L'étage de puissance :

Il s'agit d'un onduleur de tension triphasé à deux étages, alimenté par une tension continue. Les trois tensions issues de l'onduleur alimentent directement le moteur triphasé

Le schéma bloc de système est illustré sur la figure 4.1.

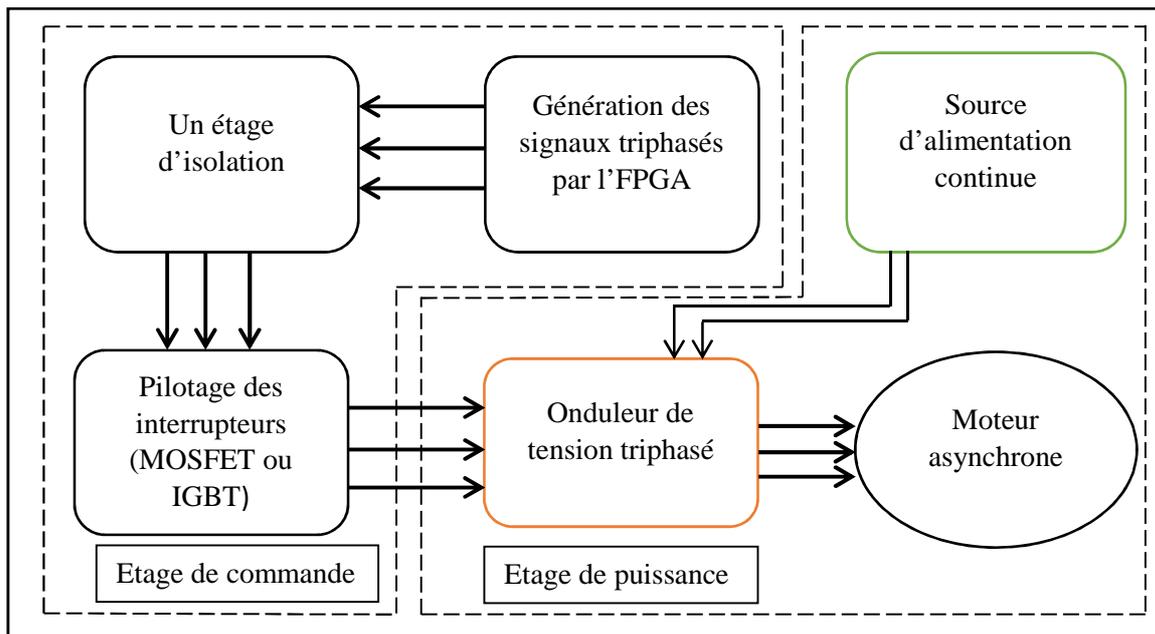


Figure 4.1 : Le schéma bloc de variateur de vitesse.

IV.2.1. L'étage de puissance :

Comme on a vu dans le premier chapitre, l'onduleur est composé de trois bras parallèles (Figure 1.13). Chaque bras contient deux interrupteurs en série commandés par l'étage de commande. Ces interrupteurs peuvent être des IGBT ou bien des MOSFET. Le choix de ces interrupteurs se fait selon la charge maximale supportée et la fréquence maximale de commutation. Le courant d'alimentation de moteur est de 3 A par phase. Le signal MLI généré par l'FPGA nécessite une rapidité de commutation qui peut atteindre 18 kHz. En effet :

La fréquence de commutation f_c est donnée par :

$$f_c = \frac{1}{T_c} = i_m \frac{360^\circ \cdot f_0}{\alpha_{\min}} \quad (4.1)$$

Où : f_0 est la fréquence nominale de courant d'alimentation de moteur $f_0 = 50\text{Hz}$.

L'angle minimal est $\alpha_{\min} = 0.01^\circ$. Cela correspond à un indice de modulation $i_m = 0.01$

On trouve $f_{c_{\max}} = 18\text{kHz}$.

IV.2.1.1. Choix de type de transistor :

Le choix est entre l'utilisation des MOSFET ou bien des IGBT.

IV.2.1.1.1. Transistor MOSFET :

Le transistor MOSFET est un composant plutôt adapté aux applications de faible ou moyenne puissance inférieure à quelques dizaines de kilowatts. En effet, c'est un dispositif à effet de champ dont la résistance apparente en conduction croît fortement avec la tenue en tension directe. Il en résulte que, pour obtenir des composants industriellement viables, la tenue en tension ne dépasse pas 1 000 V (avec quelques ampères de courant commutable) et les courants admissibles sont de l'ordre de quelques centaines d'ampères (pour des tenues en tension de 50 à 200 V). Parallèlement, l'absence de mécanisme bipolaire en fait un composant intrinsèquement très rapide en commutation, pour peu que sa commande de grille soit performante. Cette dernière est, en statique, une commande en tension (impédance d'entrée très élevée), ce qui est très intéressant sur le plan de la mise en œuvre. Il est souvent utilisé dans des applications à basse tension d'alimentation, avec des fréquences de découpage qui peuvent éventuellement dépasser 100 kHz [28].

IV.2.1.1.2. Transistor IGBT :

Le transistor bipolaire à grille isolée (IGBT) combine un transistor bipolaire PNP et transistor MOSFET, profitant ainsi des avantages de chacun, respectivement faible perte en conduction et rapidité en commutation, en conservant la simplicité de commande en tension (par une grille isolée) du MOSFET. Les IGBT modernes présentent des performances en commutation tout à fait honorables et permettent de l'utiliser à des fréquences de découpage comprises entre quelques kilohertz et une centaine de kilohertz. Avec une gamme de puissance qui s'étend maintenant de quelques kilowatts à quelques mégawatts [28].

Nous avons choisi d'utiliser des MOSFET du fait de sa rapidité de commutation par rapport aux IGBT. De plus, les IGBT sont consacrés surtout pour les applications de forte puissance. Pour les applications de puissance moyenne, les MOSFET sont largement suffisants. Présentons brièvement les caractéristiques générales des MOSFET.

IV.2.1.2. Les caractéristiques générales des MOSFET:

IV.2.1.2.1. Principe des MOSFET:

Le transistor à effet de champ à grille isolée plus couramment nommé MOSFET a été conçu de façon théorique en 1920 par Julius Edgar Lilienfeld. Le MOSFET présente la particularité d'être un interrupteur de puissance commandé qui fonctionne en mode unipolaire, c'est-à-dire que le courant n'est dû qu'au déplacement d'un seul type de porteurs de charges. Il en résulte une très grande rapidité de commutation due à l'absence de charge stockée. En revanche, la chute de tension est relativement élevée à l'état passant. Ce qui est dû à l'absence de la zone de plasma dans la zone N [29].

La (Figure 4.2) montre la structure représentative d'un MOSFET. Lorsqu'on applique une tension V_{gs} entre la grille et la source supérieure à une tension de seuil V_{th} , un canal est créé et relie les deux zones N^+ et N^- . Cela permet la circulation des électrons de la source dopée N^+ vers le drain. Le MOSFET entre donc en conduction. Pour le bloquer, il faut seulement supprimer le canal en ramenant la tension V_{gs} entre la grille et la source à une tension inférieure à la valeur de tension de seuil V_{th} [30].

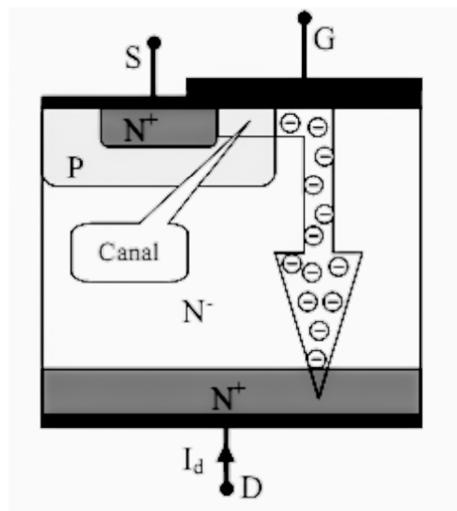


Figure 4.2 : Structure de MOSFET.

IV.2.1.2.2. Commande rapprochée des transistors MOSFET:

La commande rapprochée des MOSFET est un étage responsable à l'adaptation de signal de commande éloignée (généré par l'FPGA) au mode de fonctionnement des MOSFET (Figure 4.3). Elle doit assurer :

- Une isolation galvanique au niveau de signal de commande et assurer une alimentation isolée pour apporter l'énergie nécessaire à la commutation.
- Une tension de grille suffisamment grande pour avoir les commutations.
- Un bon fonctionnement pour n'importe quelle valeur d'indice de modulation.

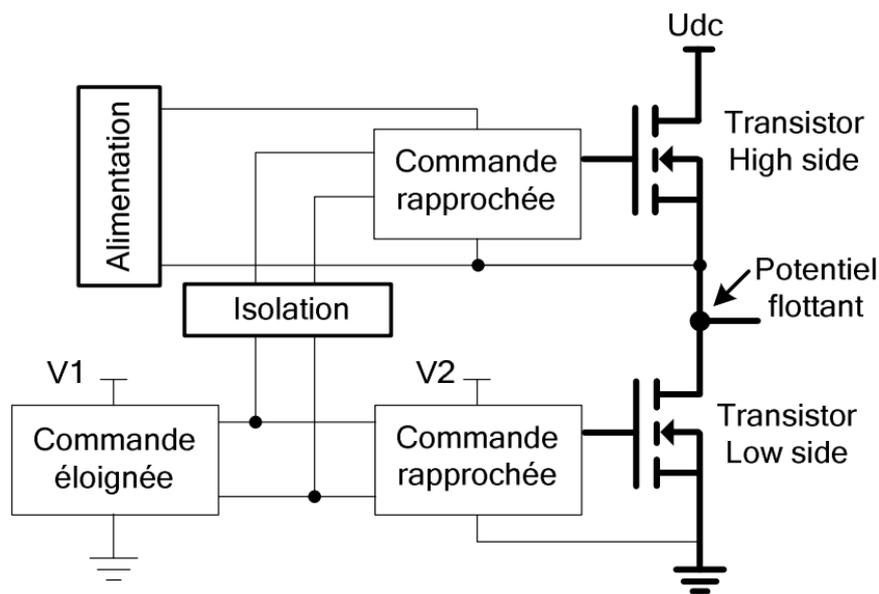


Figure 4.3 : Transistor High-side et Low-side du bras de l'onduleur.

La commande du MOSFET « high-side » est plus complexe que celle du MOSFET « low-side ». En effet, le MOSFET « high-side » est à potentiel flottant, alors le signal de commande éloignée doit être référencé à la source du MOSFET « high-side » pour garantir la bonne polarisation de l'électrode de grille par rapport à celle de la source quel que soit le potentiel au point du milieu. Pour éviter la modification de signal de commande éloignée. Nous allons utiliser une des nombreuses solutions qui ont été proposées, elle s'agit de la technique de bootstrap.

- **Technique de bootstrap :**

La technique de bootstrap permet de créer une alimentation flottante pour le circuit de commande de l'interrupteur « high side », à partir de l'alimentation du circuit de commande du transistor « low side » [31].

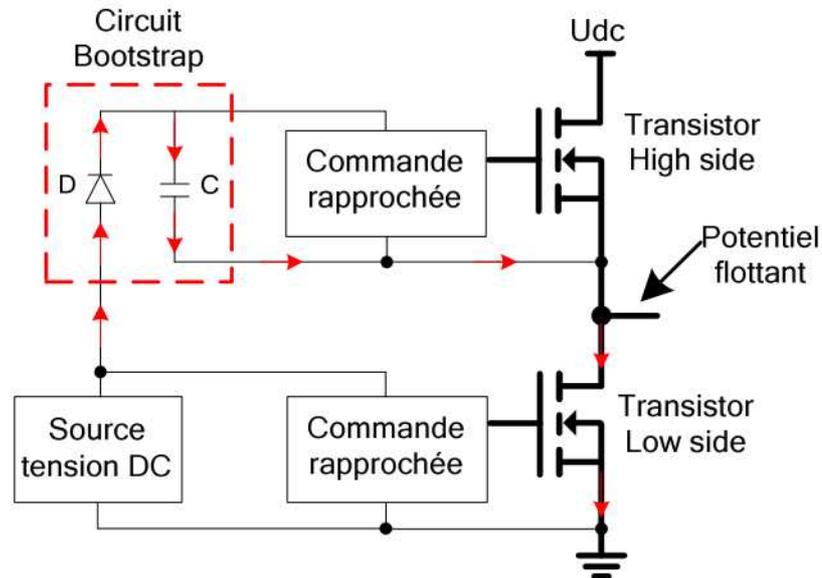


Figure 4.4 : Principe de la technique de bootstrap.

Le principe de cette technique c'est de charger le condensateur C jusqu'à la tension d'alimentation pendant la conduction de MOSFET « low-side ».

Lorsque l'interrupteur « low-side » se bloque et que la tension réapparaît à ses bornes, la diode D se bloque à son tour, et elle doit donc être capable de supporter la haute tension U_{dc}. La charge alors portée par C sert à l'alimentation du circuit de commande de l'interrupteur « high-side », à charger sa capacité d'entrée et à la maintenir chargée pendant toute la durée de conduction désirée [32].

La durée de conduction du transistor « low-side » doit être suffisante pour que le condensateur de bootstrap puisse se charger totalement. De la même façon, la durée de conduction du transistor « high-side » est limitée pour éviter la décharge complète du condensateur de bootstrap, compte tenu de la consommation du circuit de commande; pour la même raison, la fréquence de découpage ne peut être trop basse. Le dimensionnement de la capacité du condensateur de bootstrap est imposé par ces deux contraintes. Pour des raisons technologiques, elle n'est généralement pas intégrée dans les circuits de commande et doit donc être rajoutée, de même que la diode de bootstrap dans de nombreux cas [32].

IV.2.1.3. Conception de l'étage de puissance :

Nous avons utilisé des MOSFET de référence IRFP460. Avec une tension maximale supportable de 500V, un courant jusqu'à 20 A, une tension seuil de grille $V_{th} = 4V$ et une fréquence maximale de commutation de 150 kHz

IV.2.1.3.1. Calcul de la puissance dissipée. :

La puissance dissipée P_D se calcule par la formule suivante [34]:

$$P_D = P_{D_conduction} + P_{D_commutatin}$$

$$P_D = R_{DSS} \cdot I_{DSS}^2 \cdot D + C_{IN} \cdot V_{GS}^2 \cdot f_c \quad (4.3)$$

Où : C_{IN} : La capacité d'entrée du MOSFET (entre G et S).

R_{DSS} : Résistance Drain-Source en régime statique.

I_{DSS} : le courant drain-source.

D est le rapport cyclique de commutation. Il est variable mais on peut le prendre maximum pour précaution.

IV.2.1.3.2. Dimensionnement des radiateurs pour les MOSFET :

On a :

$$T_j - T_a = (R_{th(j-c)} + R_{th(c-r)} + R_{th(r-a)}) \cdot P_D \quad (4.1)$$

Où : $R_{th(j-c)}$: est la résistance thermique entre la jonction et le boîtier.

$R_{th(c-r)}$: est la résistance thermique entre le boîtier et le radiateur.

$R_{th(r-a)}$: est la résistance thermique radiateur-ambiante.

T_j, T_a : sont respectivement la température de jonction et la température ambiante.

On peut calculer la résistance thermique totale par :

$$R_{th} = \frac{T_j - T_a}{P_D} \quad (4.2)$$

La température maximale de jonction es donnée $T_j = 150^\circ$. On prend $T_a = 35^\circ$

Le datasheet de IIRFP460 nous donne la figure 4.5 [33]:

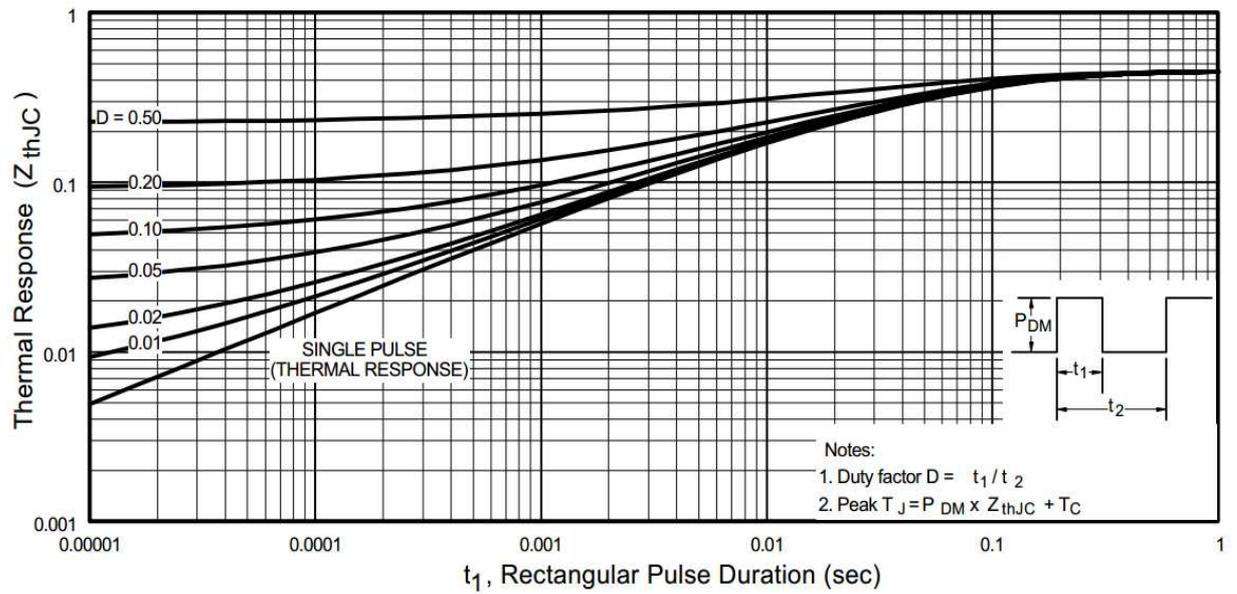


Figure 4.5: caractéristique $R_{th(j-c)}$ en fonction du rapport cyclique D et de temps de conduction du MOSFET.

D'après la Figure 4.5 :

Pour un rapport cyclique maximum ($D= 0.5$), on trouve $R_{th(j-c)} = 0.11 \text{ W}/^\circ\text{C}$.

La résistance thermique entre le boîtier et le radiateur est donnée dans le datasheet :

$$R_{th(c-r)} = 0.24 \text{ W}/^\circ\text{C}$$

On a :

$$R_{th} = R_{th(j-c)} + R_{th(c-r)} + R_{th(r-a)}$$

Donc :

$$R_{th(r-a)} = R_{th(j-a)} - R_{th(j-c)} - R_{th(c-r)}$$

Application numérique :

$$R_{th} = \frac{T_j - T_a}{P_D} = \frac{150 - 35}{0.9 \cdot 0.27 \cdot 12^2 + 3100 \cdot 10^{-12} \cdot 15^2 \cdot 150 \cdot 10^6} = 0.83 \text{ W}/^\circ\text{C}.$$

Donc :

$$R_{th(r-a)} = 0.83 - 0.24 - 0.11$$

$$R_{th(r-a)} = 0.48 \text{ W}/^\circ\text{C}.$$

Il faut donc placer un radiateur de résistance thermique inférieure ou égale à $0.48 \text{ W}/^\circ\text{C}$.

IV.2.2. L'étage de commande :

Au niveau de cet étage, deux principaux éléments doivent être mise en place : le circuit d'isolation de l'FPGA et le circuit de commande approchée des MOSFET.

IV.2.2.1. Le bloc d'isolation :

Il existe plusieurs techniques d'isolation électronique. On cite l'isolation optique et l'isolation électromagnétique, tout dépend de l'application désirée. En effet, l'isolation électromagnétique est utilisée surtout pour les applications de faibles fréquences. En revanche, l'isolation optique présente beaucoup plus des performances pour les hautes fréquences. Dans notre cas il est nécessaire d'utiliser un isolateur optique.

On a utilisé le HCPL2200 (Figure 4.6). Il est très rapide en commutation.

- Caractéristiques de HCPL2200 :
 - Un faible courant d'entrée ($I_{\min}=1.6 \text{ mA}$, $I_{\max}=5 \text{ mA}$).
 - Temps de montée (rise time) $t_r=55\text{ns}$.
 - Temps de décroissance (fall time) $t_f=15 \text{ ns}$.

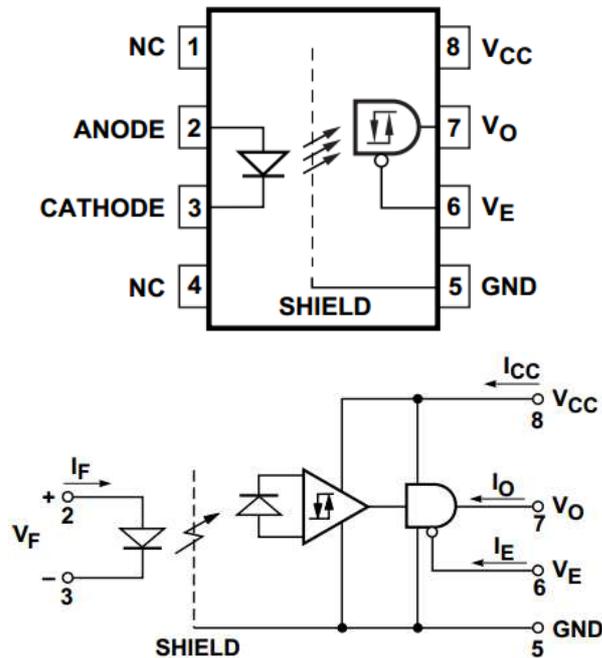


Figure 4.7 : Le Schéma et les pins de HCPL2200.

Le montage de l'HCPL2200 (Figure 4.7) nécessite une capacité de $0.1 \mu\text{F}$ aux bornes de l'alimentation. Une résistance de protection est ajoutée pour limiter le courant aux bornes de la LED ($R=1\text{k}\Omega$).

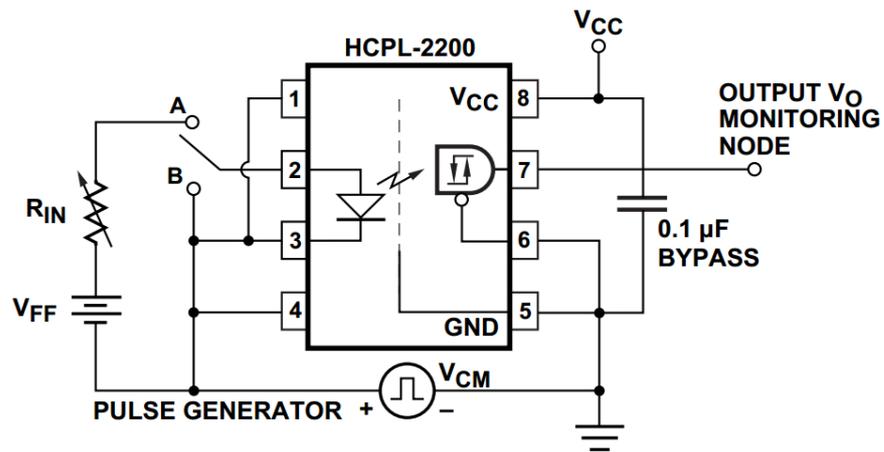


Figure 4.7 : Le montage du HCPL 2200.

Le circuit d'isolation est composé de trois phases. Une capacité de 4.7 μF est placée aux bornes de l'alimentation pour les trois optocoupleurs.

Le circuit d'isolation pour les trois phases est représenté sur la (figure 4.8).

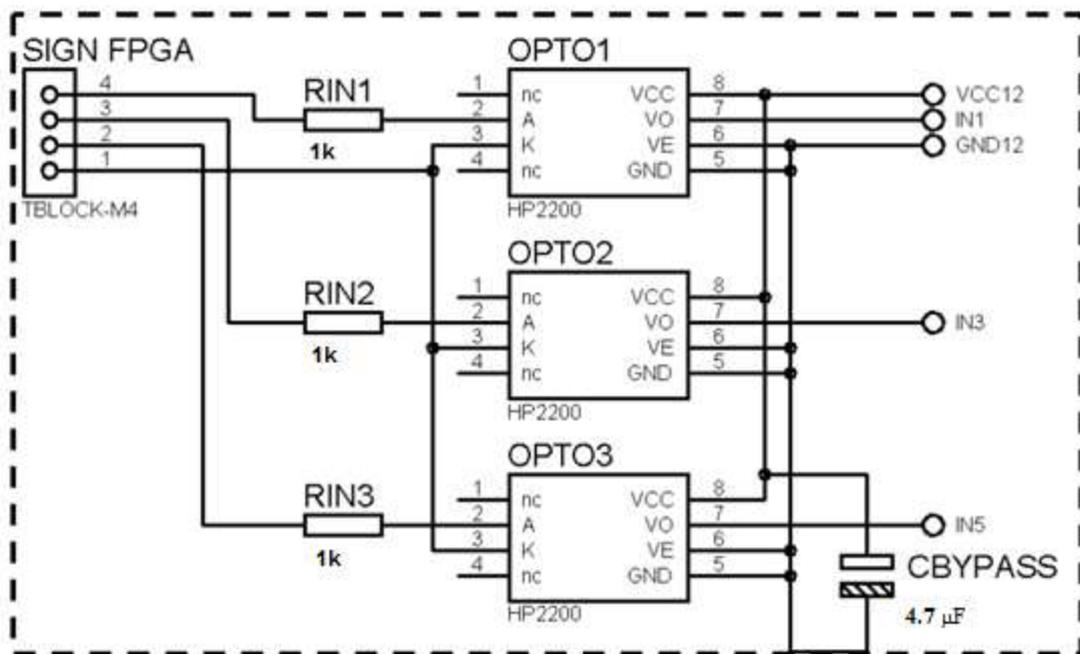


Figure 4.8 : Le circuit d'isolation.

IV.2.2.2. Le bloc de commande approchée:

Nous allons exploiter la technique de bootstrap expliquée en paragraphe (IV.2.1.2.2). Pour cela, on utilise le circuit intégré le driver IR2111 qui sert à la commande des deux MOSFET « high-side » et « low-side » à partir des signaux venants de l'étage d'isolation.

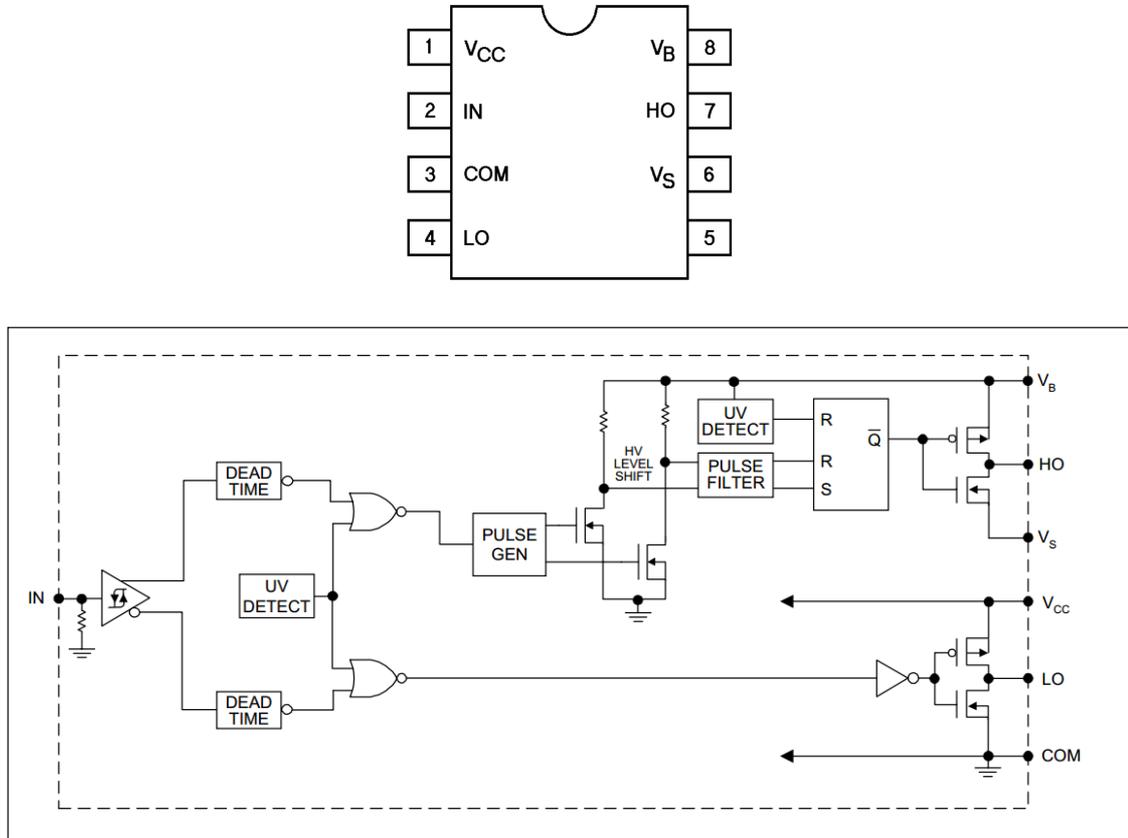


Figure 4.9 : Schéma interne et les pins d'IR2111.

L'IR2111 dispose 8 pins :

- 1 L'alimentation V_{cc} .
- 2 L'entrée de signal de commande issu de la sortie 7 V_o de l'optocoupleur.
- 3 COM il sera relié à la masse.
- 4 L_o si La sortie basse destiné à la commande de MOSFET « low-side ».
- 5 Non utilisé.
- 6 Le point de potentiel flottant.
- 7 La sorite haute pour la commande de MOSFET « high-side ».
- 8 Alimentation flottante.

- **L'utilisation typique d'IR2111 :**

Les connexions des pins d'IR2111 selon le datasheet sont montrées sur la figure 4.10.

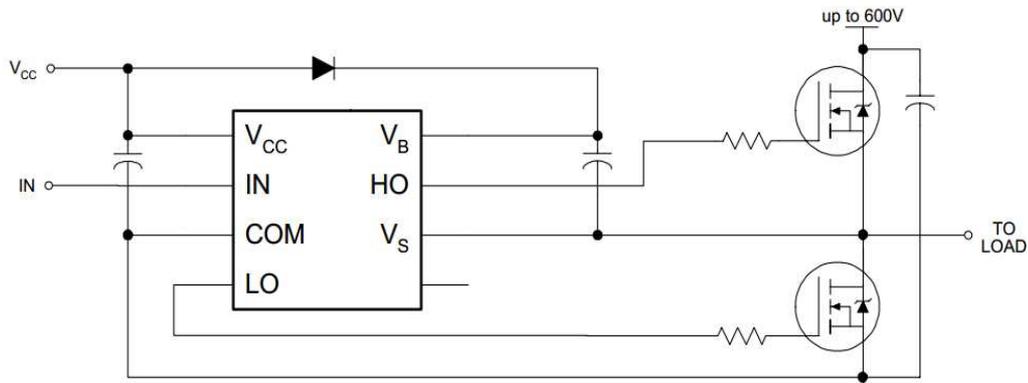


Figure 4.10: Utilisation typique d'IR2111.

La diode de bootstrap est choisie de tel sort qu'elle soit capable de supporter une tension inverse maximale équivalente à la tension d'alimentation de l'onduleur. La tension maximale que le condensateur bootstrap peut atteindre dépend de la tension aux bornes de la diode et la tension d'alimentation V_{cc} .

- **Dimensionnement de condensateur de bootstrap :**

Pour dimensionner le condensateur bootstrap, la première étape consiste à établir la tension maximale (ΔV_{BS}) que nous devons garantir au MOSFET « high-side » lorsqu'il conduit [35].

On à :

$$\Delta V_{BS} = V_{cc} - V_F - V_{GSmin} - V_{DSon} \quad (4.3)$$

Où:

$V_{cc}, V_F, V_{GSmin}, V_{DSon}$: Sont respectivement les tensions d'alimentation, la tension aux bornes de la diode, la tension grille source minimale et la tension drain- source en conduction de MOSFET « low-side » (Figure 4.11).

Application numérique :

$$\Delta V_{BS} = V_{cc} - V_F - V_{GS \min} - R_{DSS} \cdot I \quad \Delta V_{BS} = 12 - 1.6 - 4 - 0.27 \cdot 12$$

$$\Delta V_{BS} = 3.16 \text{ V}$$

$$Q_{TOT} = 105 \cdot 10^{-9} + 50 \cdot 10^{-6} \cdot 0.2 \cdot 10^{-3}$$

$$Q_{TOT} = 115 \text{ nC.}$$

$$C_b = \frac{115 \cdot 10^{-9}}{3.16} = 36 \text{ nF.}$$

Le bloc de commande approchée pour les trois phases est présenté sur la figure 4.12 :

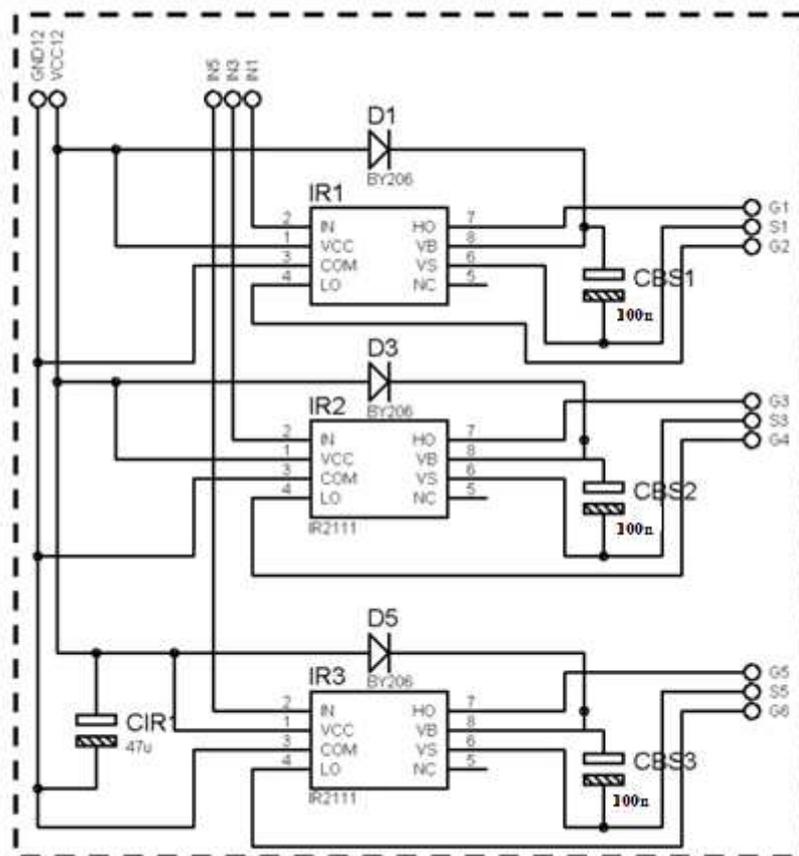
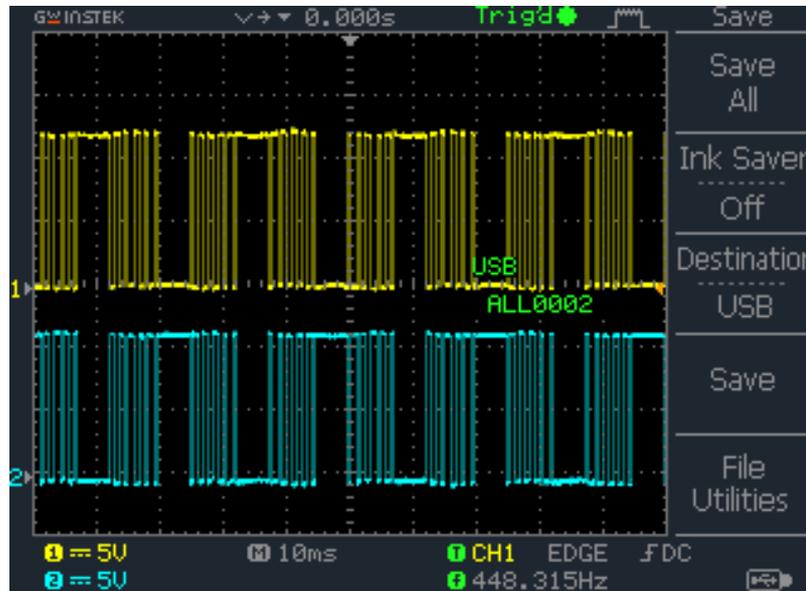


Figure 4.12 : Blocs de commande approchée pour les trois phases.

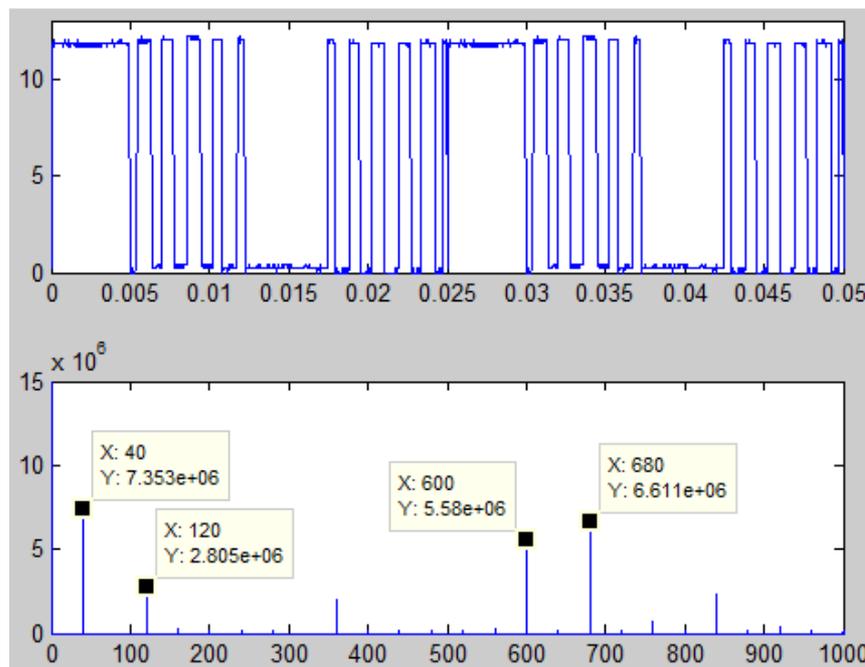
IV.3. Résultats expérimentaux :

Après la conception et la réalisation des deux étages, de commande est de puissance nous avons relevé les signaux de commande et de l'onduleur de tension.

IV.3.1. Les signaux aux sorties de l'étage de commande :



(a) Visualisation de signaux de sorties de l'étage de commande



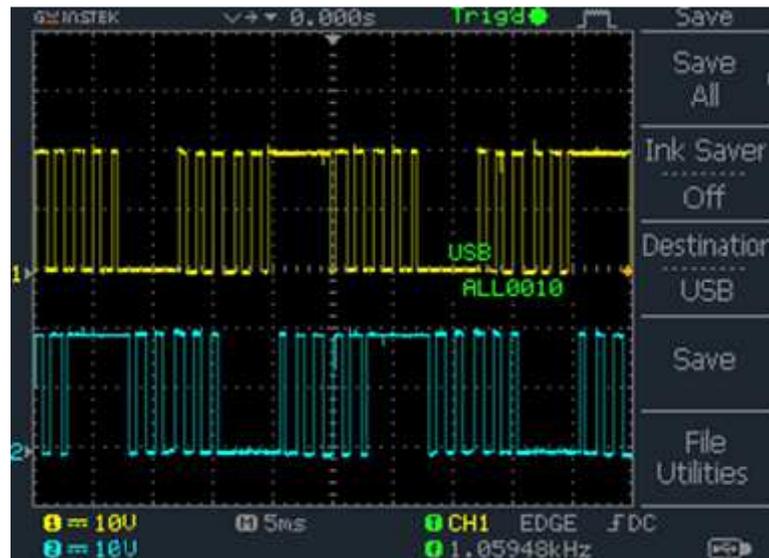
(b) Signal de sortie de l'étage de puissance et son spectre

Figure 4.13 : Le signal à la sortie de l'étage de commande et son spectre.

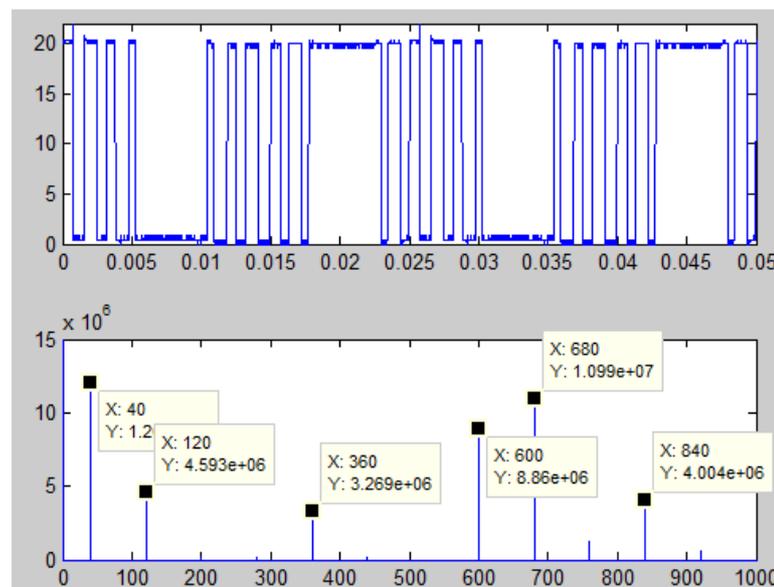
L'oscilloscope a nous a permis de stocker les données dans un fichier excel (quatre mille échantillons avec une periode d'échantillonnage de 0.2 ms a 0.4 ms)

La visualisation confirme que la fréquence $f = 40\text{Hz}$ et que les signaux V_H et V_L sont complémentaires. L'analyse spectrale par Matlab montre l'élimination des harmoniques d'ordre 5, 7, 11 et 13. Alors que les harmoniques d'ordre triple de fondamentale seront éliminées a cause de montage triphasé. Les résultats expérimentaux et celles de la simulation (Chapitre 2) sont compatibles.

IV.3.2. Les signaux aux phases de l'onduleur :



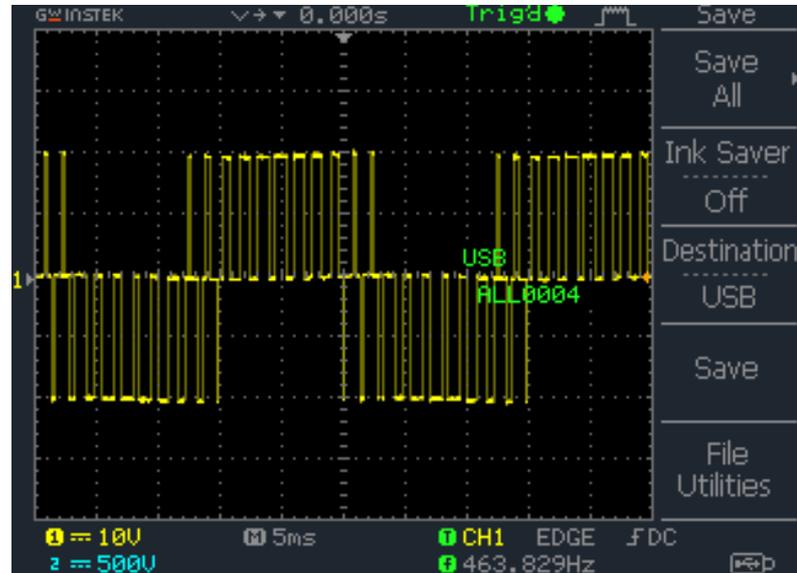
(a) Visualisation de l'oscilloscope.



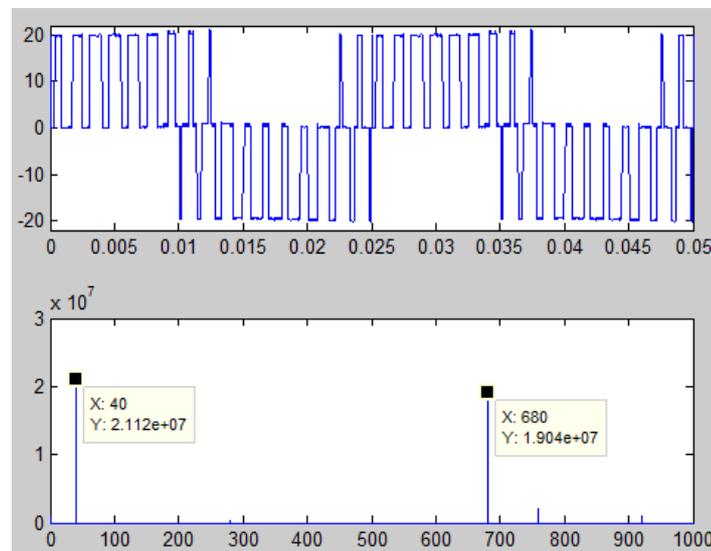
(b) Figure Matlab

Figure 4.14 : La tension entre une phase et la masse.

La figure montre les signaux de phases d'onduleur, la fréquence est de 40 Hz et les tensions sont déphasées de 120° d'une phase à l'autre. La représentation de spectre montre l'élimination d'harmonique d'ordre 5, 7, 11 et 13.



(a) Visualisation de l'oscilloscope



(b) Représentation de spectre par MATLAB

Figure 4.15: La tension entre deux phases.

La représentation de tension entre deux phases montre l'élimination des harmoniques d'ordre 5, 7, 11 et 13, en plus des harmoniques d'ordre triple. Le premier harmonique apparu est de l'ordre 17 d'où l'efficacité de la méthode utilisée.

IV.4. Conclusion :

Dans ce chapitre nous avons vu l'étude consacrée pour la réalisation de variateur de vitesse en débutant par la conception de l'onduleur triphasé et en finissant par le dimensionnement des éléments de l'étage de commande approchée.

Les signaux relevés dans la sortie de l'étage de commande montrent bien l'élimination des harmonique d'ordre désiré. En outre, l'analyse fréquentielle des signaux entre deux phases à la sortie de l'onduleur nous a bien confirmée la suppression des harmoniques d'ordre multiple de trois.

Conclusion Générale

Dans ce projet nous avons réalisé un variateur de vitesse piloté par une commande MLI dont les signaux de commande ont été calculés par élimination sélective des harmoniques et asservissement du fondamentale. Ces signaux ont été générés par un algorithme basé sur les réseaux de neurones.

Ce variateur de vitesse comporte un onduleur de tension à deux niveaux, une carte de commande pour l'isolation et la commande approchée des interrupteurs et une carte FPGA pour générer les signaux de commande.

L'algorithme utilisé a prouvé ses performances en ce qui concerne la précision et la rapidité de réponse lors de passage d'une valeurs d'indice de modulation à une autre. Ceci est du à l'efficacité des réseaux de neurones dans les applications en temps réel.

Cette efficacité est résultante de la simplicité des programmes dont le procédure de calcul des angles de commutation s'agit des simples équations où les paramètres ont été prédéfinies et calculés sur Matlab.

De plus, l'analyse spectrale de ses signaux MLI nous a bien montrée que l'algorithme utilisé a bien éliminé les harmoniques non désirés. Les harmoniques multiples de trois sont éliminés automatiquement dans les tensions entre phases.

Une amélioration significative trouvée dans la technique proposée est son aptitude à contrôler la tension et les fréquences séparément ce qui permet le contrôle séparé du couple et de la vitesse du moteur à induction. Les caractéristiques couple-vitesse peuvent alors répondre à toutes les exigences de traction et le cycle de conduite du véhicule électrique

Pour plus de performance, nous proposons comme solutions :

- L'utilisation des onduleurs à plusieurs niveaux pour éliminer plus d'harmonique.
- Ajouter une boucle de retour pour automatiser le système sur la vitesse désirée.
- Déterminer les paramètres des réseaux pour plus de valeurs d'indice de modulation pour couvrir toute la plage de travail.
- Pour réduire la puissance dissipée par l'onduleur il est recommandé d'utiliser des interrupteurs moins dissipatifs d'énergie.

Bibliographie :

- [1] H.KAHLANE, Étude et réalisation d'un hacheur à quatre quadrants, commandé par un DSP TMS 320F2812, destiné à un véhicule électrique. Mémoire de magister de l'Ecole Nationale Polytechnique, Alger 2012.
- [2] W.LAJNEF. Modélisation des super condensateurs et évaluation de leur vieillissement en cyclage actif à forts niveaux de courant pour des applications véhicules électriques et hybrides. Thèse de doctorat. Université Bordeaux 2006.
- [3] W.LHOMME Gestion d'énergie de véhicules électriques hybride basé sur la représentation énergétique macroscopique. Mémoire de doctorat Université des Sciences et Technologies de Lille 2007.
- [4] A. ZOGHBI, M. YAHIA-AMMAR, Implémentation sur FPGA d'une commande MLI à élimination des harmoniques pour un onduleur sept-niveaux en pont H cascadié. Mémoire de l'ingénierat, Ecole Nationale Polytechnique, Alger 2012.
- [5] D. NAMANE. Machine à courant alternatif machines asynchrones, synchrones, convertisseurs électromécaniques. ENP. février 2010.
- [6] E. GAUCHERON, Les moteurs électriques ...pour mieux les piloter et les protéger Cahier technique n°207 Schneider.
- [7] L. BAGHLI, "Modélisation et commande de la machine asynchrone", Notes de cours, IUFM de Lorraine, UHP 2005.
- [8] F.BEN AMMAR, Machine asynchrone à contrôle vectoriel de flux, Docteur de l'institut national polytechnique de Toulouse, 2002.
- [9] F.BERREZZEK Etude des Différentes Techniques de Commande des Onduleurs à MLI Associés à une Machine Asynchrone. Mémoire de magister, université Badji Mokhtar , Annaba 2006.
- [10] E.Acha, V.G.Agelidis,O.Anaya-Lara,"Power Electronics Control in Electrical Systems" Newnes 2002.
- [11] J.R. Espinoza,"Inveters" In: M.H.Rashid,"Power Electronic Handbook", Academic Press, USA 2001.

- [12] E.M.BERKOUK,"Contribution à la conduite des machines asynchrones monophasées et triphasées alimentées par des convertisseurs directs et indirects .Application gradateurs et onduleurs multiniveaux." Thèse de doctorat. C.N.A.M, Paris 1992.
- [13] R.KABOUCHE,"Etude et élaboration d'une technique MLI à élimination d'harmoniques optimale", Mémoire de Magister en Electronique, Ecole Nationale Polytechnique, Alger, 2002.
- [14] A.GUELLAL. Implémentation sur FPGA d'une commande MLI on-line basée sur le principe des réseaux de neurones, Mémoire de Magister en Electronique ENP 2009
- [15] Amar Guellal, Cherif Larbes, Douadi Bendib, Douadi Bendib, Ali Malek. FPGA based on-line Artificial Neural Network Selective Harmonic Elimination PWM technique, CDER, ENP 2014
- [16] Daniel Depernet. Optimisation de la commande d'un onduleur mli a trois niveaux de tension pour machine asynchrone. . Thèse de doctorat. Université de Reims - Champagne Ardenne, French. 1995.
- [17] Patel Hasmukh S, Hoft Richard G. Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part I-harmonic elimination. IEEE Trans Indus Appl 1973.
- [18] J.A.TAUFIQ. B.MELLITT And C.J.GOODMAN, 'Novel algorithme for Generating near optimal PWM waveforms for AC traction drives', IEE Proceedings, Vol.133,PT.B,No2,March 1986.
- [19] Khider M, " commande de vitesse en temps réel d'un moteur asynchrone triphasé " Mémoire de magister, Ecole Nationale Polytechnique, Alger, 2003.
- [20] C. TOUZET, Les réseaux de neurones artificiels introduction au connexionnisme cours, exercices et travaux pratiques, Juillet 1992.
- [21] Bullinaria. JA. Analyzing the internal representations of trained neural networks. Institute of Physics Press, Bristol, UK, 1997.
- [22] Denis Rabasté, stage de programmation des CPLD et FPGA en VHDL sous Max plus II mars 2002, IUFM d'Aix-Marseille
- [23] M. Philippe, Les circuits logiques programmables, Letenneur –Lycée Julliot de la Morandière, GRANVILLE.

- [24] N. MARQUES, Méthodologie et architecture adaptative pour le placement efficace de tâches matérielles de tailles variables sur des partitions reconfigurables, Doctorat de l'Université de Lorraine. 2012.
- [25] A.SAHOUR, Implémentation sur FPGA d'un algorithme de bruitage en utilisant 1D-DWT. Thèse de doctorat, Université Badji Mokhtar Annaba. 2013.
- [26] F. CHEKIRED; Etude et implémentation d'une commande MPPT neuro-floue sur FPGA. Mémoire de magister; ENP 2008.
- [27] T. BLOTIN, le langage de description VHDL, Lycée PAUL ELUARD, 93206 SAINT-DENIS.
- [28] FOCH (H.), FOREST (F.) et MEYNARD (T.). Onduleurs de tension. Structures. Principes. Applications. D 3 176, 1998.
- [29] WEI MI. Extraction de paramètre et domaine de validité du modèle d'un composant de puissance. Thèse de doctorat de l'institut National des sciences Appliquées de Lyon. Juillet 2002.
- [30] DANG BANG VIET. Conception d'une interface d'électronique de puissance pour Pile à combustible. Thèse de doctorat de l'Université Joseph Fourier, Décembre 2006.
- [31] SCHULTZ (W.). Drive techniques for high side N channel MOSFET. Revue PCIM, Juin 1987, pp. 34-40.
- [32] D.BENDIB Etude et réalisation d'une commande MLI on-line sur circuit FPGA, mémoire de magister. ENP 2007.
- [33] International Rectifier. SMPS MOSFET IRFP460A. Data Sheet No. PD- 91880.
- [34] B. Jayant Baliga. Fundamentals of Power Semiconductor Devices. Power Semiconductor Research Center North Carolina State University. 2008 Springer Science + Business Media, LLC. p427.
- [35] Merello, A. Rugginenti and M. Grasso. Gate drivers using high voltage gate driver monolithic. International Rectifier. DT04-04.
- [36] Andrea Merello. Bootstrap Network Analysis: Focusing on the Integrated Bootstrap Functionality. International Rectifier. AN1123.

La carte FPGA utilisée :



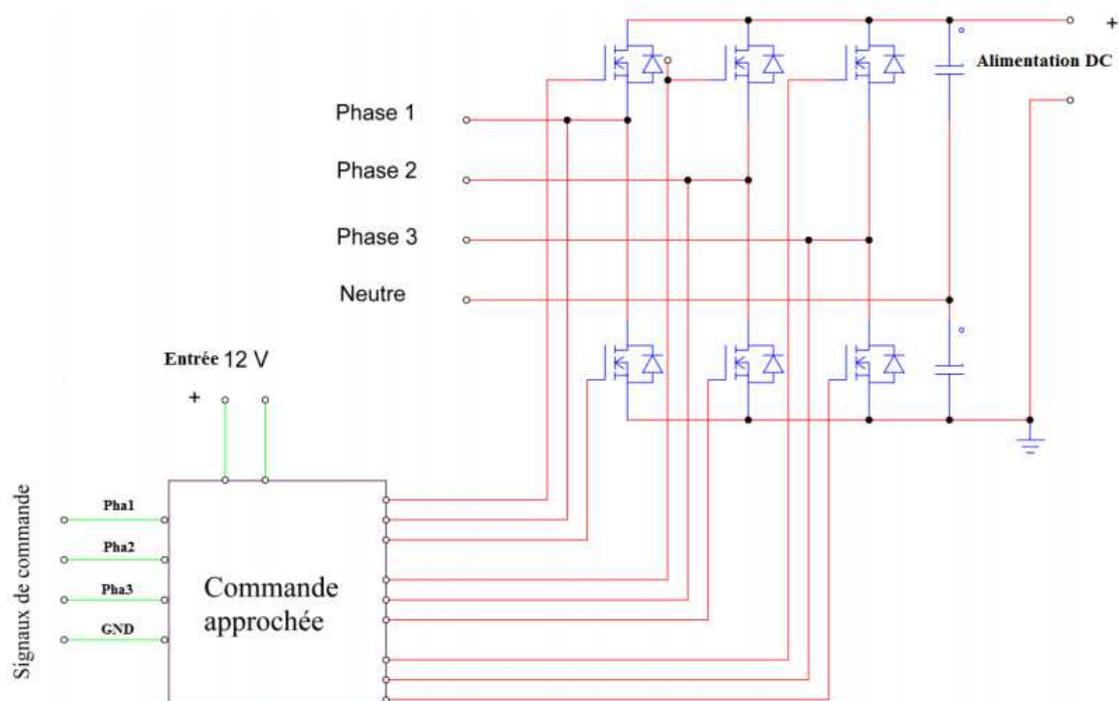
Figure 3.9 : Carte de développement utilisée.

Les dispositifs principaux de la carte Spartan-3E sont :

- Xilinx XC3S500E Spartan-3E FPGA
 - Jusqu'à 232 bornes d'entrée/sortie utilisateur.
 - 320 pins FBGA package.
 - Plus de 10.000 cellules logiques.
- Mémoire PROM de Xilinx à 4 Mbit de configuration instantané.
- 64 macrocell XC2C64A CoolRunner™ CPLD de Xilinx.
- DDR SDRAM (512 Mbit), interface des données × 16.
- Flash NOR parallèle 16 Moctet (128 Mbit) (Intel StrataFlash).
 - Stockage de la configuration du circuit FPGA.
 - Stockage du code de MicroBlaze/filature.
- flash série SPI de 16 Mbits (STMicro).
 - Stockage de configuration de FPGA.
 - Filature de code de MicroBlaze.
- 2 lignes, d'écran LCD de 16 caractères.
- Port PS/2 (de la souris ou de clavier).
- Port d'affichage VGA.
- Port Ethernet PHY 10/100 (exige le MAC d'Ethernet dans FPGA)

- Deux 9 ports d RS-232 (DTE et DCI-modèle)
- Port USB.
- Horloge de 50 mégahertz.
- EEPROM périodique pour le système anti-copie de bitstream.
- Connecteur d'expansion de Hirose FX2.
- Quatre convertisseur numérique-analogique SPI (DAC).
- Deux entrée de convertisseur analogique-numérique SPI-basé (CDA).
- Port d'élimination des imperfections de ChipScope™ SoftTouch.
- Rotatoire-encodeur avec l'axe `a bouton-poussoir.
- Huit LED discrètes.
- Quatre contacts coulissants.
- Quatre commutateurs à bouton-poussoir.
- Entrée d'horloge de SMA.
- douille d'IMMERSION de 8 bornes pour l'oscillateur auxiliaire d'horloge.

Schéma de l'onduleur triphasé:



Circuit imprimé de l'étage de commande :

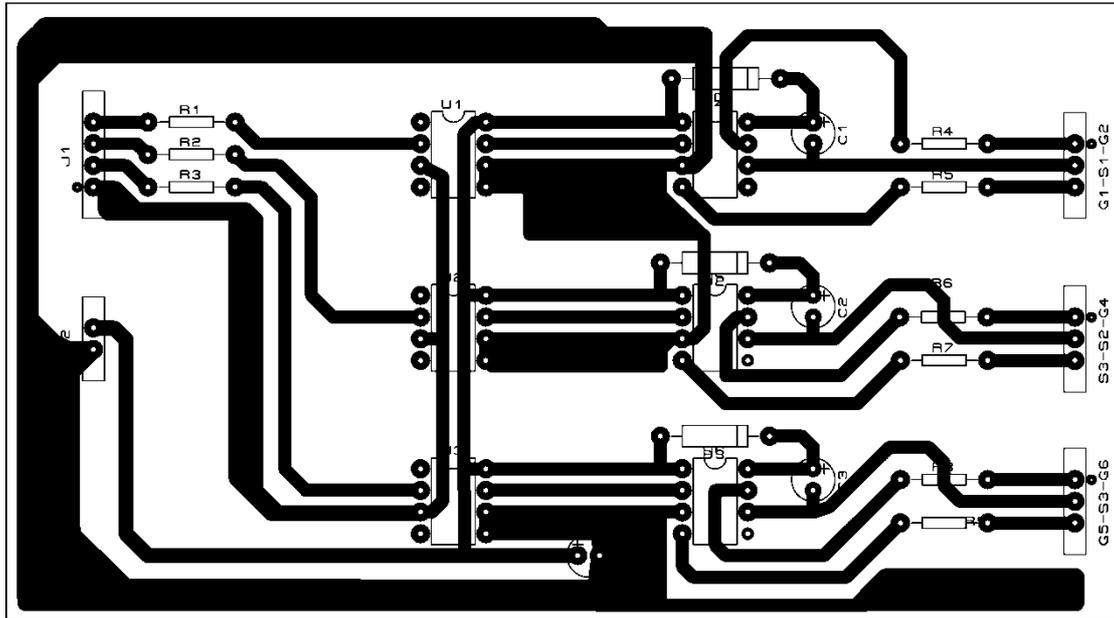


Photo de la carte de commande :

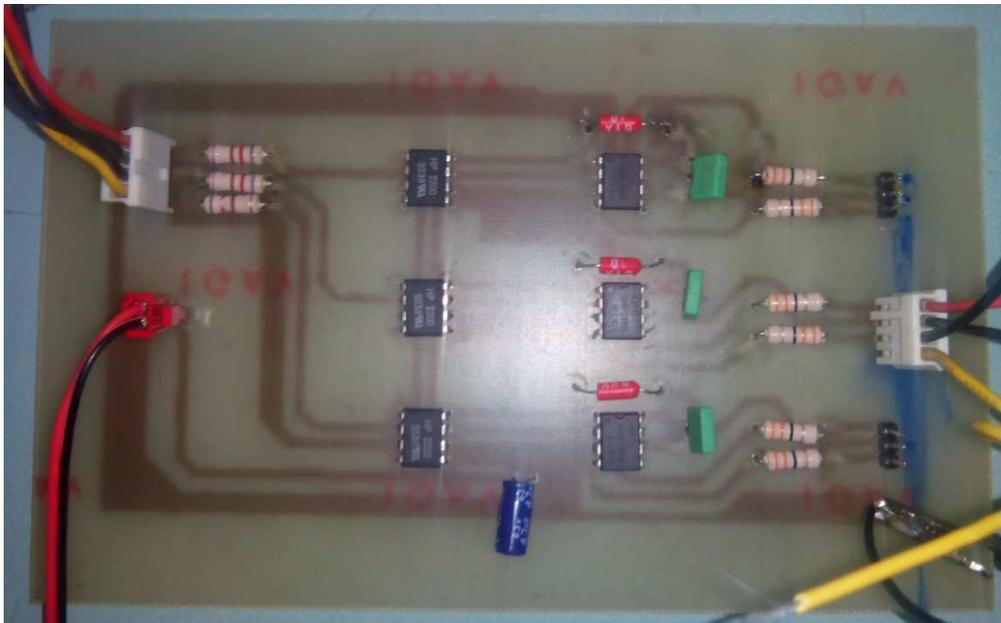


Photo de variateur de vitesse réalisé :

